Applied 3D Vision - An Empirical Study.

Jensen, Sebastian Hoppe Nesgaard

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

Ph.D. Thesis
Doctor of Philosophy

DTU Compute
Department of Applied Mathematics and Computer Science

Applied 3D Vision

An Empirical Study

Sebastian Hoppe Nesgaard Jensen

Kongens Lyngby 2017
PhD-2017-467

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (english)

3D vision technology is the process of estimating 3D geometry from 2D image data. In recent years, it has reached a maturity that allows for real world usage, no longer being confined to a laboratory. We see this in the availability of commercial 3D scanners (e.g. Kinect, RealSense, GOM) and their many applications (e.g self-driving cars, automation, quality control). However, as any engineer knows the transition from lab to real world application is not trivial, often with unforeseen challenges. In this sense, much 3D vision technology is built on an unknown foundation, as there have been few studies on it's practical problems and limitations.

This thesis contributes to several subjects within the field of 3D vision with such studies. These encompass dataset creation, empirical evaluation and system engineering.

Datasets are essential to quantitative evaluation and testing. Thus we have created datasets for two fields which are lacking in that area.

The first being a dataset for Non-Rigid Structure from Motion (NRSfM). NRSfM estimates the 3D geometry of a deforming object from a 2D point sequence, thus the dataset is comprised 2D point sequences with a recorded 3D reference. We accomplished this using structured light scanning and several stop-motion animatronics. This allowed for much greater deformation variety than what has previously been available. Structured light scanning provides dense reference geometry and surface normals, which allowed us to created occlusion-based missing data for each point sequence. Something which has not been done before.

The second dataset is built for evaluation of rendering techniques for challenging scenes. In it, we record a series of images along with precise geometry, radiometry, environment and camera pose. The intent is for a rendering algorithm to use said data to recreate the recorded image.

Datasets serves little purpose unless they are used. Therefore, we have applied our NRSfM dataset to analyze the field using 16 methods representative of the state-of-the-art. Our factorial analysis shows not only which methods gives the most precise results, but also overall trends in the field. For example which deformations are the most challenging to reconstruct and how the camera impacts reconstruction quality. We also show that the previous reliance on random missing data has lead to algorithms that handles the missing data from self-occlusion poorly.

We have also evaluated several structured light techniques on biological material. Structured light is designed with the assumption of diffuse reflection, but most biological material has heavy subsurface scattering. We show that this results in subtle, systematic overestimation of depth (up to 1mm), even for state-of-the-art techniques. However, we also demonstrate that a large part of this error can be corrected with a linear, geometry based model.

This thesis also presents some vision-based solutions to practical problem, as some information can only be gained through application. First, we investigate the interaction between 3D vision and robotics by engineering a solution for non-rigid bin picking. Our system shows that the problem is solvable, but error correction remains a big concern. Errors from multiple sources such as calibration, 3D scanner, segmentation and pose estimation might seem insignificant individually, but are problematic when taken as a whole.

Second, we designed an algorithm for automatic measurement of contact surface areas for usage in tribology testing. The method performs measurements with an error of less than $0.4\mu m$.

Summary (danish)

3D vision er teknologi der estimerer 3D geometry fra 2D billede data. I løbet af de seneste år, har feltet nået en teknologisk modenhed der muliggør brug uden for laboratiet. Vi ser denne tendens i antallet af komercielle 3D skannere (fx. Kinect, Realsense, GOM) og deres mange applikationer (fx. selv-kørende biler, automation, kvalitets kontrol). Som enhver ingeniør ved er overførelsen af teknologi fra laboratoriet til den virkelige verden langt fra triviel, ofte præget af uforudsete problemstillinger. In den forstand, bygger meget 3D vision teknologi på et usikkert fundament eftersom at der har kun være få studier af de praktiske problemstillinger og udfordringer.

Denne afhandling bidrager med sådanne studier til flere felter indefor 3D vision.

Datasæt er en essentiel del af kvantitativ evaluering og test. Derfor er to datasæt blevet designet og implementeret som en del af denne afhandling.

Det første er et Non-Rigid Structure from Motion (NRSfM) datasæt. NRSfM estimerer deformerbare objekters 3D geometri udefra en sekvens af 2D punkter, vores datasæt består derfor af sådanne sekvenser med tilhørende 3D reference. Vi lavede dette data vha. struktureret lys skanninger og en håndful stop-motion animatronic. Derfor har vi kunne inkludere nye deformationstyper der ikke har været tilgængelig før. Derudover har vores brug af struktureret lys skanning givet os tætpakket overflade geometri og overflade normaler. Denne data har vi brugt til at skabe okklusions-baseret 'missing data', hvilket ikke er set før indefor NRSfM.

Det andet datasæt blev skabt med henblik på evaluering af renderings teknikker for udfordrende scener. Vi har optaget en series billeder med tilhørende scene geometri, radiometri, lysmiljø og kamera positioner.

Datasæt tjener ikke med formål hvis de ikke bliver brugt. Derfor har vi udført en evaluering af NRSfM feltet vha. vores datasæt. Feltet er her repæsenteret af 16 af de mest relevante NRSfM metoder. Vores faktoranalyse viser ikke blot hvilke metoder er de mest præcise, men også overordnede tendenser indefor NRSfM feltet. For eksempel hvilke deformationer er de sværeste at rekonstruere og hvorledes kamera påvirker rekonstruktions kvalitet. Vi har også påvist at tidligere brug tilfældig 'missing data' har ført til algorithmer der håndtere vores okklusions-baseret 'missing data' dårligt.

Vi har også evalueret flere struktureret lys metoder på biologisk materiale. Struktureret lys er designed med en antagelse om diffus refleksion, men de fleste biologiske

materialer spreder lys under deres overflade. Vi har påvist at dette resulterer i en systematisk overestimation af dybde (op til 1mm), selv for de nyeste teknikker. Heldigvis kan en stor del af denne fejl rettes vha. en simpel linear model.

Denne afhandling behandler også et par pratiske problemstillinger ved brug af vision. Dette blev gjort eftersom visse informationer kun kan indsamles under praktiske studier. Vi har undersøgt interaktionen mellem 3D vision og robotteknologi ved at designe og implmentere en løsning for bin picking af deformerbare objekter. Vores system viser at problemstillingen kan løses, men også at fejlhåndtering er meget vigtigt. Fejl fra flere kilder såsom kalibrering, 3D skanning, segmentering og positur estimation kan synes individuelt ubetydelige, men summen af disse er problematisk. Derudover har vi designed et system for automatisk måling af kontakt overfalde areal til brug i tribologi test. Metode udfører målinger med en præcision på under $0.4\mu m$.

Preface

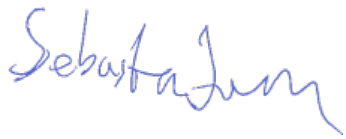
This thesis was prepared at the Image & Computer Graphics section of the Department of Applied Mathematics and Computer Science at the Technical University of Denmark (DTU). It was done in fulfilment of the requirements for obtaining a doctor of philosophy (Ph.D.) within the topic of computer vision.

The work was primarily funded by the Manufactoring Academy of Denmark (MADE) as part of MADE Spir's Workpackage 8 named Hyperflexible Automation.

This thesis presents research done on 3D modeling of deformable objects with vision technology. This work is primarily empirical with contributions to both basic and applied science. Specifically this thesis contributes to Non-Rigid Structure from Motion and vision-guided robotic handling of deformable objects. This work is primarily documented in three papers included in this thesis found in appendix A, B and C. Additional papers is included regarding work done in other fields such as geometric metrology (appendix F), games (appendix G) and vision datasets (appendix D and E).

The project has been supervised by Associate Professor Henrik Aanæs and co-supervised by Professor Norbert Krüger, respectively of DTU and the Southern University of Denmark (SDU). The work was primarily carried out at DTU, but with an external research stay at Istituto Italiano di Tecnologia under supervision of Alessio del Bue.

Kongens Lyngby, December 24, 2017



Sebastian Hoppe Nesgaard Jensen

Acknowledgements

First and foremost I would like to thank my supervisors Associate Professor Henrik Aanæs and Professor Norbert Krüger. I feel that I have been especially fortunate in having Henrik Aanæes as my main supervisor. You have been greatly supportive, providing good advice on both matters of both science and life.

I would also like to thank the Manufacturing Academy of Denmark for starting this project and providing me with this opportunity.

Of course, an acknowledgement section wouldn't be complete without mentioning my fantastic colleagues at the section for Image Analysis and Computer Graphics at DTU. Always up for either a good discussion and/or laugh, you truly made the workdays both fun and fulfulling.

Lastly I most thank my friends and family, their support has been absolutely vital, especially during the more turbulent times. Without you, I wouldn't have made it through.

List of Contributions

The following lists the contributions which are part of this thesis, ordered according to type. Of these paper A, paper B, paper C and workshop H should be considered the main contributions. The full text and context will be provided in later chapters.

Papers

Paper A

Sebastian Hoppe Nesgaard Jensen, Alessio Del Bue, Henrik Aanæs, and Mads Emil Brix Doest. "A Benchmark and Evaluation of Non-Rigid Structure from Motion". In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (Under Review)* (TBD)

Paper B

Troels Bo Jørgensen, Sebastian Hoppe Nesgaard Jensen, Henrik Aanæes, and Norbert Krüger. "An Adaptive Robotic System for Doing Pick and Place Operations with Deformable Objects". In: *Robotics and Computer-Integrated Manufacturing (Under Review)* (TBD)

Paper C

Sebastian Nesgaard Jensen, Jakob Wilm, and Henrik Aanæs. "An Error Analysis of Structured Light Scanning of Biological Tissue". In: *Scandinavian Conference on Image Analysis*. Springer. 2017, pages 135–145

Paper D

Jonathan Dyssel Stets, Alessandro Dal Corso, Jannik Boll Nielsen, Rasmus Ahrenkiel Lyngby, Sebastian Hoppe Nesgaard Jensen, Jakob Wilm, et al. "Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering". In: *Applied Optics* 56.27 (2017), pages 7679–7690

Paper E

Henrik Aanæs, Knut Conradsen, Alessandro Dal Corso, Anders Bjorholm Dahl, A Del Bue, Sebastian Hoppe Nesgaard Jensen, et al. "Our 3D Vision Data-Sets in the Making". In: *The Future of Datasets in Vision 2015* (2015)

Posters

Poster F

Ömer Can Kücükyildiz, Sebastian Hoppe Nesgaard Jensen, and Leonardo De Chiffre. "Contact area measurements on structured surfaces". In: *Euspen's SIG Meeting*. 2017

Poster G

Alessandro Dal Corso, M Olsen, Kasper Hornbak Steenstrup, Jakob Wilm, Sebastian Hoppe Nesgaard Jensen, Rasmus R Paulsen, et al. "VirtualTable: a projection augmented reality game". In: *SIGGRAPH Asia 2015 Posters*. ACM. 2015, page 40

Workshops

Workshop H

Sebastian Hoppe Nesgaard Jensen, Alessio Del Bue, Henrik Aanæs, and Yaser Sheikh. *Non-Rigid Structure from Motion Challenge 2017*. 2017. URL: `http://nrsfm2017.compute.dtu.dk/` (visited on October 26, 2017)

Acronyms

NRSfM Non-Rigid Structure from Motion

SfM Structure from Motion

MRF Markov Random Field

DFT Discrete Fourier Transform

DCT Discrete Cosine Transform

ANOVA Analysis of Variance

SVD Singular Value Decomposition

EM Expectation-Maximization

CAD Computer Assisted Design

CNN Convolutional Neural Network

ICP Iterative Closest Point

PCA Principal Component Analysis

MRF Markov Random Field

CRF Conditional Random Field

GMM Gaussian Mixture Model

DoF Degree of Freedom

ROS Robot Operating System

PCL Point Cloud Lib

Notation

Matrix
Given by upper case, bold letters e.g. $\mathbf{A}$.

Vector
Given by lower case, bold letters e.g. $\mathbf{x}$.

Entry of a Matrix
Given by a lower case of the corresponding matrix symbol with location as subscript. The (i, j) entry of $\mathbf{A}$ would be a_{ij}.

Matrix Row
Given by bold, lower case of the corresponding matrix symbol with location and * as subscript. Row i of $\mathbf{A}$ would be $\mathbf{a}_{i*}$.

Matrix Column
Given by bold, lower case of the corresponding matrix symbol with * and location as subscript. Column j of $\mathbf{A}$ would be $\mathbf{a}_{*j}$.

Mean
Given by the expectation operator e.g. $E[\mathbf{x}]$.

Contents

CHAPTER 1

Introduction

3D vision is the science of estimating 3D geometry from 2D data, typically from images. While it has long been confined to laboratories, recent advances has seen many 3D vision methods move from the lab to application on real-world problems. This is evident by the availability of commercial 3D scanning technology (e.g Kinect, RealSense and GOM) and their many applications (e.g. self-driving cars, robotics and quality control). And it is easy to see why, as 3D vision is a fast and non-destructive method for acquiring rich 3D data. However, as any engineer knows the move from lab to real-world application is not trivial. In this sense, the application of much 3D vision technology is built on an unknown foundation, as there has been few studies on the complications and limitations that appear in the real world.

1.1 Scope

This thesis presents a study on the practical applications of 3D vision. We wanted to clearly define current possibilities and challenges. The field of 3D vision is vast therefore the scope of this thesis is limited to studying a handful of problems. However, we do feel that these are representative of the overall state of 3D vision.

1.1.1 Evaluation of Non-Rigid Structure from Motion

Structure from Motion (SfM) is the science of estimating 3D geometry from a set of 2D points, typically obtained from an image sequence. It invokes a rigidity prior to constrain the problem. Non-Rigid Structure from Motion (NRSfM) forgoes this prior, allowing for reconstruction of a deforming scene. While SfM is well understood, NRSfM is not quite as technologically mature. One reason for this is that NRSfM is inherently a more difficult problem than SfM. Another is that there has been little effort to evaluate and study the many methods that has been published over the years. Thus it is unknown which algorithms work well and which areas future research should focus on.

As we see it there are two primary barriers. First, a lack of high-quality, varied datasets with ground truth, which can be used for quantitative study. Second, a factorial evaluation protocol that can examine the state-of-the-art in NRSfM in sta-

tistical sound manner. A resolution for both is presented in this thesis. As such, we will make it clear what contemporary NRSfM can do and where the main challenges lie.

1.1.2 Flexible Automation using 3D Vision

Contemporary automation has a rigidity problem, though not one of form, but of setup. Current automation solutions are designed to do one task and one task only. This means that automation is only financially feasible for products that are constantly manufactured. As such, many tasks still has to be performed with 100% manual labor. Additionally, the robots timing and movement is completely preprogrammed. This means that great care must also be taken in designing the environment around the robot, as unknowns cannot be dealt with. This also means that rigid objects are far easier to deal with than non-rigid. Therefore, it is of interest to combine 3D vision technology with robotics to create more flexible automation. The idea being that online gathered geometric information can be used to make decisions and adjustments in real-time.

Flexible automation with 3D vision is studied in this thesis. Not only can we learn which problems are feasible to solve, but we will also uncover the major challenges in integrating vision and robotic path planning.

The subject will be studied through a use case at Danish Crown, Ringsted. Danish Crown, Ringsted is a large slaughterhouse which make meat products and cutouts. The final stage for a product is packaging, where the cutouts is picked up from an plastic box and placed on a conveyor belt or a cardboard box and then sent to shipping. This process is illustrated in Figure 1.1. While this process seems simple, it has so far proven infeasible to automate. The reason being that cutout shapes and sizes often changes, which makes offline programmed solutions unworkable. Additionally, products arrives in an unordered pile. As such, We study how 3D vision may be used to overcome both of these challenges.

1.1.3 Error Analysis of Structured Light

Structured light scanning is an active 3D scanning technique. It projects one or more patterns onto a scene, captures images and estimates surface geometry based on these. Examples of commercial structured light includes Kinect V1, RealSense and GOM. While structured light has provided a quick and easy way of gathering rich geometric information, it is based on assumptions that is unfortunately often violated in the real-world.

The primary assumption being that the observed signal in the captured images is primarily the results of the direct reflection of the projected pattern. Generally speaking, the real world breaks this assumption in two ways inter-reflections and sub-

Figure 1.1: The practical use case for this thesis. The deceptively simple process of picking meat from an unorganized pile (left) and placing it on e.g. a conveyor belt.

surface scattering. Inter-reflections is when the scene reflects the projected pattern internally. Subsurface scattering describes light entering the scene material and being scattered before being emitted back into the environment.

The effects of subsurface scattering on structured light scanning is studied in this thesis, as this was previously poorly understood. And it is important to understand as many real world materials (e.g. plastic, biological tissue and cloth) exhibits subsurface scattering.

1.1.4 Evaluation of Photorealistic Rendering

As is evident from the video game industry and cinema, computer graphics has come a long way since it's inception. We are infact capable of creating imagery that closely resemble real world photographies. However, little effort has gone into quantifying the realistism of state-of-the-art rendering. The main reason for this is that creating a dataset with input (geometry, radiometry, camera pose, environment map) and a corresponding ground truth image is extremely difficult.

Understanding the precision of state-of-the-art computer graphics is quite important for computer vision, as it is increasingly being used to create synthetic training data for deep learning. Indeed, training on synthetic data might not transfer well to the real world, if said data is subtlety biased or flawed.

Therefore, we present the implementation of such a dataset with a corresponding evaluation of state-of-the-art rendering techniques in this thesis.

1.1.5 Geometric Metrology using Vision

Geometric metrology is the science of geometric measurements and uncertainty estimation. One of it's central concepts is traceablity, which is an unbroken chain

of comparisons to a stated reference (e.g. the meter standard) with uncertainty estimates. As such geometric metrology is relevant for 3D vision and vice versa. However, uncertainty estimates and traceability for computer vision is still somewhat of an open question.

In this thesis we will study establishing traceability and uncertainty for vision using a specific problem. The problem being the automatic measurement of contact surface area using microscopy data.

1.2 Objectives

So in summary the objectives of this thesis was,

1. Create a high-quality, realistic dataset for NRSfM.
2. Evaluate the field of NRSfM and find the major challenges.
3. Create a flexible automation system for bin-picking of meat by integrating 3D vision and robotics.
4. Study the accuracy of structured light w.r.t. subsurface scattering.
5. Create a realistic dataset for evaluation of photorealistic rendering, with input (geometry, radiometry, camera pose, environment map) and corresponding ground truth imagery.
6. Study the uncertainty and traceability of vision by solving the metrological problem of automatic contact surface area estimation.

1.3 Thesis Overview

This thesis is structured as follows:

Chapter 2
Here we will go over some of the theory needed to understand this thesis. It will primarily concern camera geometry and NRSfM. Structured light scanning has served a substantial role, both in implementing solutions and gathering data. Therefore this chapter also features the basic concepts and theory of structured light scanning.

Chapter 3
In this chapter, we will review the relevant literature. We will primarily be focusing on the main contributions. As such it will be divided into two parts: one concerning state-of-the-art NRSfM algorithms and one concerning robotic handling and bin-picking of deformable objects.

Chapter 4

In this chapter, we will summarize the primary contribution that has been made during this thesis. It will primary be based on the peer-review publications which are listed in the thesis frontmatter with additional details added as needed.

Chapter 5

Here we will summarize the thesis and reflect on it as a whole. We will consider whether the thesis goals has been met.

CHAPTER 2

Background

In this chapter we will outline some of the background theory necessary to understand the contents of this thesis. Since a lot of the work herein is based on camera and multi-view geometry, we will go over it first. That is briefly defining important camera models and epipolar geometry. Next we will discuss structured light scanning, both the basics and the implementation of a popular line of techniques known as phase shifting. We do this as structured light has played a huge role in this thesis both in application and data collection. Finally we will end the chapter with an introduction to the problem of Non-Rigid Structure from Motion (NRSfM) and relevant theory.

2.1 Camera Geometry

Camera geometry is the theory behind image formation and how it relates to the 3D geometry of a given scene. This is understood as both the geometry of the scene itself and the camera's spatial properties. Camera geometry is the foundation of the vast majority of 3D estimation techniques such as stereo-vision, structured light, structure from motion and non-rigid structure from motion.

In this thesis, camera geometry is described in terms of projective models. That is, a model that can tell us a given 3D point's projection on the image plane. We will primarily be dealing with two camera models; orthographic and pinhole. The orthographic camera is arguably the simplest camera which finds usage in structure from motion and non-rigid structure from motion. Here, a 3D point $\mathbf{x}$ is projected along a line orthogonal to image plane I. Mathematically it is modeled as;

$$\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{x}, \tag{2.1}$$

where,

$\mathbf{R}$ = rotation matrix,
$\mathbf{t}$ = translation vector,
s = scale,
$\mathbf{u}$ = $\mathbf{x}$'s projection in homogenous coordinates.

As can be seen, an orthographic camera simply discards the z-coordinate during projection. This camera model is illustrated on the left side of Figure 2.1. While the orthographic camera provides a decent model of real world cameras for small volumes, it does not work well with large object or distances. The reason is that it does not model perspective foreshortening.

The pinhole camera model is a more accurate representation of real-world cameras. Here all projection lines must pass through a single point $\mathbf{f}$ called the focal point. Mathematically it is described as;

$$\mathbf{u} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}}_{\mathbf{E}} \mathbf{x}, \tag{2.2}$$

where,

$$\begin{aligned} \mathbf{R} &= \text{rotation matrix}, \\ \mathbf{t} &= \text{translation vector}, \\ \mathbf{u} &= \mathbf{x}\text{'s projection in homogenous coordinates}, \\ f_x, f_y &= \text{focal length}, \\ c_x, c_y &= \text{principal point}, \\ \mathbf{A} &= \text{intrinsic matrix}, \\ \mathbf{E} &= \text{extrinsic matrix}. \end{aligned}$$

Unlike the orthographic camera, the pinhole camera models perspective foreshortening. For this reason it is also sometimes referred to as a perspective camera. It can even be extended to model lens distortion [Dua71]. Overall it is a quite good representation of most real-world cameras despite leaving out concepts such as depth of field and focus. Parameters for the model in (2.2) for a given camera can be efficiently estimated calibrated using e.g. the method of [Zha00].

2.1.1 Epipolar Geometry

Neither of the above camera models are invertible, that is you cannot deduce a point's 3D position from it's projection onto the image. However 3D estimation is possible

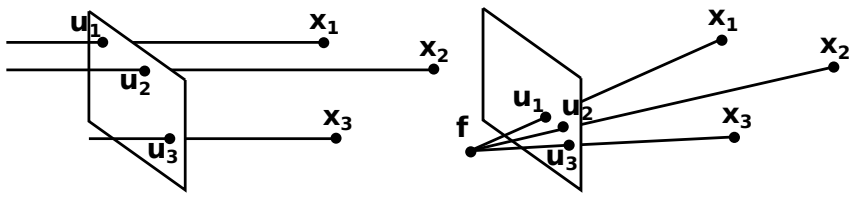


Figure 2.1: Illustration of orthographic camera to the left and pinhole to the right.

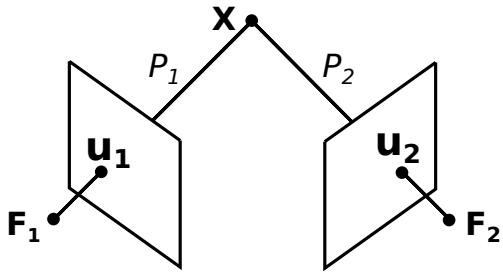


Figure 2.2: Geometry of a pair of pinhole cameras. $\mathbf{F}_1$ and $\mathbf{F}_2$ are the focal points, $\mathbf{u}_1$ and $\mathbf{u}_2$ are perspective projections of $\mathbf{X}$, and P_1 and P_2 are the projective lines

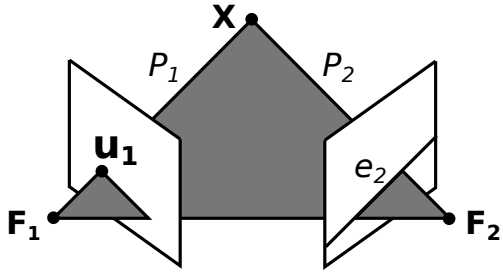


Figure 2.3: Epipolar geometry of the correspondence problem. The gray is the plane spanned by focal points $\mathbf{F}_1$ and $\mathbf{F}_2$ as well as $\mathbf{u}_2$ which is $\mathbf{X}$'s projection along P_1. This planes intersection with the image plane of $\mathbf{F}_2$ is the epipolar line e_2 on which $\mathbf{u}_2$ can be found

using two or more observations of the same point. Indeed, much of a human's sense of depth is a result of us having two eyes and can thus make two simultaneous observations. To understand how, we need to take a look at the geometry of a point and two pinhole cameras. Consider the notation of of Figure 2.2. Suppose we would like to estimate $\mathbf{X}$ and we know of $\mathbf{X}$'s projection onto both camera images; $\mathbf{u}_1$ and $\mathbf{u}_2$. Also suppose that we know the positions of each camera $\mathbf{F}_1$ and $\mathbf{F}_2$ along the corresponding intrinsics. With this information we can deduce the projective lines P_1 and P_2. We known that $\mathbf{X}$ must lie on both P_1 and P_2, therefore $\mathbf{X}$ must lie on the intersection between P_1 and P_2. This process of finding $\mathbf{X}$ using projective lines is known as **triangulation**.

In practice, the positions of $\mathbf{F}_1$ and $\mathbf{F}_2$ can be determined beforehand in a calibration step [Its15], but $\mathbf{u}_1$ and $\mathbf{u}_2$ must be deduce using the available image data. This is typically formulated as a pair assignment problem, meaning given some $\mathbf{u}_1$ we would like to deduce it's correspondence $\mathbf{u}_2$ in the other image. This is problem known as the correspondence problem. Fortunately, the camera geometry itself can be used to constraint the search space.

Let us say that we know $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{u}_1$ and we would like to find $\mathbf{u}_2$. Consider

the plane spanned by $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{u}_1$, we known that $\mathbf{X}$, P_2 and $\mathbf{u}_2$ must also lie in this plane. This plane is illustrated as the gray triangle shape in Figure 2.3. The intersection between this plane and the image plane of $\mathbf{F}_2$ gives us a line on which $\mathbf{u}_2$ lies. This is referred to as an **epipolar** line and is shown in Figure 2.3 as e_2. Thus epipolar geometry is used to constrain the search space for the correspondence problem with known camera geometry. For specifics, please consult [HZ04].

Even with the constraints of epipolar geometry, the correspondence problem can be quite hard to solve accurately for natural images due to weak or repeating texture. Structured light is an effective way of overcoming these limitation and often provides much more accurate 3D data than passive stereo.

2.2 Structured Light

Structured light is an active depth measurement technique, which eases the correspondence problem by projecting artificial texture onto the measured scene. The simplest example is merely projecting random noise, but typically the projected patterns are structured such that they directly encode positional information. Structured light can be implemented in many ways, but is typically achieved with a light projector and cameras, as was done in this thesis. The many structured light methods that have been developed over the years can be roughly grouped according to two factors;

Camera
: Single or Multi. Meaning whether a single or multiple cameras are used. For single camera the correspondence search is performed between the camera and the projector. For multi the correspondence search is performed between the cameras.

Pattern
: One-shot or multiplex. Meaning whether one or multiple patterns are projected onto the scene. The former is often used for real-time applications and the latter is often used for precision measurements.

Table 2.1 illustrates this taxonomy with examples. The general idea of multiplexing structured light is to compile the multiple patterns into a cohesive set of information that can be used to solve the correspondence problem. Formally let us say that we have a series of patterns $p = \{p_0 \dots p_{N-1}\}$ which we sequentially project

Table 2.1: Structured light taxonomy with cited examples for each entry.

	Oneshot	**Multiplex**
Single-Camera	Kinect V1	SLStudio [WOL14]
Multi-Camera	Assisted Stereo [Kon10]	SeemaLab [Eir+15]

and record. As such we obtain a series of images $i = \{i_0 \dots i_{N-1}\}$, which are to be used in solving the correspondence problem.

There are quite many ways of doing this like e.g. Gray Codes [PA82] or Unstructured Light [CMR11]. During this project, phase shifting was primarily used for it's precision and versatility. The following text will cover the base version of phase shifting, which will give an overview of how it operates.

2.2.1 Phase Shifting

As the name suggests, this category of structured light techniques seeks to encode a unique phase onto the scene. The pattern sequence is defined by the following,

$$p_n(u, v) = \frac{1}{2} + \frac{1}{2} \cos\left(2\pi\left(\frac{n}{N} + u\right)\right), \tag{2.3}$$

where,

$$\begin{aligned} u, v &= \text{normalized projector coordinates,} \\ N &= \text{sequence steps,} \\ n &= \text{pattern step.} \end{aligned}$$

Notice that v is not present in the pattern definition of (2.3). The reason for this is that patterns need only be horizontally unique because of the epipolar constraint. The values in the observed image then is,

$$i_n(x, y) = o(x, y) + a(x, y) \cos\left(\frac{n}{N} + \theta(x, y)\right), \tag{2.4}$$

where,

$$\begin{aligned} o(x, y) &= \text{background illumination,} \\ a(x, y) &= \text{albedo,} \\ \theta(x, y) &= \text{phase.} \end{aligned}$$

So for each pixel, we observed a sinusoidal signal that evolves over the sequence. We refer to this signal as $i(x, y) = \{i_0(x, y), \dots, i_{N-1}(x, y)\}$. The background illumination and albedo of the signal in (2.4) is not necessarily unique, however $\theta(x, y)$ is unique along epipolar lines. Therefore the goal of phase shifting is to recover $\theta(x, y)$ and use it for the correspondence problem. Indeed all of $\theta(x, y)$ is referred to as a phase map, and the process for creating one is illustrated in Figure 2.4. To properly recovering the phase, we apply the Discrete Fourier Transform (DFT) to this signal:

$$\begin{aligned} I(x, y) &= \mathcal{F}\left\{i(x, y)\right\}, \\ &= \{I_0(x, y), \dots, I_{N-1}(x, y)\} \end{aligned} \tag{2.5}$$

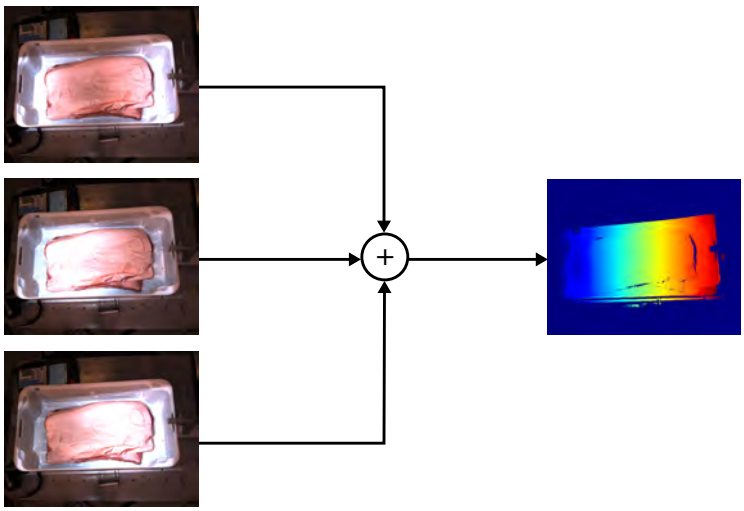


Figure 2.4: Illustration of single phase encoding. The signal defined by (2.3) is projected onto the scene. The observed images are then combined into a single phase-map, which defines $\theta(x, y)$, using (2.5) and (2.9)

where,

$$I_n(x, y) = \text{Fourier components of } i(x, y).$$

Ideally, the resulting spectrum should be,

$$I_0(x, y) = o(x, y) \tag{2.6}$$

$$I_1(x, y) = Na(x, y)e^{i\theta(x,y)} \tag{2.7}$$

$$I_2(x, y) = \ldots = I_{N-1}(x, y) = 0 \tag{2.8}$$

With this, we obtain the phase by,

$$\theta(x, y) = \arg\left(I_1(x, y)\right) \tag{2.9}$$

In practice $I_2(x, y), \ldots, I_{N-1}(x, y)$ is not exactly zero due to sensor noise and slight variations in background illumination. In fact some of this noise spectrum will likely spill over into $I_1(x, y)$ as dictated by the Nyquist-Shannon sampling theorem. Luckily, this can be mitigated by simply adding more samples to our signal. Thus, phase shifting is quite scalable requiring a minimum of 3 patterns with the option of adding more patterns for more precision.

Typically only real-time applications goes for this base pattern approach. The reason being that there are many external sources of noise (e.g. signal discretization, unstable background illumination and sensor noise) that sets an effective limit to how

accurate we can know $\theta(x, y)$. Instead, it is more accurate to observe multiple phases with the above method and combine them. One of the most common ways of doing this, and the technique used in this project, is called phase unwrapping.

2.2.1.1 Phase Unwrapping

The use of phase unwrapping requires that one projects two sinusoidal signals onto the measured scene, one with 1-period and one with K-periods. The idea is then to combine the phase of each into a single, accurate phase map. This process is illustrated in Figure 2.5. The 1-period signal is defined in (2.3) which we refer to this pattern sequence as $p = \{p_0, \ldots, p_{N-1}\}$. Let us then denote the K-spatial period signal as $p_K = \{p_{K,0}, \ldots, p_{K,N-1}\}$ with each pattern being defined by,

$$p_{K,n}(u, v) = \frac{1}{2} + \frac{1}{2} \cos\left(2\pi \left(\frac{n}{N} + uK\right)\right), \tag{2.10}$$

where,

$$\begin{aligned} u, v &= \text{normalized projector coordinates}, \\ N &= \text{sequence steps}, \\ n &= \text{pattern step}, \\ K &= \text{number of spatial periods}. \end{aligned}$$

In the above, the phase lies in the range $[0, 2K\pi]$. However, the observed phase $\theta_K(x, y)$ is still the same (2.4) and thus lies in the range $[0, 2\pi]$. Let us refer to the true phase as $\phi_K(x, y)$ which is given by,

$$\phi_K(x, y) = \theta_K(x, y) + 2\pi k(x, y), \tag{2.11}$$

where,

$$k(x, y) = \text{some integer giving the period},$$

Phase unwrapping is basically the problem of determining $k(x, y)$ which is the purpose of p. For this reason, p is typically referred to as the unwrapper. As p only has one spatial period, it's phase is unambiguous and can thus be used to determine $k(x, y)$. This is done by dividing the range of $\theta(x, y)$ into K parts and then finding the right one. In other words,

$$k(x, y) = \text{round}\left(\frac{K\theta(x, y) - \theta_K(x, y)}{2\pi}\right). \tag{2.12}$$

By using the results of (2.12) in (2.11), we can recover $\phi_K(x, y)$.

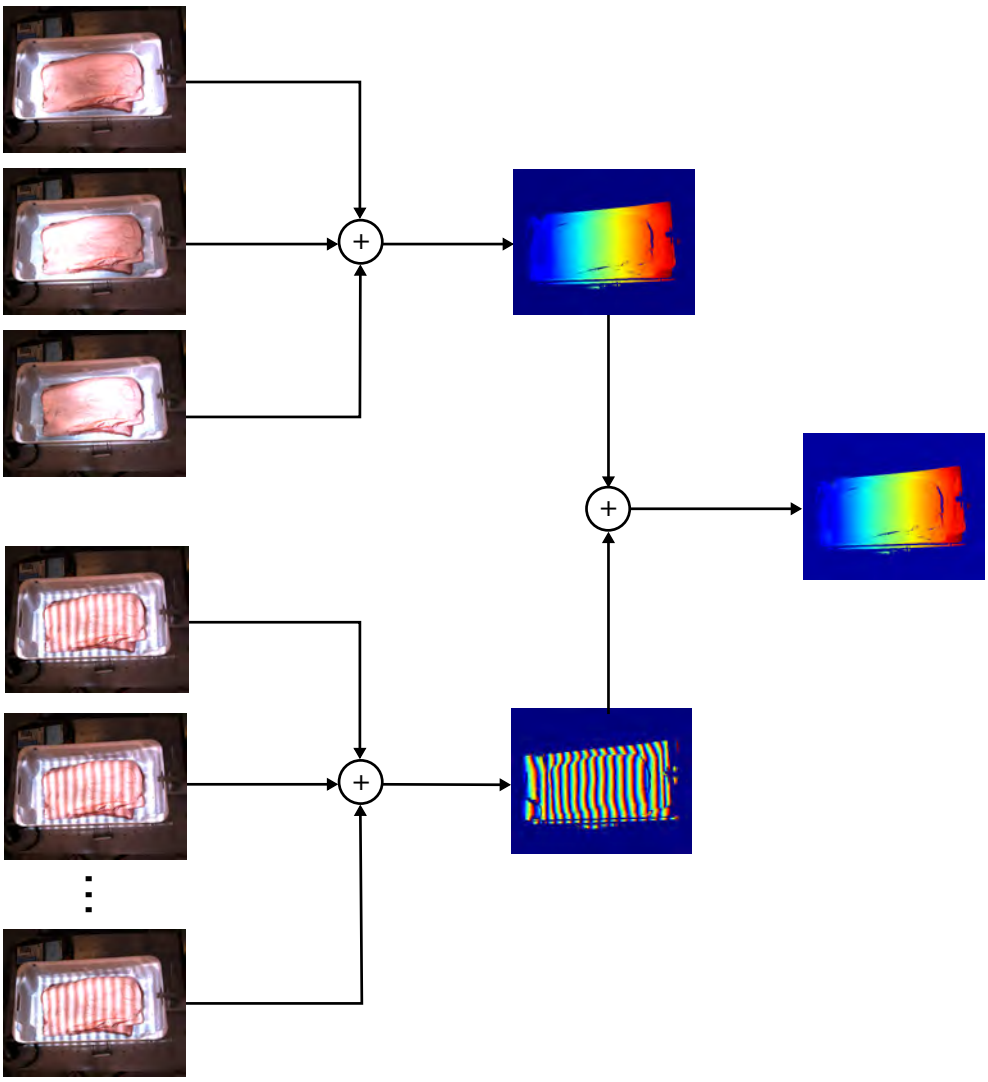


Figure 2.5: Illustration of the encoding process of unwrapping phase shifting. First, two sequences of patterns projected onto the scene are recorded. Then, encoded into two phase-maps, $\theta(x, y)$ and $\theta_K(x, y)$, respectively at the top and at the bottom. Finally they are encoded into a complete phase map via (2.11) and(2.12)

2.2.1.2 Advanced Techniques

The aforementioned technique is very accurate, but also makes critical assumption; the observed signal is only the result of the direct reflection of the projected patterns. This assumption is often broken in two ways in the real world. First, if the background illumination is not constant, it will contaminate the signal spectrum. Second, is if a spatial echo of the signal is measured along with the primary reflection. This is typically the result of either inter-reflections or subsurface scattering. The former refers to light reflected from one part of the scene to another, the latter refers to light being scattered beneath an object's surface. Subsurface scattering can in particular introduce a subtle bias in measured depth, as we have shown in one of the studies included in this thesis [JWA17].

Various solutions to these problems have been proposed. Authors of [GN12] points out that the effect of inter-reflection dependent on the spatial frequency of the projected patterns. They also conclude that lower frequencies are more affected than higher frequencies. Thus, they propose micro phase shifting, where only a narrow band of high-frequent patterns are used. This minimizes the effect of inter-reflections by ensuring that it is approximately the same for all patterns. Modulated phase shifting follows a similar strategy, wrapping the signal in a high frequency carrier wave [CSL08].

2.3 Non-Rigid Structure from Motion

NRSfM is, as Figure 2.6 illustrates, the science of estimating geometry from a set of 2D observations, that is both view and scene geometry. Unlike regular Structure from Motion (SfM), NRSfM makes no rigidity assumption, which makes a much broader and a much harder problem. To see why, let us take a look at the classical SfM factorization problem formulated by Tomasi et al. [TK92],

$$\mathbf{W} = \mathbf{MS}, \tag{2.13}$$

where,

$$\mathbf{W} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1P} \\ v_{11} & v_{12} & \cdots & v_{1P} \\ u_{21} & u_{22} & \cdots & u_{2P} \\ v_{21} & v_{22} & \cdots & v_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ u_{F1} & u_{F2} & \cdots & u_{FP} \\ v_{F1} & v_{F2} & \cdots & v_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} & \cdots & \mathbf{w}_{1P} \\ \mathbf{w}_{21} & \mathbf{w}_{22} & \cdots & \mathbf{w}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{F1} & \mathbf{w}_{F2} & \cdots & \mathbf{w}_{FP} \end{bmatrix},$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{M}_F \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,P} \\ y_{1,1} & y_{1,2} & \cdots & y_{1,P} \\ z_{1,1} & z_{1,2} & \cdots & z_{1,P} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,P} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,P} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{F,1} & x_{F,2} & \cdots & x_{F,P} \\ y_{F,1} & y_{F,2} & \cdots & y_{F,P} \\ z_{F,1} & z_{F,2} & \cdots & z_{F,P} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{11} & \mathbf{s}_{12} & \cdots & \mathbf{s}_{1P} \\ \mathbf{s}_{21} & \mathbf{s}_{22} & \cdots & \mathbf{s}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{F1} & \mathbf{s}_{F2} & \cdots & \mathbf{s}_{FP} \end{bmatrix},$$

$\mathbf{M}_f =$ 2x3 orthographic projection matrix,

$F =$ number of frames,

$P =$ number of points.

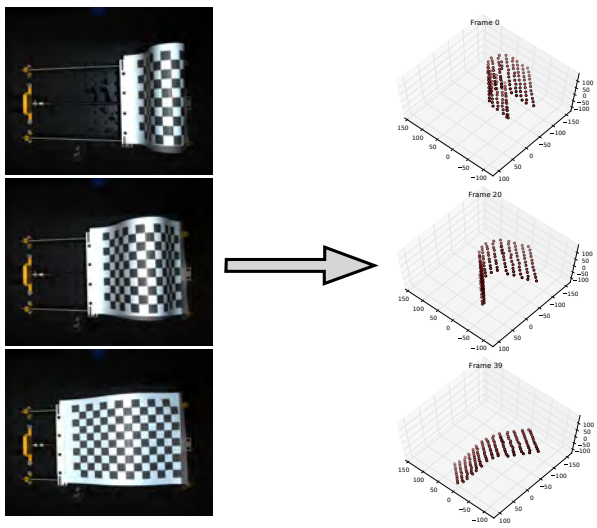


Figure 2.6: Illustration of NRSfM. The general idea is to take a set of 2D observations (typically from images) and produce an estimate of the scene and view geometry. The reconstructions on the right have been created using the algorithm described in [KDL17a]

In other words, $\mathbf{W}$ is the observation matrix, $\mathbf{M}$ is the motion matrix and $\mathbf{S}$ is the shape matrix. For SfM we have the following constraint,

$$x_{1,p} = x_{2,p} = \cdots = x_{F,p}, \tag{2.14}$$

$$y_{1,p} = y_{2,p} = \cdots = y_{F,p}, \tag{2.15}$$

$$z_{1,p} = z_{2,p} = \cdots = z_{F,p}. \tag{2.16}$$

As such, we know that $\text{rank}(\mathbf{S}) \leq 3$. For NRSfM we do not make any such assumptions. This means that the factorization problem in (2.13) becomes highly ill-posed, as a non-singular corrective transform $\mathbf{G}$ can be applied to arrive at a different valid factorization,

$$\mathbf{W} = \mathbf{M}\mathbf{G}\mathbf{G}^{-1}\mathbf{S} \tag{2.17}$$

In order to solve (2.13) one needs to add the appropriate priors and regularization. While SfM is a quite mature field, NRSfM still remains a largely unsolved problem. There is not as of yet a clear consensus of the best approach. However, many methods follow the same overall strategy, that is using a low-rank basis.

2.3.1 Low-Rank Basis

This prior was first proposed by Bregler et al. [BHB00]. It assumes that the shape in each frame can be modeled as the linear combination of a set of basis shapes of some rank K. As such the general problem of (2.13) becomes,

$$\mathbf{W} = \underbrace{\mathbf{D}(\mathbf{C} \otimes \mathbf{I}_3)}_{\mathbf{M}} \underbrace{\begin{bmatrix} \hat{\mathbf{S}}_1 \\ \hat{\mathbf{S}}_2 \\ \vdots \\ \hat{\mathbf{S}}_K \end{bmatrix}}_{\mathbf{S}} \tag{2.18}$$

where,

$$\mathbf{D} = \begin{bmatrix} \hat{\mathbf{R}}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \hat{\mathbf{R}}_2 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,K} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ c_{F,1} & c_{F,2} & \cdots & c_{F,K} \end{bmatrix}$$

This particular formulation of the problem has become quite popular due to it's simplicity and expressive power, see [GM11a], [GM11b], [GM11c] or [HGM12] for examples. While the formulation in (2.18) is much more constrained than the base formulation in (2.13), there still quite many unknowns to be determined. Furthermore low-rank shapes only ensures spatial smoothness, but neglects temporal smoothness.

For this reason the idea of using a Discrete Cosine Transform (DCT) basis was introduced. Akhter et al. [Akh+08] first used the DCT as a shape basis, however Gotardo et al. [GM11c] later proposed to use the DCT basis to model the weight matrix $\mathbf{C}$ instead,

$$\mathbf{C} = \Omega_d \begin{bmatrix} x_1 & x_2 & \cdots & x_K \end{bmatrix} = \Omega_d \mathbf{X} \tag{2.19}$$

where,

$$\begin{aligned} \Omega_d &= \text{DCT basis with } d \text{ components,} \\ x_k &= \text{DCT coefficient.} \end{aligned}$$

Such that (2.18) becomes,

$$\mathbf{W} = \underbrace{\mathbf{D}(\Omega_d \mathbf{X} \otimes \mathbf{I}_3)}_{\mathbf{M}} \underbrace{\begin{bmatrix} \hat{\mathbf{S}}_1 \\ \hat{\mathbf{S}}_2 \\ \vdots \\ \hat{\mathbf{S}}_K \end{bmatrix}}_{\hat{\mathbf{S}}} \tag{2.20}$$

This ensures temporal smoothness for both shape deformation and for the camera as well. Gotardo et al. [GM11c] and Ansari et al. [DGS17] suggest first estimating $\mathbf{D}$ and $\mathbf{X}$, then $\hat{\mathbf{S}}$ can be found by,

$$\hat{\mathbf{S}} = \mathbf{M}^{+}\mathbf{W}, \tag{2.21}$$

where,

$$\mathbf{M}^{+} = \text{Moore-Penrose pseudoinverse of } \mathbf{M}.$$

2.3.2 Missing Data

The base formulation of the NRSfM problem in (2.13) implicitly assumes that the position of all points are known in all frames. However, in the real-world this is rarely the case due to self-occlusion and occlusions from the environment. This adds additional complexity on top of an already difficult problem. Most attempt to deal with this problem as a matrix completion problem, that is estimating the missing entries in $\mathbf{W}$. This can be accomplished using either a DCT-basis [GM11c] or repeated factorizations [Pal+09]. This described in detail in Section 3.1.1.

2.4 Conclusion

In this chapter, we have gone through some of the background theory of this thesis. We have seen that image formation can be modeled using either orthographic or

perspective projection. We have also seen how these models can be used to recover the 3D position of a point from multiple camera observations.

Structured light has been defined as a method for solving the correspondence problem inherent to multi-view geometry estimation. We have shown how this may be implemented by gradually shifting a sinusoidal pattern over the scene, which encodes a horizontal phase. This technique is what is known as phase-shifting.

Finally, SfM was defined as the problem of recovering camera pose and scene geometry from a series of monocular images. NRSfM was defined as when the observed scene is not static, but changes over the course of the series. We have shown why this problem is inherently ill-posed and some common priors used to constrain the solution space.

CHAPTER 3

Related Work

In this chapter, we will go through the relevant body of research related to the work done in this thesis. The work done in Non-Rigid Structure from Motion (NRSfM) and flexible robotics is considered the main contribution of this thesis, and will thus be the focal point of this section. For literature related to the other works included in this thesis, we refer to the respective publications themselves in the appendices.

First, we will go over the recent literature in NRSfM, which solutions have been proposed and how they relate to eachother. We will go over what types of priors have been employed and how the camera has been modeled. Then, we will examine how missing data has been handled in past work.

Secondly, we will examine the literature relevant to the flexible robotics problem. As problem is picking piece of meat from an unorganized pile, we consider the problem as bin-picking of non-rigid objects. To our knowledge, there is little work done on this exact topic. As such we will divide our attention to two closely related fields; bin-picking and instance segmentation.

3.1 Non-Rigid Structure from Motion

As briefly mentioned in section 2.3, NRSfM is still a field undergoing significant development. In the following, we will give a review of related work in NRSfM. It should be seen as expanding upon the existing literature review of paper A. We will first focus on the shape and motion recovering aspects of NRSfM and how the idea of low-rank shapes has been implemented in previous work. The base NRSfM factorization problem assumes an orthographic camera. However, most real-world image data is obtained via perspective projection. Thus, we will examine how this gap has been bridged in previous work. As previously mentioned, much of field assumes a complete $\mathbf{W}$ which is rarely the case in real-world data. Thus, a portion of this text will be dedicated to examining how missing data has been dealt with previously.

NRSfM is an inherently ill-posed problem and thus a solution needs regularization and priors to be physically meaningful. Despite the physical nature of the problem, purely statistical priors are often employed. Methods that use the low-rank basis prior are a part of this category, as are methods which employ spatio-temporal smoothness as well as orthonormality.

Bregler et al. [BHB00] was the first to employ a low-rank statistical prior, inspired by Tomasi et al.'s work in Structure from Motion (SfM) [TK92]. Indeed, they also recover the camera motion and base shapes using the Singular Value Decomposition (SVD),

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{3.1}$$

$$= \hat{\mathbf{M}}\hat{\mathbf{S}} \tag{3.2}$$

where,

$$\hat{\mathbf{M}} = \mathbf{U}\mathbf{\Sigma}$$
$$\hat{\mathbf{S}} = \mathbf{V}^T$$

and simply taking the $3K$ largest singular vectors and values. While reasonable results were achieved with the above factorization method, it only acts as a constraint on the spatial distribution of the reconstruction. So for a rapidly moving camera, results can quickly deteriorate. Akhter et al. [Akh+08] proposed that a low frequency Discrete Cosine Transform (DCT) should be used as shape basis instead,

$$\mathbf{W} = \mathbf{D}\mathbf{\Omega_d}\mathbf{S}. \tag{3.3}$$

where,

$$\begin{aligned} \mathbf{D} &= \text{trace orthographic projection matrix,} \\ \mathbf{\Omega_d} &= \text{DCT basis with } d \text{ vectors,} \\ \mathbf{S} &= \text{shape bases.} \end{aligned}$$

They derive this form by applying a rectification matrix $\mathbf{Q}$ to (3.2),

$$\mathbf{D}\mathbf{\Omega_d} = \hat{\mathbf{M}}\mathbf{Q} \tag{3.4}$$

$$\mathbf{S} = \mathbf{Q}^{-1}\hat{\mathbf{S}}. \tag{3.5}$$

$\mathbf{Q}$ is found using the DCT basis. This dual representation is referred to as the point trajectory approach. Gotardo et al. [GM11c] later argue that the trajectory approach could be expanded to the entire shape, viewing the deformation as a smooth point trajectory in K dimensional space,

$$\mathbf{W} = \underbrace{\mathbf{D}(\Omega_d\mathbf{X} \otimes \mathbf{I}_3)}_{\mathbf{M}} \underbrace{\begin{bmatrix} \hat{\mathbf{S}}_1 \\ \hat{\mathbf{S}}_2 \\ \vdots \\ \hat{\mathbf{S}}_K \end{bmatrix}}_{\hat{\mathbf{S}}} \tag{3.6}$$

The motion matrix $\mathbf{M}$ can then effectively be estimated using their column space fitting algorithm [GM11a]. The shape basis is then determined by,

$$\hat{\mathbf{S}} = \mathbf{M}^{+}\mathbf{W}, \tag{3.7}$$

where,

$$\mathbf{M}^{+} = \text{Moore-Penrose pseudoinverse of } \mathbf{M}.$$

This approach was later expanded with the kernel trick, to provide effective means of modeling non-linear deformations like articulated motion [GM11b]. Torresani et al. [THB08] points out that the linear subspace representation of NRSfM in (2.18) is inherently quite unstable w.r.t. the size of the subspace. Choose a K that is too large and the problem becomes totally unconstrained, choose a K that is too small and the problem becomes too constrained, unable to accurately model real world motion. Thus, making these methods work would require extensive parameter tuning. They argue that treating the shape estimation as a probabilistic problem is a better approach. Specifically that the shape weights of $\mathbf{C}$ in (2.18) should be viewed as a normal distribution,

$$c_{f,k} = \mathcal{N}(0, 1) \tag{3.8}$$

Then $\mathbf{S}$ can be found via an Expectation-Maximization (EM) algorithm. Olsen et al. [OB08] included both temporal and spatial smoothness into their NRSfM algorithm by including corresponding penalty terms into an optimization step. Olsen et al. [Bar+08] combined this approach with another key assumption; that base shapes S_k are ordered in a coarse to fine manner similar to the components of Principal Component Analysis (PCA). This means that the first base shapes describes the coarse movements while the later shapes describes finer motion. In their algorithm, shapes are estimated in an iterative manner, allowing for automatic selection of rank K to a certain error threshold. Each shape is determined in an optimization step with spatial and temporal smoothness terms. In a similar spirit, Brandt et al. [Bra+09] argues that the best way to select the shape base is via statistical independence. They proposed using the independent component analysis to accomplish this.

Kong et al. [KL16] argues that the linear combination of a number of shapes is too restrictive to express generic deformations. Instead, one should exploit the inherent compressibility of SfM to enforce sparsity in the shape basis. As such we can assume a full rank shape basis $\mathbf{S}$ which is sparse in the sense that it only has K non-zero entries for each row. Kong et al. [KL16] showed that this prior is strong enough to yield decent reconstructions without any additional priors like smoothness.

A low-rank basis implicitly assumes that only one deforming object is present. Kumar et al. [KDL17b] argues that this is too inflexible to handle real-world scenes, where more than one object is typically present. Instead, they proposed formulating the reconstruction problem as a joint segmentation and reconstruction problem. Indeed, they argue this can be done by exploiting the inherent spatial and temporal

clustering of NRSfM. This is formulated as a self-expressive property of $\mathbf{S}$. For spatial clustering this is given by,

$$\mathbf{S} = \mathbf{S}\mathbf{C}_1, \tag{3.9}$$

subject to,

$$\text{diag}(\mathbf{C}_1) = 0, \tag{3.10}$$

$$\mathbf{1}^T\mathbf{C} = \mathbf{1}. \tag{3.11}$$

Here, some clustering matrix $\mathbf{C}_1$ approximates each column (which is a trajectory) in $\mathbf{S}$ as a linear combination of other columns in $\mathbf{S}$. Constraint (3.10) ensures that we avoid the obvious solution of $\mathbf{C}_1 = \mathbf{I}$ and (3.11) ensures that the combination remains affine. A similar self-expressive constraint can be formulated for trajectories [KDL17b]. Kumar et al. then shows that an accurate reconstruction can be estimated in solving for the spatio-temporal clustering. While this approach has been designed for multiple bodies, it also works quite will for single bodies as we have shown with our NRSfM dataset and evaluation (paper A).

All of the above methods deal with an orthographic camera, however most real world image data is best approximated with a perspective camera. A perspective projection $\mathbf{q}$ of a point $\mathbf{s}$ is given by,

$$\lambda\mathbf{q} = \mathbf{P}\mathbf{s} \tag{3.12}$$

where,

$$\mathbf{s} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \mathbf{q} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix},$$

$\mathbf{P} = 3\text{x}4$ perspective projection matrix,

$\lambda =$ projective depth.

In other words, perspective projection can be seen as a scaled affine projection. For this reason many NRSfM and SfM methods account for perspective projection by considering a scaled measurement matrix $\widetilde{\mathbf{W}}$ instead of the original,

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \lambda_{11}\mathbf{q}_{11} & \lambda_{12}\mathbf{q}_{12} & \cdots & \lambda_{1P}\mathbf{q}_{1P} \\ \lambda_{21}\mathbf{q}_{21} & \lambda_{22}\mathbf{q}_{22} & \cdots & \lambda_{2P}\mathbf{q}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{F1}\mathbf{q}_{F1} & \lambda_{F2}\mathbf{q}_{F2} & \cdots & \lambda_{FP}\mathbf{q}_{FP} \end{bmatrix} \tag{3.13}$$

where,

$$\begin{aligned}\mathbf{q}_{fp} &= \begin{bmatrix}\mathbf{w}_{fp}\\ 1\end{bmatrix},\\ \mathbf{w}_{fp} &= \text{ image coordinate as given by (2.13)},\\ \lambda_{fp} &= \text{ projective depth.}\end{aligned}$$

Estimating the projective depths is, of course, not a trivial task, especially for non-rigid scenes. Wang et al. [WTH07] proposed a solution where the observation matrix $\mathbf{W}$ is iteratively reweighted. The new weights are calculated from the estimated factorization $\mathbf{M}$'s and $\mathbf{S}$'s deviation from the observed perspective projection.

Hartley and Vidal [HV08] derived a closed-form algrebraic solution for recovering the perspective projection matrices along with the shape basis and shape weights. The algorithm simply requires an initial estimate of a multifocal tensor, though the authors have reported it as being very noise sensitive.

Llado et al. [LDA10] proposed a method for segmenting a deforming scene into rigid and non-rigid points using an initial rough projective depth estimation. The camera is then self-calibrated using the rigid subset, which is then used to refine the perspective factorization problem.

Chhatkuli et al. [Chh+17] completely forgoes the standard factorization procedure in dealing with perspective. Instead, they resolve the projection depth in second-order cone programming formulation by representing shape as a set of view invariant features. Specifically, they use the assumption of isometry in the form of a maximum depth heuristic.

3.1.1 Missing Data

In real world observations $\mathbf{W}$ is rarely complete due to occlusions. As such dealing with this missing data is essential for NRSfM. The BALM algorithm [Del+12] treats shape, motion and missing data filling as a joint iterative optimization problem. Consider that we want to estimate to factorization $\mathbf{W} = \mathbf{MS}$, but only some values of $\mathbf{W}$ are known. Let this set of known values be the set $O = \{(i,j) : \mathbf{W}_{i,j} \text{ is known}\}$. Now instead of optimizing on $\mathbf{W}$, we instead optimize on a function $Y(\mathbf{Z})$ which fills $\mathbf{W}$'s unknown entries with estimates $\mathbf{Z}$. In other words, $Y(\mathbf{Z})$ is defines as,

$$Y(\mathbf{W},\mathbf{Z})_{ij} = \begin{cases}\mathbf{w}_{ij}, & \text{if}(i,j) \in O\\ \mathbf{z}_{ij}, & \text{otherwise}\end{cases} \tag{3.14}$$

Then we want to optimize on the following loss function,

$$L(\mathbf{W},\mathbf{Z},\mathbf{S},\mathbf{M};\lambda) = ||Y(\mathbf{W},\mathbf{Z}) - \mathbf{MS}||^2 + \Lambda(\mathbf{M},\lambda), \tag{3.15}$$

where,

$$\Lambda = \text{Lagrangian constraint function},$$
$$\lambda = \text{Lagrange Multipliers}.$$

Note that (3.15) is a simplified version of the one found in [Del+12]. Specifically, it excludes a manifold projection penalty term as it is not important for understanding how the algorithm deals with missing data. $\mathbf{M}$, $\mathbf{S}$ and $\mathbf{Z}$ are then determined by iteratively solving (3.16) and (3.17),

$$(\mathbf{S}^{k+1}, \mathbf{M}^{k+1}) = \underset{\mathbf{S},\mathbf{M}}{\operatorname{argmin}} L(\mathbf{W}, \mathbf{Z}^k, \mathbf{S}, \mathbf{M}; \lambda), \tag{3.16}$$

$$\mathbf{Z}^{k+1} = \underset{\mathbf{Z}}{\operatorname{argmin}} L(\mathbf{W}, \mathbf{Z}, \mathbf{S}^{k+1}, \mathbf{M}^{k+1}; \lambda), \tag{3.17}$$

where $\mathbf{S}^k, \mathbf{M}^k, \mathbf{Z}^k$ denotes the results of iteration k. In other words $\mathbf{Z}$ is estimated, based on the best estimates of $\mathbf{S}$ and $\mathbf{M}$ and vice versa.

Paladini et al. [Pal+09] follows a similar iterative missing data estimation approach. Indeed, theirs is quite generic, requiring only an initial estimate of a filled $\mathbf{W}$ via $\mathbf{Z}$. The complete procedure is described in Algorithm 1.

Algorithm 1: Iterative factorization and missing data estimation algorithm [Pal+09]. Note that $E[*]$ denotes the expectation (or mean) operator. $Y(\mathbf{W}, \mathbf{Z}^{[k]})$ is as defined in (3.14). Runs for K iterations or until convergence.

for $k \in K$ **do**
- Fill missing entries: $\mathbf{Y}^{[k]} = Y(\mathbf{W}, \mathbf{Z}^{[k]})$
- Estimate centroid: $\mathbf{t}^{[k]} = \left[E\left[\mathbf{y}_{1*}^{[k]}\right] \quad E\left[\mathbf{y}_{2*}^{[k]}\right] \quad \cdots \quad E\left[\mathbf{y}_{F*}^{[k]}\right]\right]^{\mathrm{T}}$
- Remove centroid: $\hat{\mathbf{Y}}^{[k]} = \mathbf{Y}^{[k]} - \left[\mathbf{t}^{[k]} \quad \mathbf{t}^{[k]} \quad \cdots \quad \mathbf{t}^{[k]}\right]$
- Solve NRSfM factorization: $\hat{\mathbf{Y}}^{[k]} = \mathbf{M}^{[k]}\mathbf{S}^{[k]}$
- Add centroid: $\mathbf{Z}^{[k+1]} = \mathbf{M}^{[k]}\mathbf{S}^{[k]} + \left[\mathbf{t}^{[k]} \quad \mathbf{t}^{[k]} \quad \cdots \quad \mathbf{t}^{[k]}\right]$

end

Another strategy for dealing with missing data is to fill the missing entries in $\mathbf{W}$ before applying a NRSfM algorithm, as was done in [GM11c] and [GM11a]. Indeed, they assume that, similar to their camera trajectory, that each projected point trajectory can be expressed in terms of a low frequency DCT basis. With this assumption they recover an initial $\mathbf{M}$ and $\mathbf{S}$ from the known entries, which is used to fill the missing entries in $\mathbf{W}$.

Chhatkuli et al. [Chh+17] poses the NRSfM reconstruction problem as an optimization problem. As such they simply do not include the missing terms in their optimization algorithm.

3.2 Bin-Picking of Non-Rigid Objects

As stated in the start of this chapter, our study of flexible robotics is related to two fields of research. The first being bin-picking which is the task of picking separate object from a clutter (typical in a bin). Second being instance-level segmentation which is the unique segmentation of several object instances. We will go over both fields in teh following

3.2.1 Bin Picking

This task has been around for many years and still remains an active field of research. The archetype of this task is picking out objects of known geometry from a clutter, typically in the form of a Computer Assisted Design (CAD) model. Mahler and Goldberg [MG17] proposed a transferred deep learning solution to the bin picking problem. To be specific, they used a Convolutional Neural Network (CNN) as a regressor to a partially observed Markov decision process formulation of the bin picking process. The network is trained using a collecting of synthetic depth maps generated using a variety of CAD models in various poses.

Kim et al. [Kim+16] uses a cascaded kernel convolution score with Haar-like filers to robustly detect object poses in a cluttered pile. Indeed, they show their method to be robust even under severe specular reflections. However, the method is specifically designed to work with approximately planar objects, which are assumed to be parallel with bottom of their container.

Zeng et al. [Zen+17] successfully applied a CNN for object classification and detection to RGB-D data. Object pose was deduced afterwards by fitting the appropriate CAD model to the measured point cloud using fine-grained Iterative Closest Point (ICP). This information was then used to guide a 6-DoF robot arm with a claw gripper.

Chang and Wu [CW14] proposed a more traditional approach for detecting and pose estimating cluttered rigid bodies. Using a structured light scanner, they obtain the surface geometry of the object pile. Object instances are then simultaneously detected and pose estimated via registering a CAD model onto the point cloud.

Wang et al. [Wan+17] argued for a strategy based on point pair features for handling cluttered texture-less objects. Using a voting framework and PCA they implemented a clustering technique based these features, which can retrieve the pose of a known object in a cluttered scene. Their voting scheme also allows for mismatch detection.

Indeed, point pair features are quite commonly used for pose estimation of known cluttered objects as Abelloos and Goedemé [AG16] point out in their review of the field. However, they argue that point pair features can be prohibitively slow when trying to register to a large point cloud. Thus they purpose a search heuristic, based iterative highest point detection.

Ellekilde et al. [Ell+12] models the bin picking problem as a search through a probability space. This space can be efficiently searched using either a weighted

random selection scheme or, if samples are sparse, a priority based scheme. A camera is used for input and for learning correction.

3.2.2 Instance-Level Segmentation

Segmentation is the task of separating one or more objects from the background in an image. This problem has been relevant since the inception of computer vision and image analysis, and thus has been and is an active research field. Needless to say, this is also quite relevant for vision guided bin-picking. However, ordinary segmentation is basically a classification process that assigns a class to each pixel (e.g. foreground/background). This is insufficient for the bin-picking problem, as we need to know the location of each instance of said class (e.g. where is each screw in this pile). Instance segmentation is an extension of the segmentation problem, where you seek to assign a unique label for each object instance.

One of the most common approaches is segmentation by detection, where objects are first localized on a bounding box level and then segmented within each box. Kumar et al. [KTZ05] proposed an Markov Random Field (MRF) based implementation of this strategy with category specific shape priors. Similarly Yao et al. [YFU12] used shape priors in combination with a holistic Conditional Random Field (CRF) to formulate detection and segmentation as a joint problem. Riemenschneider et al. [Rie+12] proposed a two step process in which object centers are located via a voting scheme in a Hough graph. Fine-grained segmentation is then achieved via back-projection onto a CRF. The watershed transform is a traditional technique for instance-level segmentation in image analysis, but it requires a well-defined potential field to yield good results. Thus, it is rarely used on complex scenery. Bai and Urtasun [BU17] proposed using a CNN to transform an image into an instance semantic energy field, where the center of each object corresponds to a well-defined local minimum. Then, watershed transform can be used to achieve accurate instance-level segmentation.

Instance-level segmentation is quite a daunting task, thus human-in-the-loop segmentation strategies has been often been applied. This is typically in the form of initialization data, such as a bounding box or a mask. The quintessential example is probably the GrabCut algorithm [PMC10]. Here, a user supplied mask is used to derive a color-based Gaussian Mixture Model (GMM) model for background and object, which is later used in a MRF formulation of the segmentation problem. The latter is solved via a graph cut. The OSVOS [Cae+17] and OnAVOS [VL17] utilizes a similar approach, though here a user supplied mask is used to refine a CNN for instance tracking and segmentation.

Another approach is segmentation via registration of a known CAD model. This prominent strategy in the field of bin packing, as shown Section 3.2.1. For example Zeng et al. [Zen+17] used a CNN to detect object instances in an RGB-D image. Then the corresponding CAD model was registered onto the scene to obtain exact location and orientation.

Newer approaches seeks exploit the power of deep learning to solve instance segmentation without the need for object detection or shape priors. Zhang et al. [ZFU16] proposed merging local CNN label predictions into a global MRF energy problem along with smoothness and inter-connectedness priors. The maximum expectation solution is then obtained using mean fields. Liang et al. [Lia+15] prosed training a CNN for predicting the bounding boxes and number of instances. The information is then used in a simple clustering scheme to obtain the final instance-level segmentation.

3.3 Conclusion

In this chapter, we reviewed the related work in two fields; NRSfM and bin-picking of deformable objects.

As we have shown, there is quite an impressive and diverse body of literature on NRSfM. We have seen how statistical priors like low-rank and/or DCT basis have been used to constraint the solution spaces. Thus, enabling one to find both spatially and temporally smooth solutions. Other priors have also been efficiently applied such as spatio-temporal clustering or isometry. We have also seen that the orthographic camera model is still widely used, despite efforts to employ a perspective model. Missing data remains a challenging issue with many suggested solutions, though mostly modeled as a matrix completion problem. Some performs an initial fill-in and then proceeds to the ordinary NRSfM algorithm whereas others employ an iterative fill-in factorization approach.

As shown in this chapter, there are many proposed solutions to the problem of bin-picking and instance segmentation. Some formulate the problem as finding the EM solution to a MRF or CRF, others leverage the power of deep-learning and CNNs to achieve high quality segmentation. For bin-picking prior known CAD models are still widely used for detection and pose estimation, which implies a rigidity prior.

CHAPTER 4

Contributions

In this chapter, we go over the contributions made, in this thesis, in fulfilment of the objectives specified in Section 1.2. We will first discuss our evaluation of the field of Non-Rigid Structure from Motion (NRSfM) (paper A). This will cover both the creation of a new NRSfM dataset as well as our factorial analysis of the field. Then, we shall discuss the flexible robotics cell that was developed during our studies (paper B). We will focus both on the implementation and on the lessons learned. In continuation of this project, we shall move on to our study of structured light scanning of biological material (paper C). This is effectively also a study on the effects of subsurface scattering on structured light scanning. Finally, we will examine the work done on the creation of a rendering dataset as well as traceable vision in geometric metrology. In the end, we will reflect on the lessons learned and provide perspective on future avenues of research.

4.1 Evaluation of Non-Rigid Structure from Motion

As shown in Section 3.1, many NRSfM methods have been proposed. So many, that it is unclear what the field can do and where the challenges are. Thus, in the following, we will go over this thesis's evaluation of NRSfM. This is divided into two parts; the creation of a proper dataset and a factorial analysis. Our work also provides a basis for future evaluation and studies.

4.1.1 Dataset

In our opinion, the lack of coalescence in NRSfM is largely due to a lack of interesting, realistic data with ground truth. Therefore, we set out to create one. Formally a NRSfM dataset consists of three correlated types of data:

Observations
: A set of 2D points which is to be given to a NRSfM algorithm as input, denoted as matrix $\mathbf{W}$.

Missing Data
: Indicative data which shows which observation points are visible. Given to a NRSfM algorithm as input along with the observations.

Ground Truth
The 3D deformation corresponding to the 2D observations, denoted as matrix **S**.

As described in paper A, our NRSfM dataset contains several innovations. First we used stop-motion mechatronics to approximate real deformations. This allowed us to record the ground truth with very precise structured light scanning. To be specific, we employed the robot-mounted structured light scanner, shown in Figure 4.2, used in previous work at DTU [Ste+17; Aan+15; Jen+BD; Aan+16]. It also allowed for us to include varied deformations. To be specific, our dataset includes five deformation types; articulated motion (Figure 4.1a), bending (Figure 4.1b), deflation (Figure 4.1c), stretching (Figure 4.1d) and tearing (Figure 4.1e).

Like other data used in evaluation NRSfM (like MOCAP) we create observations by projecting the ground truth using a synthetic camera. However, unlike previous work we create observations according to a factorial design. Specifically, we have six different camera paths and two camera models. An observation matrix is then created for each factorial combination. Our evaluation shows that this factorial design is quite important as the camera has a significant influence on the reconstruction error, regardless of the algorithm used.

Unlike previous work, we create missing data based on self-occlusions. The aforementioned structured light scans not only records the ground truth, but also gives us complete, dense surface reconstructions. As such, creating missing data can be done by raycasting from each point into the camera, while occlusion testing against the recorded surface. As we will later see, this kind of realistic missing data is very different from the randomly removed missing data used in previous work.

The ground truth and observation points were sampled using standard optical flow procedure. Therefore we obtain a more naturally distributed set points compared to marker-based motion capture.

4.1.2 Evaluation Pipeline

Our evaluation is based on a factorial analysis using Analysis of Variance (ANOVA). It is described in paper A. While ANOVA is an well established statistical tool for comparing distributions, it has, to our knowledge, never been applied for NRSfM (or most comparisons in computer vision for that matter). The previous standard has been to simply aggregate some collected error metric for various categories and analyzing the difference without regard for whether the difference is statistical significant. Not only can the ANOVA help to avoid this kind of type I error, it can also separate the influence of various factors on reconstruction quality.

To be precise, our ANOVA model included the influence of five factors on the reconstruction error. These were as follows;

Algorithm a_i
Which algorithm was used.

(a) Articulated (b) Bending (c) Deflation (d) Stretching (e) Tearing

Figure 4.1: Stop-motion mechatronics used for NRSfM dataset creation.

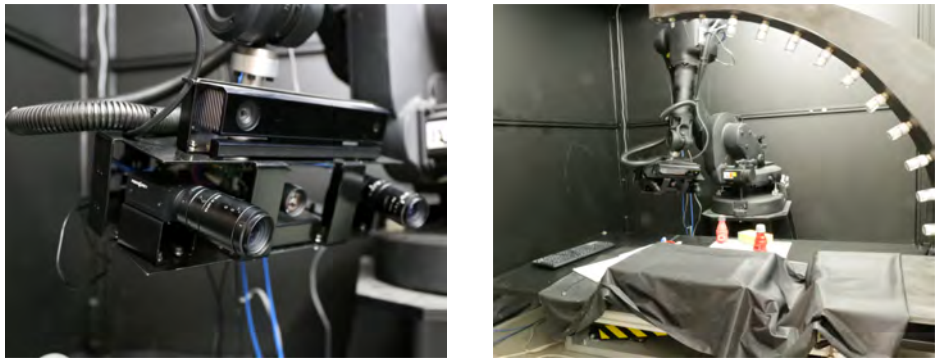
Camera Model m_j
Was the camera model perspective or orthographic.

Animatronics s_k
Which animatronics sequence was reconstructed.

Camera Path p_l
How did the camera move.

Missing Data d_n
Whether occlusion based missing data was used.

Our model allowed us to learn much about the state-of-the-art in NRSfM. Our first analysis without missing data revealed that there is a statistical significant difference between the average reconstruction error for all 16 NRSfM methods included. This is not particularly surprising, so we redid the same factorial analysis with the 5



(a) Scanner (b) Environment

Figure 4.2: The mobile structured light scanner that was used for recording our NRSfM dataset.

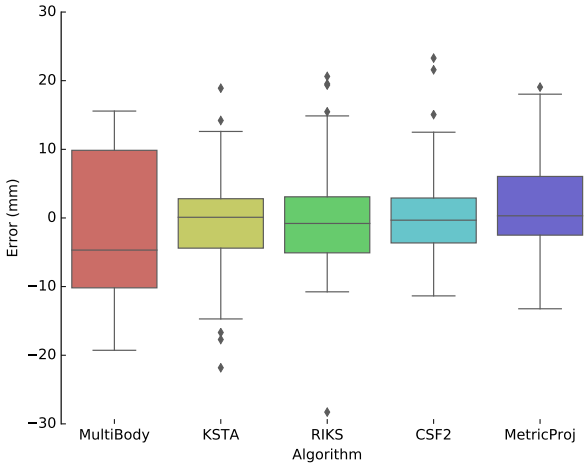


Figure 4.3: Boxplot of the error distribution of the five NRSfM algorithms with the lowest mean error. The distribution has been adjusted for factors like camera motion and deformation type. Notice how closely the means of each distribution resemble each other.

algorithms that had the lowest mean error. These were MultiBody [KDL17a], KSTA [GM11b], RIKS [HGM12], CSF2 [GM11c] and MetricProj [Pal+12]. Interestingly, the analysis revealed no significant difference. Indeed, visualization the error distribution of each after correcting for other factors supports this hypothesis[1]. This is shown in Figure 4.3, as can be seen the error distributions are very similar. Introducing our occlusion-based missing data into the model changes this conclusion however. Almost all methods see a large increase in reconstruction error when subjected to our occlusion-based missing data. The only algorithm that is relatively stable is MetricProj [Pal+12]. Curiously the authors of the method designed their method around the spatio-temporal structure of missing data, whereas other merely focus on the ratio.

The camera model has long been an open question in NRSfM. Specifically, employing a perspective camera model has proven to be challenging. Indeed, we observed this in paper A as only 2 out the 16 included methods uses a perspective camera model. However, our study indicates that the employed camera model is actually not that important. While the camera model does have a statistically significant influence on the reconstruction error, it is small compared to the influence of the deformation type and camera path. Indeed, the few perspective methods we tested did not significantly outperform methods that employ an affine camera on perspectively projected

[1] Using the residuals of an ANOVA model without algorithm terms.

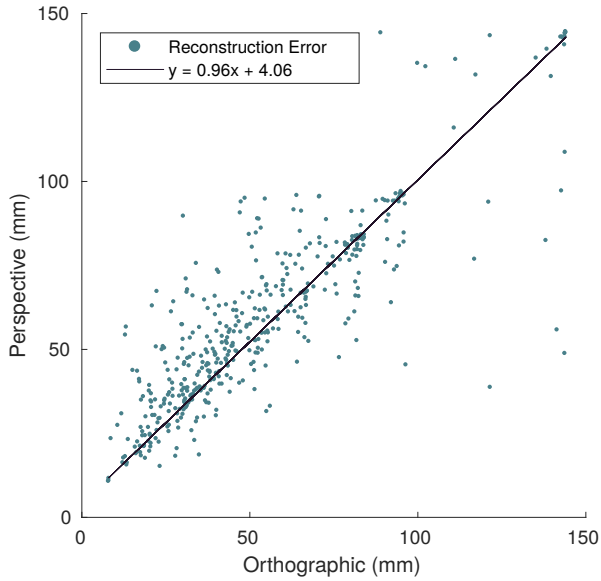


Figure 4.4: Repeated reconstructions with varied camera model. Each point is the reconstruction a factorial combination of algorithm, camera path and deformation under orthographic and perspective camera model. As can be seen there is a rough linear trend. This was intended to be included in paper A, but cut due to page limit.

observations. A look at the data, shown in Figure 4.4, illustrates an interesting trend. The relationship between a reconstruction done under affine and perspective camera can be approximately modeled as line. This line has an approximate unit slope and a positive intercept. Thus, reconstructing perspectively projected observations adds a small constant increase in error, compared to reconstructions using orthogonally projected observations. Though it should be noted that the distance changes in the used camera paths are not particularly extreme, thus keeping perspective artifacts to a minimum (distance changes with a factor 1.6 on average).

We concluded that deformation type has a significant influence on reconstruction error. Particularly articulated motion and stretching (shown in Figure 4.1a and 4.1d) results in a large reconstruction error, no matter which algorithm is used. Indeed, we also showed that the camera path has a significant impact on reconstruction, independent of algorithm used.

4.1.3 Discussion

Our evaluation results has several implications as to the future development of NRSfM. Most of the state-of-the-art methods handle occlusion-based missing data poorly, thus

this area needs attention. It is unclear whether it is the algorithms themselves or their matrix completion that needs improvement. For example, a DCT basis is often used for track completion before doing NRSfM. Perhaps the successful completion method of MetricProj [Pal+12] could be used with other algorithms.

Our studies shows the camera model to be have a small influence on the reconstruction error. Thus, we do not see at being a priority to employ a perspective projection model, especially considering the needed effort.

Articulated motion is an issue which should be dealt with in future work. The non-linear motion of the densely sampled joints poses a challenge regardless of employed prior.

Our analysis also demonstrates the need for controlling the camera. Especially, the camera motion has a significant impact on reconstruction error. Thus employing a taxonomy similar to our work would be benefitial. The influence of the camera path also indicates that future NRSfM research should investigate how to deal with this variance.

4.2 Flexible Robotics for Bin-Picking of Non-Rigid Objects

This section provides an overview our work in solving the bin-picking problem at Danish Crown as described in Section 1.1.2. Details can be found in paper B. Our system for solving this task can be seen in Figure 4.5. It consists of four core components:

Vision
: There are two subparts to this system. First, a structured light scanner which recovers the surface geometry of the box content. Second, a fast segmentation algorithm that separates each meat piece instance. Developed during this thesis.

Robot Arm
: Standard issue 6-Degree of Freedom (DoF) robotic arm.

Suction Cup Gripper
: Gripper intended to provide flexible gripping without damage the meat. Suction cup positions can be adjusted to account for different cutout sizes and shapes. Developed by Jørgensen et al. [Jør+17].

Simulation Framework
: Runs simulation of meat handling and optimizes the robots movement in terms of fast and proper placement. Also yields the optimal suction cup placement for the gripper. Developed by Troels Bo Jørgensen of the Southern University of Denmark.

During online operations, the vision system detects the target piece which is then fed to the path planning algorithm. This algorithm has been parameter tuned using

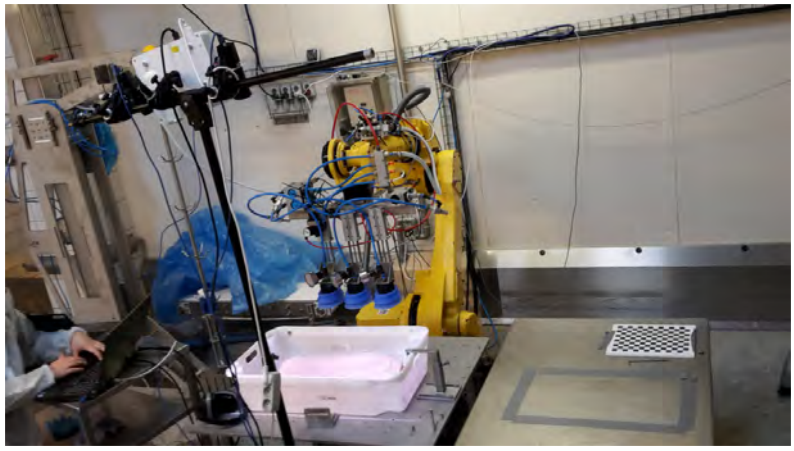


Figure 4.5: The automated solution for bin-picking that has been developed during this thesis. A structured light scanner is used to recover surface geometry, which is then used to guide the robotic arm with suction cup gripper.

the simulation framework, and generates and optimal path for picking and placing. The suction cup gripper is then placed on the meat piece, which is then lifted and placed. This process is then repeated until the box has been emptied. The system was implemented using Robot Operating System (ROS) [Qui+09].

The vision component of this system was developed during this thesis. It uses the structured light method described in Section 2.2.1 to acquire 3D information. This information is then fed to an instance segmentation algorithm that we have developed. It is based on the idea of region growth segmentation which is implemented in Point Cloud Lib (PCL) [RC11; RC17]. Briefly described, it grows a region from a seed of low curvature and terminates at high curvature which is typically at the edge of an object instance. The data from phase shifting structured light arrives in a 2D grid, which we exploit to greatly reduce the region growth runtime. Specifically, the algorithm goes through many neighborhood searching steps, which was originally done in 3D Cartesian coordinate system. We instead search for neighbors in the 2D grid which is why our implementation can segment a 675x540 point cloud in 100-150ms on a laptop [2]. The segmentation algorithm is illustrated in Figure 4.6.

4.2.1 Discussion

We have demonstrated that it is possible to implement a flexible system for bin-picking of non-rigid objects. We successfully tested it on several cutouts of meat. That said our experience also shows that it is quite challenging to control automation with 3D

[2] Specifically on an Intel Core i7-4610M

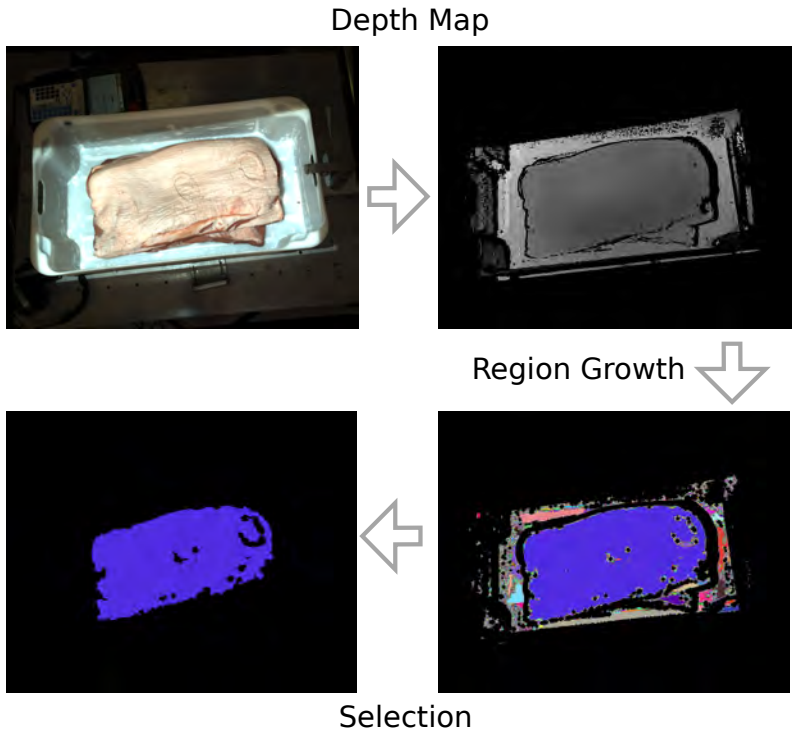


Figure 4.6: Example of the segmentation algorithm in process. Each region/instance is given by a separate color. Last stage is selecting the meat piece to be picked, which is a decision weighted on depth and size.

Figure 4.7: Example of the robot cell doing bin-picking. First frame shows a pattern from the structured light scanner. Second frame shows the robot placing the gripper for lift. Third frame is the lift in action.

vision in a stable manner. One major problem is the aggregation of error from the various components. Individually, the error from simulation, calibration, scanning, segmentation and path planning might seem insignificant, but they can quickly add up to cause major problems. As such one must expect partially erroneous data when designing the systems module.

Another major challenge is the sheer number of unique situations the system can encounter in a real world industrial hall. This is especially true when working with unorganized bins of non-rigid objects. It is very easy to be blinded by good performance on a dataset, and then be unpleasantly surprised when testing in the real environment. It is therefore incredible important to test early and test often in the actual operational environment when developing applied computer vision solutions. A dataset is fine as a performance indicator, but it is important to realize that it only encompasses a subset of the problem domain.

Another good argument for early testing is that some error sources can only be uncovered this way. For example, we found that specular reflections from the ceiling lighting would interfere with our structured light scanner. Varying sunlight due to changing cloud cover would also cause disruptions.

4.3 Error Analysis of Structured Light Scanning of Biological Material

Biological material exhibits heavy subsurface scattering properties, e.g. only 5-7% of the light transmitted by human skin is the results of a direct reflection. To ensure that the structured light method used in the robotic solution produces accurate data, we engaged in a study to ascertain the effects of subsurface scattering on the precision of structured light.

As covered in Section 1.1.3, we wanted to examine how subsurface scattering influence structured light scanning. This was also motivated by our development of the flexible robotics cell at Danish Crown.

Our work is covered in paper C. The approach was simple, scan the surface of some meat. Then coat said meat in a thin layer of diffuse material, which is our case was chalk. Scanning the coated meat then yields the true surface, which we used as a reference. Then we simply applied a standard linear model, based on surface normals and view geometry, to model the error behavior. We discovered that subsurface scattering largely causes a positive bias to structured light scanning, which means the scanned surface seems to be further away than it actually is. This bias is dependent on the specific material and structured light method used and can be up to 1mm. We also found that the error is largely systematic and can be corrected for using the aforementioned linear model. Before correction, methods like micro phase shifting [GN12] and modulated phase shifting [CSL08] has a lower scanning error than standard phase shifting. However after correction with the fitted model, the three methods actually have approximately the same average scanning error.

4.4 Other Contributions

Our work in creating a dataset for evaluation of photorealistic rendering is documented in paper D. We created input data for scenes with glass objects as well as diffuse geometry. The dataset includes fully defined camera pose, BRDFs, scene geometry and environment lightning maps as well as a ground truth image for each camera pose. This was accomplished using the robotics setup shown in Figure. 4.2 as well as CT scanning.

Our efforts in applying computer vision to the geometric metrology problem described in Section 1.1.5 were also successful. The work is documented in poster F.

CHAPTER 5

Conclusion

In this chapter, we will examine the contributions made and see how it aligns with the objectives stated in Section 1.2.

We introduced a new high-quality dataset for Non-Rigid Structure from Motion (NRSfM) that boast of realism, variety and accurate ground truth. In addition, we created missing data via self-occlusion which is much closer to reality than the previously used method of randomly removed missing data. This work is described in detail in paper A.

We used this dataset to perform a factorial analysis of the performance of 16 state-of-the-art methods. Not only did this provide us with valueable insights, such as articulated motion remains difficult to reconstruct, but it also lays the ground work for future evaluations of the field. We also show that employing a perspective camera model in future work should not be a priority. The complete evaluation can be found in paper A.

Our efforts in creating a flexible robotics cell for bin-picking of non-rigid objects resulted in an integration of structured light 3D and a 6-DoF robot arm with a suction cup gripper. The scanner provides accurate 3D data which is used to guide the arm. This prototype demonstrates that the problem is indeed solvable with a single flexible automation solution. We learned that robust error handling is vital for stable operations. Additionally, early field testing should be a priority. This work is fully described in paper B.

We studied the effects of subsurface scattering on structured light scanning by studying it's accuracy in scanning biological material. The scanning error could be described using a simple linear model based on view geometry. It demonstrates that the error manifests largely as a positive offset and that much of it can be corrected with the aforementioned model. This work is described in paper C.

A dataset for evaluation of photorealistic rendering was created with complete input data and reference images. We created it using a variety of vision techniques, such as structured light scanning as well as CT. The depicted scenes contains challenging optical objects, such as glass, and was effectively used for evaluation. Furthermore,

we also demonstrate it's applicability in analysis by synthesis. The dataset is fully described in paper D.

We have implemented an automatic solution for measuring the area of contact surfaces. With this we demonstrated the importance of traceability for 3D vision as it effectively determines systematic errors and biases. This work is presented in poster F.

All in all we must conclude that the objectives of thesis has been met.

APPENDIX A

A Benchmark and Evaluation of Non-Rigid Structure from Motion

A Benchmark and Evaluation of Non-Rigid Structure from Motion

Sebastian Hoppe Nesgaard Jensen, Alessio Del Bue, Mads Emil Brix Doest, Henrik Aanæs,

Abstract—Non-Rigid structure from motion (NRS*f*M), is a long standing and central problem in computer vision, allowing us to obtain 3D information from multiple images when the scene is dynamic. A main issue in the further development of this important computer vision topic, is the lack of high quality data sets. We here address this issue by presenting of data set compiled for this purpose, which is made publicly available, and considerably large than previous state of the art. To validate the applicability of this data set, and provide and investigation into the state of the art of NRS*f*M , including potential directions forward, we here present a benchmark and a scrupulous evaluation for this data set. This benchmark evaluates 16 different methods with available code, which we argue reasonably spans the state of the art in NRS*f*M . We also hope, that the presented and public data set and evaluation, will provide benchmark tools for further development in this field.

Index Terms—Non-Rigid Shape Recovery, Non-Rigid Structure from Motion, Deformation Modelling.

THE estimation of structure and motion from an image sequence, i.e. the structure from motion (SfM) or monocular simultaneous localization and mapping (SLAM) problem, is one of the central problems within computer vision. This problem has received a lot of attention, and truly impressive advances has been made over the last ten to twenty years. It plays a central role in robot navigation, self-driving cars, and 3D reconstruction of the environment, to mention a few. A central part of maturing regular SfM is the availability of sizeable data sets with rigorous evaluations, e.g. [1][2].

The regular SfM problem, however, primarily deals with rigid objects, which is somewhat at odds with the world we see around us. That is, trees sway, faces express themselves in various expressions, and most non-static organic objects are generally non-rigid. The issue of making this obvious and necessary extension of the SfM problem, is referred to as the non-rigid structure from motion problem (NRS*f*M). A problem that also has a central place in computer vision. The solution to this problem is, however, not as mature as the regular SfM problem. A reason for this is the scarcity of high quality data sets and accompanying evaluations. Such data and evaluations allow us to better understand the problem domain and better determine what works best and why.

- *Sebastian Hoppe Nesgaard Jensen is with DTU Compute, Richard Petersens Plads, Building 324, Kongens Lyngby 2800, Denmark. Email: snje@dtu.dk*
- *Alessio Del Bue is with the Visual Geometry and Modelling (VGM) Lab, Istituto Italiano di Tecnologia (IIT), Genova, 08028, Italy. Email: alessio.delbue@iit.it*
- *Mads Emil Brix Doest is with DTU Compute, Richard Petersens Plads, Building 324, Kongens Lyngby 2800, Denmark. Email: mebd@dtu.dk*
- *Henrik Aanæs is with DTU Compute, Richard Petersens Plads, Building 324, Kongens Lyngby 2800, Denmark. Email: aanes@dtu.dk*

To address this issue we here introduce a high quality data set, with accompanying ground truth (or reference data to be more precise) aimed at evaluating non-rigid structure from motion. To the best of our knowledge, this data set is significantly larger and more diverse than what has previously been available – c.f. Section 3 for a comparison to previous evaluations of NRS*f*M . The presented data set better captures the variability of the problem, and gives higher statistical strength of the conclusions reached via it. Accompanying this data set, we have conducted an evaluation of 16 state of the art methods, hereby validating the suitability of our data set, and providing insight into the state of the art within NRS*f*M . This evaluation was part of the competition we held at a CVPR 2017 workshop, aimed at NRS*f*M . It is our hope and belief that this data set and evaluation will help in furthering the state of the art in NRS*f*M research, by providing insight and a benchmark. The data set is publicly available at
http://nrsfm2017.compute.dtu.dk/dataset.

This paper is structured by first giving an overview of the NRS*f*M problem, followed by a overview of related work, wrt. other data sets. This is then followed by a presentation of our data set, including an overview of the design considerations, c.f. Section 3, which is followed by a presentation of our proposed protocol for evaluation, c.f. Section 4. This leads to the result of our benchmark evaluation in Sections 5. The paper is rounded of by a discussion and conclusions in Section 6.

2 THE NRS*f*M PROBLEM

In this section, we will provide a brief introduction of the NRS*f*M problem, followed by a more detailed overview of ways this problem has been addressed. The intention is to establish a taxonomy to base our experimental design

and evaluation upon. For a more in-depth review of NRS*f*M , we recommend the survey of Salzmann et al. [3].

The standard/rigid SfM problem, c.f. e.g. [4], is an inverse problem aimed at finding the camera positions (and possibly internal parameters) as well as 3D structure – typically represented as a static 3D point set, Q – that best describe a sequence of 2D images of a rigid body. Where the 2D images are typically reduced to a sparse set of tracked 2D point features, corresponding to the 3D point set, Q. The most often employed observation model linking 2D image points to 3D points and camera models, is either the *perspective camera model*, or the *weak perspective* approximation here of. The weak perspective camera model is derived from the full perspective model, by simplifying the projective effect of 3D point depth, i.e. the distance between camera and 3D point.

The extension from rigid structure from motion to the non-rigid case is by allowing the 3D structure, here points Q_f, to vary from frame to frame, i.e.

$$\mathbf{Q}_f = \begin{bmatrix} \mathbf{Q}_{f,1} & \mathbf{Q}_{f,2} & \cdots & \mathbf{Q}_{f,P} \end{bmatrix} , \tag{1}$$

Where $\mathbf{Q}_{f,p}$ is the 3D position of point p at frame f. To make this NRS*f*M problem well-defined, a prior or regularization is often employed. Here most of the cases target the spatial and temporal variations of $\mathbf{Q}_f$. The fitness of the prior to deformation in question is a crucial element in successfully solving the NRS*f*M problem, and a main difference among NRS*f*M methods is this prior.

In this study, we denote NRS*f*M methods according to a three category taxonomy, i.e. the **deformable model** used (statistical or physical), the **camera model** (affine, weak or full perspective) and the ability to deal with **missing data**. In the remainder of this section, this taxonomy will be elaborated on and related to the litterature, leading up to a discussion of how the NRS*f*M methods we evaluate, c.f. TABLE 1, span the state of the art.

2.1 Deformable Models

The description of our taxonomy will start with the underlying structure deformation model category, divided into statistical and physical based models.

2.1.1 Statistical

This set of algorithms apply a statistical deformation model with no direct connection with the physical process of structure deformations. They are in general heuristically defined a priori to enforce constraints that can reduce the ill-posedness of the NRS*f*M problem. The most used low-rank model in the NRS*f*M literature falls into this category, utilizing the assumption that 3D deformations are well described by linear subspaces (also called basis shapes). This property was first used in 2000 by Bregler, Hertzmann and Biermann [5] to first instantiate the solution of NRS*f*M by solving a factorization problem, as analogously made by Tomasi and Kanade for the rigid case [6]. However, strongly nonlinear deformations, such as the one appearing in articulated shapes, may drastically reduce the effectiveness of such models. Moreover, the low-rank model acts mainly as a constraint over the spatial distribution of the deforming point cloud and it does not restrict the temporal variations of the deforming object.

Given this observation, Akhter et al. [7] was the first to propose constraining the temporal deformations of the object, using a set of DCT bases, thus, assuming that deformations act with low-frequency components. This principle was supported by a study indicating a correlation between 3D bases extracted by PCA on MoCap sequences of human motion: the distribution of the linear weights closely resemble the DCT ones [8]. Even at the expenses of introducing a new parameter, this principle of smoothing deformations in the temporal domain was able to achieve reasonable results with human motion modelling, even applied to synthetically generated sequences with a large camera motion [7].

Differently, Gotardo et al. [9] had the intuition to use the very same DCT bases to model camera and deformation motion instead, assuming those factors are smooth in a video sequence. This approach was later expanded on to explicitly modeling a set of complementary rank-3 spaces, and to constrain the magnitude of deformations in the basis shapes [10]. An extension of this framework, increased the generalization of the model to non-linear deformations, with a kernel transformation on the 3D shape space using radial basis functions [11]. This switch of perspective, addressed the main issue of increasing the number of available DCT bases, allowing more diverse motions, while not restricting the complexity of deformations. Later, further extension and optimization have been made to low-rank and DCT bases approaches. Valmadre and Lucey [12] noticed that the trajectory should be a low-frequency signal, thus laying the ground for an automatic selection of DCT basis rank via penalizing the trajectory's response to one or more high-pass filters. Moreover, spatio-temporal constraints have been imposed both for temporal and spatial deformations [13]. Recently a new prior model, related to the Kronecker-Markov structure of the covariance of time-varying 3D point, very well generalizes several priors introduced previously [14]. Another recent improvement is given by Ansari et al.'s usage of DCT basis in conjunction singular value thresholding for camera pose estimation [15].

Similar spatial and temporal priors have been introduced as regularization terms while optimizing a cost function solving for the NRS*f*M problem, mainly using a low-rank model only. Torresani et al. [16] proposed a probabilistic PCA model for modelling deformations by marginalizing some of the variables, assuming Gaussian distributions for both noise and deformations. Moreover, in the same framework, a linear dynamical model was used to represent the deformation at the current frame as a linear function of the previous. Brand [17] penal-

izes deformations over the mean shape of the object by introducing a sensible parameters over the degree of flexibility of the shape. Del Bue et al. [18] instead compute a more robust non-rigid factorization, using a 3D mean shape as a prior for NRS*f*M [19]. In a non-linear optimization framework, Olsen et al. [20] include l_2 penalties both on the frame-by-frame deformations and on the closeness of the reconstructed points in 3D given their 2D projections. Of course, penalty costs introduce a new set of hyper-parameters that weights the terms, implying the need for a further tuning, that can be impracticable when cross-validation is not an option. Regularization has also been introduced in formulations of Bundle Adjustment for NRS*f*M [21] by including smoothness deformations by using l_2 penalties mainly [22] or constraints over the rigidity of pre-segmented points in the measurement [23].

Another important statistical principal is enforcing that low-rank bases are independent. In the coarse to fine approach of Bartoli et al. [24], bases shapes are computed sequentially by adding a basis, which explain most of the variance in respect to the previous ones. They also impose a stopping criteria, thus, achieving the automatic computation of the overall number of bases. The concept of basis independence clearly calls for a statistical model close to Independent Component Analysis (ICA). To this end, Brandt et al. [25] proposed a prior term to minimize the mutual information of each basis in the NRS*f*M model. Low-rank models are indeed compact but limited in the expressiveness of complex deformations, for this reason, an over complete representation can still be used by imposing sparsity over the selected bases [26]. In this way, 3D shapes in time can have a compact representation, and they can be theoretically characterized as a block sparse dictionary learning problem. In a similar spirit, Hamsici et al. propose to use the input data for learning spatially smooth shape weights using rotation invariant kernels [27].

All these approaches for addressing NRS*f*M with a low-rank model have provided several non-linear optimization procedures, mainly using Alternating Least Squares (ALS), Lagrange Multipliers and alternating direction method of multipliers (ADMM). Torresani et al. first proposed to alternate between the solution of camera matrices, deformation parameters and basis shapes. This first initial solution was then extended by Wang et al. [28] by constraining the camera matrices to be orthonormal at each iteration while Paladini et al. [29] strictly enforced the matrix manifold of the camera matrices to increase the chances to converge to the global optimum of the cost function. All these method were not been designed to be strictly convergent, for this reason a Bilinear Augmented Multiplier Method (BALM) [30] was introduced to be convergent while implying all the problems constraints being satisfied. Furthermore, robustness in terms of outlying data was then included to improve results in a proximal method with theoretical guarantees of convergence to a stationary point [31].

Despite the non-linearity of the problem, it is possible to relax the rank constraint with the trace norm and to solve the problem with convex programming. Following this strategy Dai et al. provided one of the first effective closed form solutions to the low-rank problem [32]. Although their convex solution, resulting from relaxation, did not provide the best performance, a following iterative optimization scheme gave improved results. In this respect, Dai et al. proposed a further improvement on their previous approach where deformations are represented as a spatio-temporal union of subspaces rather than a single subspace [33]. Thus complex deformation can be represented as the union of several simple ones.

More recently, the Procrustean Normal Distribution (PND) model was proposed as an effective way to implicitly separate rigid and non-rigid deformations [34]. This separation provides a relevant regularization, since rigid motion can be used to obtain a more robust camera estimation, while deformations are still sampled as a normal distribution as similarly done previously [16]. Such separation is obtained by enforcing an alignment between the reconstructed 3D shapes at every frame that in practice should factor out the rigid transformations from the statistical distribution of deformations. The PND model has been then extended to deal with more complex deformations and longer sequences [35].

2.1.2 Physical

Physical models represents a less studied class, but in practice the one being able to achieve most accuracy in spite of the higher number of parameters required to tune. They are characterized by the use one of several properties of deforming materials, and we will start from the most general ones towards the most specialized.

The first class of physical model assume that the non-rigid object is piecewise, i.e. a collection of pre-defined or estimated patches that are mostly rigid or slightly deformable. One of the first approaches to use this strategy is Varol et al. [36]. By preselecting a set of overlapping patches from the 2D image points, and assuming each patch being rigid, homography constraints can be imposed at each patch, followed by global 3D consistency being enforced using the overlapping points. However, the rigidity of a patch, even if small, is a very hard constraint to impose and it does not generalise well for every non-rigid shape. Moreover, dense point-matches over the image sequence are required to ensure a set of overlapping points among all the patches. Relaxation to the piece-wise rigid constraint was given by Fayad et al. [37], assuming each patch deforming with a quadratic physical model, thus, accounting for linear and bending deformations. These methods all require an initial patch segmentation and the number of overlapping points, to this end, Russel et al. [38] optimize number of patches and overlap by defining a cost function with energy terms. The method of Lee et al. [39] instead use 3D reconstructions of multiple combination of patches and define a 3D consensus between a set of pacthes. This

approach provides a fast way to bypass the segmentation problem and robust mechanism to prune out wrong local 3D reconstructions.

Differently from these approaches, Taylor et al. [40] constructs a triangular mesh, connecting all the points, and considering each triangle as being locally rigid. Global consistency is here imposed to ensure that the vertexes of each triangle are coinciding in 3D. Again, this approach is to a certain extent similarly to [36], which requires a dense set of points in order to comply with the local rigidity constraint.

A strong prior, which helps dramatically to mitigate the ill-posedness of the problem, is obtained by considering the deformation isometric, i.e. the metric length of curves does not change when the shape is subject to deformations (e.g. paper, metallic materials to some extent). Using assumption that a surface can be approximated as infinitesimally planar, Chhatkuli et al. [41] proposed a local method that frame NRS*f*M as the solution of Partial Differential Equations (PDE) being able to deal with missing data as well. A further update [42] formalizes the framework in the context of Riemannian geometry, that led to a practical method for solving the problem in linear time and scaling for a relevant number of views and points. Furthermore, a convex formulation for NRS*f*M with inextensible deformation constraints was implemented using Second-Order Cone Programming (SOCP) leading to a closed form solution to the problem [43]. Vincente and Agapito implemented soft inextensibility constraints [44] in an energy minimization framework, e.g. using recently introduced techniques for discrete optimization.

Another set of approaches try to directly estimate the deformation function using high order models. Del Bue and Bartoli [45] extended and applied 3D warps such as the thin plate spline, to the NRS*f*M problem. Starting from an approximate mean 3D reconstruction, the warping function can be constructed and the deformation at each frame can be solved by iterating between camera and 3D warp field estimation. Finally, Agudo et al. introduced the use of Finite Elements Models (FEM) in NRS*f*M [46], [47]. As these models are highly parametrized, requiring the knowledge of the material properties of the object (e.g. the Young modulus), FEM needs to be approximated in order to be efficiently estimated, however, in ideal conditions it might achieve remarkable results, since FEM is a consolidated technique for modelling structural deformations.

2.2 Missing Data

The initial methods for NRS*f*M assumed complete 2D point matches among views, when observing a deformable object. However, given self and standard occlusions, this is rarely the case. Most approaches for dealing with such missing data in NRS*f*M were framed as a matrix completion problem, i.e. estimate the missing entries of the matrix $\mathtt{W}$ given known constraints (mainly matrix low-rank). Torresani et al. [48] first proposed removing rows and lines of the matrix corresponding to missing entries in order to solve the NRS*f*M problem. However, this strategy suffers greatly from even small percentages of missing data, since the subset of know completely entries can be very small. Dai et al. [32] complete the missing entries via convex optimisation by relaxing the rank constraint using a matrix trace norm. Indeed, this method can be robust to more missing entries even do being computationally viable only for smaller scale problems. Most of the iterative approaches indeed include an update step of the missing entries [29], [30] where the missing entries become an explicit unknown to estimate. Gotardo et al. [9] instead strongly reduce the number of parameters by estimating only the camera matrix explicitly under severe missing data. This variable reduction, also know as VARPRO in the optimization literature. It has been recently revisited in relation to several structure from motion problems [49].

2.3 Camera Model

Most NRS*f*M method research focus on modelling and optimization aspects, and most assume a weak perspective camera model. However, in cases where the object is close to the camera and undergoing strong changes in depth, time-varying perspective distortions can affect the measured 2D trajectories.

As low-rank NRS*f*M is treated as a factorization problem, a straightforward extension was to follow best practices from rigid SfM for perspective camera. Xiao and Kanade [50] have e.g. developed a two step factorization algorithm for reconstruction of 3D deformable shapes under the full perspective camera model. This is done using the assumption that a set of basis shapes are known to be independent. Vidal and Abretske [51] have also proposed an algebraic solution to the non-rigid factorization problem. Their approach is, however, limited to the case of an object being modelled with two independent basis shapes and viewed in five different images. Wang et al. [52] proposed a method able to deal with the perspective camera model, but under the assumption that its internal calibration is already known. They update the solutions from a weak perspective to a full perspective projection by refining the projective depths recursively, and then refine all the parameters in a final optimization stage. Finally, Hartley and Vidal [53] have proposed a new closed form linear solution for the perspective camera case. This algorithm requires the initial estimation of a multifocal tensor, which the authors report is very sensitive to noise. Llado et al. [54], [55] proposed a non-linear optimization procedure. It is based on the fact that it is possible to detect nearly rigid points in the deforming shape, which can provide the basis for a robust camera calibration.

2.4 Evaluated Methods

We have chosen a representative subset of the aforementioned methods, which are summarized according

to our taxonomy in TABLE 1. This gives us a good representation of recent work, distributed according to our taxonomy with a decent span of deformation models (statistical/physical) and camera models (orthographic, weak perpsective or perspective). This also takes into account in-group variations such as DCT basis for statistical deformation and isometry for physical deformation. Even lesser used priors, such as compressibility, are represented. While this is not a full factorial study, we think this reasonably spans the recent state of the art of NRS*f*M . Our choice has, of course, also been influence by method availability, as we want to test the author's original implementation, to avoid our own implementation bias/errors. All in all, we have included 16 methods in our evaluation. A further omission is Taylor et al.'s work [40] since the approach has the option to remove points associated to triangle patches which are likely to provide a wrong 3D estimate. This method was removing a considerable number of points in the tested dataset so it was not included in the evaluation since every NRS*f*M approach is able to reconstruct a complete 3D reconstruction[1].

3 Dataset

As stated, in order to compare state of the art methods for NRS*f*M , we have compiled a larger data set for this purpose. Even though there is a lack of empirical evidence w.r.t. NRS*f*M , it does not imply, that no data sets for NRS*f*M exist.

As an example in [39], [9], [10], [11], [33], [8], [27] and [32], a combination of two data sets are used. Namely seven sequences of a human body from the CMU motion capture database [56], two MoCap sequences of a deforming face [57], [58], a computer animated shark [57] and a challenging flag sequence [37]. To the best of our knowledge, this list represents the most used evaluation data sets for NRS*f*M with available ground truth.

The CMU data set [56] captures motion of humans. Since the other frequently used data sets are also related to animated faces [57], [58], this implies that there is a high over representation of humans in this state of the art, and that a higher variability in the deformed scenes viewed is deemed beneficial. In addition, the shark sequence [57] is not based on real images and objects, but on computer graphics and pure simulation. As such there is a need for new data sets, with reliable ground truth or reference data[2], and a higher variability in the objects and deformations used.

As such, we here present a data set consisting of five widely different objects/scenes and motions. Based on mechanically - and therefor repeatable - object motions, we have defined six different camera motions employing two different camera models. This setup, all in all, gives 60 different sequences organized in a factorial experimental design, thus, enabling a more stringent statistical analysis. In addition to this, since we have tight 3D surface models of our objects or scenes, we are able to determine occlusions of all 2D feature points. This in turn gives a realistic handling off missing data, which is often due to object self occlusion.

Fig. 1. Mobile structured light scanner used to acquire 3D data for the data set.

As indicated, these data sets are achieved by stop-motion using mechanical animatronics. These are recorded in our robotic setup previously used for generating high quality data sets c.f. e.g. [59]. We will here present details of our data capture pipeline, followed by a brief outline and discussion of design considerations.

The goal of the data capturing is to produce 3 types of correlated data:

Ground Truth: A series of 3D points that change over time.
Input Tracks: 2D tracks used for input for NRS*f*M .
Missing Data: Binary data representing which tracks are occluded at what frame.

We record the step-wise deformation of our animatronics from K static views, obtaining both image data and dense 3D surface geometry. We obtain 2D point features by applying standard optical flow tracking to the image sequence obtained from each of the K views, which is then reprojected onto the recorded surface geometry. The ground truth is then the union of these 3D tracks. By using optical flow for tracking instead of MoCap markers, we obtain a more realistic set of ground truth points. We create input 2D points by projecting the recorded ground truth using a virtual camera in a fully factorial design of camera paths and camera models.

In the following we will detail some of the central parts of the above procedure.

3.1 Animatronics & Recording Setup

Our stop-motion animatronics are five mechatronic devices capable of computer controlled gradual deformation. They are shown in Fig. 2, and cover five types of deformations: articulated motion, bending, deflation, stretching and tearing. We belive this covers a good range of interesting and archetypal deformations. It is noted, that NRS*f*M has previously been tested on bending and tearing [40], [44], [43], [39], but without ground

1. this issue has been noticed in other experimental data for NRS*f*M , rarely this approach has been included for evaluation in other papers.
2. With real measurements like ours the 'ground truth' data also include noise, why 'reference data' is a more correct term.

TABLE 1
Methods included in our NRS*f*M evaluation with annotations of how they fit into our taxonomy.

Method	Citation	Deformable Model	Camera Model	Missing Data
BALM	[30]	Statistical	Orthographic	Yes
Bundle	[22]	Statistical	Weak Perspective	Yes
Compressible	[26]	Statistical	Weak Perspective	-
Consensus	[39]	Statistical	Orthographic	-
CSF	[9]	Statistical	Weak Perspective	Yes
CSF2	[10]	Statistical	Orthographic	Yes
EM PPCA	[16]	Statistical	Weak Perspective	Yes
KSTA	[11]	Statistical	Orthographic	Yes
MDH	[43]	Physical	Perspective	Yes
MetricProj	[29]	Statistical	Orthographic	Yes
MultiBody	[33]	Statistical	Orthographic	-
PTA	[8]	Statistical	Orthographic	-
RIKS	[27]	Statistical	Orthographic	-
ScalableSurface	[15]	Statistical	Orthographic	Yes
SoftInext	[44]	Physical	Perspective	Yes
SPFM	[32]	Statistical	Orthographic	-

truth for quantitative comparison. Additionally, elastic deformations, like deflation and stretching, are quite commonplace, but hasn't appeared in any previous data sets, to the best of our knowledge.

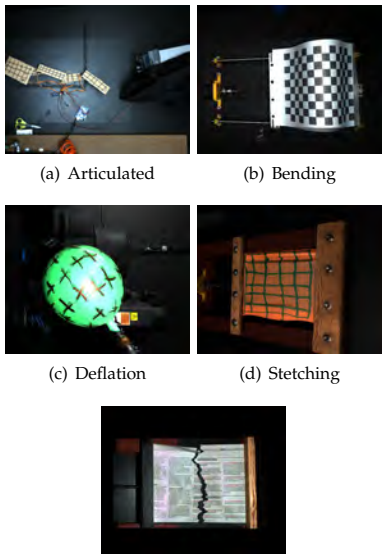
The animatronics can hold a given deformation or pose for a large extent of time, thus, allowing us to record accurately the object's geometry. We, therefore, do not need a real-time 3D scanner or elaborate multi-scanner setup. Instead our recording setup consists of an in-house built structured light scanner mounted on an industrial robot. Tested according to standard VDI 2634-2 [60] the scanner has a form error of [0.01mm, 0.32mm], a sphere distance error of [-0.33mm 0.50mm] and a flatness error of [0.29mm, 0.56mm]. This setup is shown in Fig. 1. This does not only provide us with accurate 3D scan data, but the robot's mobility also enables a full scan of the object at each deformation step.

3.2 Recording Procedure

The recording procedure acquires for each shape a series of image sequences and surface geometries of its deformation over F frames. We record each frame from K static views with our aforementioned structured light scanner. As such we obtain K image sequences with F images in each. We also obtain F dense surface reconstructions, one for each frame in the deformation. The procedure is summarized in pseudo code in Algorithm 1. Fig. 3 illustrates a sample images of three views obtained using the above process.

3.3 3D Ground Truth Data

The next step is to take acquired images $I_{f,k}$ and surfaces S_f, and extract the ground truth points. We do this by applying optical flow tracking [61] to obtain 2D tracks, which are then reprojected onto S_f. The union of of these reprojected tracks gives us the ground truth, Q. This process is summarized in pseudo code in Algorithm 2.



(a) Articulated (b) Bending

(c) Deflation (d) Stetching

(e) Tearing

Fig. 2. Animatronic systems used for generating specific types of non-rigid motion.

3.4 Projection using Virtual Camera

To produce the desired input, we project the ground truth $\mathbf{Q}$ using a virtual camera, similar to what has been done in [39], [9], [32], [58]. This step has two factors related to the camera that we wish to control for: path and camera model. To keep our design factorial, we define six different camera paths which will all be used to create the 2D input. They are illustrated in Fig. 4. We believe these are a good representation of possible camera motion with both linear motion and panoramic panning. The camera model can be either orthographic or perspective. The factorial combination of these el-

Algorithm 1: Process for recording image data for tracking and dense surface geometry for an animatronic.

```
Let F be the number of frames
Let k be the number of static scan views K
for f ∈ F do
    Deform animatronic to pose f
    for k ∈ K do
        Move scanner to view k
        Acquire image I_{f,k}
        Acquire structured light scan S_{f,k}
    end
    Combine scans S_{f,k} for full, dense surface S_f
end
```

Algorithm 2: Process for extracting the ground truth Q from recorded images and surface scans.

```
Let F be the number of frames
Let k be the number of static scan views K
Let S_f be the surface at frame f
Let I_{f,k} be the image from view k, frame f
S = {S_1 ... S_F}
for k ∈ K do
    I_k = {I_{1,k} ... I_{F,k}}
    Apply optical flow [61] to I_k to get 2D tracks T_k
    Reproject T_k onto S to get 3D tracks Q_k
end
Q = {Q_1 ... Q_K}
```

ements yields to 12 input sequences for each ground truth. Additionally, as we have previously recorded the dense surface for each frame (see Sec. 3.2), we estimate missing data via self-occlusion. Specifically, we create a triangular mesh for each S_f and estimate occlusion via raycasting into the camera along the projection lines. This process is summarized in pseudo code in Algorithm 3.

3.5 Discussion

While stop-motion does allow for diverse data creation, it is not without drawbacks. Natural acceleration is easily lost when objects deform in step-wise manner and recordings are unnaturally free of noise like motion blur. However, without this technique, it would have been prohibitive to create data with the desired diversity and accurate 3D ground truth.

The same criticism could be levied against the use of a virtual camera, it lacks the shakiness and acceleration of a real world camera. On the other hand, it allows for us to precisely vary both the camera path and camera model. This enables us to perform a factorial analysis, in which we can study the effects of both on NRS*f*M . As we show in Sec. 5 some interesting conclusions are drawn from this analysis. Most NRS*f*M methods are designed with an orthographic camera in mind. As such investigating the difference between data under orthographic and perspective projection is of interest. Such investigation is only possible using a virtual camera.

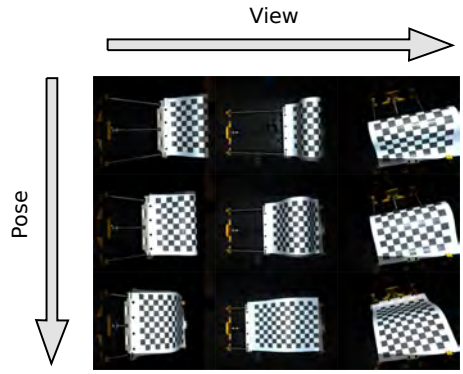


Fig. 3. Illustrative sample of our multi-view, stop-motion recording procedure. Animatronic pose evolves vertically and scanner view change horizontally.

4 Evaluation Metric

In order to compare the methods of TABLE 1 w.r.t. our data set, a metric is needed. The purpose is to project the high dimensional 3D reconstruction error into (ideally) a one dimension measure. Several different metrics have been proposed for NRS*f*M evaluation in the past literature, e.g. the Frobenius norm [62], mean [27], variance normalized mean [10] and RMSE [40].

All of the above mentioned evaluation metrics are based on the $L2$-norm in one form or another. A drawback of this is, that the $L2$-norm is very sensitive to large errors, often letting a few outliers dominate an evaluation. To address this, we incorporate robustness into our metric, by introducing truncation of the individual

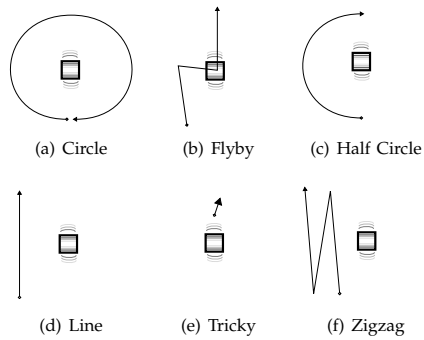


Fig. 4. Camera path taxonomy. The box represents the deforming scene and the wiggles illustrates the main direction of deformation, e.g. the direction of stretching.

Algorithm 3: Creation of input tracks $W_{c,p}$ and missing data $D_{c,p}$ from ground truth Q for each combination of camera path p and model c.

```
Let F be the number of frames
Let P be the set of camera paths shown in Fig. 4
Let C be the either perspective or orthographic
Let Q_f be the ground truth at frame f
Let S_f be the surface at frame f
for S_f ∈ {S_1 ... S_F} do
    Estimate mesh M_f from S_f
end
for c ∈ C do
    for p ∈ P do
        for f ∈ F do
            Set camera pose to p_f
            Project Q_f using model c to get points w_f
            Do occlusion test q_f against M_f to get
             missing data d_f
        end
        W_{c,p} = {w_1 ... w_F}
        D_{c,p} = {d_1 ... d_F}
    end
end
```

3D point reconstruction errors. In particular, our metric is based on a RMSE measure similar used in Taylor et al. [40].

The robust truncation is achieved in a manner similar to the widely used box plot's outlier detection strategy [63]. Consider E being the set of point-wise errors ($||\mathbf{X}_{f,p} - \mathbf{Q}_{f,p}||$) and E_1, E_3 as being the first and third quartile of that set. Now define the whisker as $w = \frac{3}{2}(E_3 - E_1)$, then any point that is more than a whisker outside of the interquantile range (E_3 - E_1) is considered as an outlier. This strategy works well for approximately normally distributed data [64]. With this in mind, our truncation function is defined as follows,

$$t\left(\mathbf{x}, \mathbf{q}\right) = \begin{cases} ||\mathbf{x} - \mathbf{q}||, & ||\mathbf{x} - \mathbf{q}|| < E_3 + w \\ E_3 + w, & \text{otherwise} \end{cases} \tag{2}$$

Thus the robust RMSE is defined as,

$$m\left(\mathbf{Q}, \mathbf{X}\right) = \sqrt{\frac{1}{FP} \sum_{f,p}^{F,P} t\left(\mathbf{X}_{f,p}, \mathbf{Q}_{f,p}\right)}. \tag{3}$$

A NRS*f*M reconstruction is given in an arbitrary coordinate system, thus we must align the reference and reconstruction before computing the error metric. This is typically done via Procrustes Analysis [65], but as it minimizes the distance between two shapes in a $L2$-norm sense it is also sensitive to outliers. Therefore we formulate our alignment process as an optimization problem based on the robust metric of Eq. 3. Thus the combined metric and alignment is given by,

$$m(\mathbf{X}, \mathbf{Q}) = \min_{s,\mathbf{R},\mathbf{t}} \sqrt{\frac{1}{FP} \sum_{f,p} t\left(s\left[\mathbf{R}\mathbf{X}_{fp} + \mathbf{t}\right], \mathbf{Q}_{fp}\right)}, \tag{4}$$

where s = scale,
$\mathbf{R}$ = rotation and reflection,
$\mathbf{t}$ = translation.

An implication of using a robust, as opposed to a $L2$-norm, is that the minimization problem of (4) cannot be achieved by a standard Procrustes alignment as done in [40]. As such we optimize (4) using the Levenberg-Marquardt method, where s, $\mathbf{R}$ and $\mathbf{t}$ have been initialized via Procrustes alignment [66].

In summary, (4) defines the alignment and metric that has been used for the evaluation presented in Sec. 5.

Since the choice of metric, always has a streak of subjectivity to it, we wanted to investigate the sensitivity of our choice. We did this by repeating our evaluation with another robust metric, where minimum track-wise distance between the ground truth and reconstruction was used. The major findings and conclusions, as presented in Sec. 5, were the same. As such we conclude that our conclusions are not overly sensitive to the choice of metric. Note, that due to space limitations and clarity of presentation this sensitivity study is not treated further in this text.

5 Evaluation

With our data set and robust error metric, we have performed a thorough evaluation and analysis of the state-of-the-art in NRS*f*M , which is presented in the following. This is done in part as an explorative analysis and in part to answer some of what we see as most pressing, open questions in NRS*f*M . Specifically:

- Which algorithms performs the best?
- Which deformable models have best performance/generalization?
- How well can the state-of-the-art handle data from a perspective camera?
- How well can the state-of-the-art handle occlusion-based missing data?

To answer these questions, we perform our analysis in a factorial manner, alligned with the factorial design of our data set. To do this, we view a NRS*f*M reconstruction as a function of the following factors:

Algorithm a_i**:**	Which algorithm was used.
Camera Model m_j**:**	Which camera model was used (perspective or orthographic).
Animatronics s_k**:**	Which animatronics sequence was reconstructed.
Camera Path p_l**:**	How the camera moved.
Missing Data d_n**:**	Whether occlusion based missing data was used.

We design our evaluation to be almost fully crossed, meaning we obtain a reconstruction for every combination of the above factors. The only missing part

is that the authors of MultiBody [33] only submitted reconstructions for orthographic camera model.

Our factorial experimental design allows us to employ a classic statistical method known as ANalysis Of VAriance (ANOVA) [67]. The ANOVA not only allows us to deduce the precise influence of each factor on the reconstruction, but also allows for testing their significance. To be specific, we model the reconstruction error in terms of the following bilinear model,

$$\begin{aligned} y = \mu + a_i + m_j + s_k + p_l + d_n \\ + as_{ik} + ap_{il} + ad_{in} + ms_{jk} \\ + mp_{jl} + md_{jn} + sp_{kl} + sd_{kn} + pd_{ln}, \end{aligned} \tag{5}$$

where,

$$\begin{aligned} y &= \text{reconstruction error}, \\ \mu &= \text{overall average error}, \\ xy_{i,j} &= \text{interaction term between factor } x_i \text{ and } y_j. \end{aligned}$$

This model, Eq. (5), contains both linear and interaction terms, meaning the model reflects both factor influence as independent and as cross effects, e.g. as_{ik} is the interaction term for 'algorithm' and 'animatronics'. For each term, we test for significance by choosing between two hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: c_0 = c_1 = \ldots = c_N \\ \mathcal{H}_1 &: c_0 \neq c_1 \neq \ldots \neq c_N \end{aligned} \tag{6}$$

with c_n being a term from (5) e.g. a_i or md_{jn}. The value $\mathcal{H}_0$ is typically referred to as the null hypothesis, meaning the term c_n has no significant effect. ANOVA allows for estimating the probability of falsely rejecting the null hypothesis for each factor. This statistic is referred to as the p-value. A term is referred to as being statistically significant if it's p-value is below a certain threshold. In this paper we consider a significance threshold of 0.0005 or approximately 3.5σ. As such, we clearly evaluated which factors are important for NRS*f*M and which are not.

Another interesting property of the ANOVA is that all coefficients in a given factor sums to zero,

$$\sum_{i=0}^{N} c_i = 0. \tag{7}$$

So each factor can be seen as adjusting the predicted reconstruction error from the overall average. It should be noted that the 'algorithm'/'camera model' interaction am_{ij} has been left out of (5) due to MultiBody [33] only being tested with one camera model.

The error model of (5) is not directly applicable to the error of all algorithms as not all state-of-the-art methods from TABLE 1 can deal with missing data. As such we perform the evaluation in two parts. One where we disregard missing data and include all available methods from TABLE 1, and one where we use the subset of methods that handle missing data and utilize the full model of (5). The former is covered in Sec. 5.1 and the latter is covered in Sec. 5.2.

TABLE 2
ANOVA table for NRS*f*M reconstruction error without missing data. Sources as as defined in (5). All factors are statistically significant at a 0.0005 level except ms_{jk} and mp_{jl}.

Factor	Sum Sq.	DoF	Mean Sq.	F	p-value
a_i	3.6×10^5	15	2.4×10^4	204.8	5.5×10^{-242}
m_j	1.1×10^4	1	1.1×10^4	90.4	3.2×10^{-20}
s_k	1.0×10^5	4	2.6×10^4	219.0	3.6×10^{-121}
p_l	1.5×10^4	5	3.0×10^3	25.6	9.3×10^{-24}
as_{ik}	4.1×10^4	60	6.9×10^2	5.9	2.9×10^{-33}
ap_{il}	4.1×10^4	75	5.5×10^2	4.7	2.3×10^{-28}
ms_{jk}	1.3×10^3	4	3.2×10^2	2.7	0.03
mp_{jl}	1.8×10^3	5	3.6×10^2	3.1	0.0086
sp_{kl}	1.1×10^4	20	5.7×10^2	4.9	2.3×10^{-11}
Error	8×10^4	689	1.2×10^2		
Total	7×10^5	878			

TABLE 3
Linear term $\mu + a_i$ sorted in ascending numerical order, this is the average error for the given algorithm. Algorithms are referred to by their alias in TABLE 1. All numbers are given in millimeters.

MultiBody	**KSTA**	**RIKS**	**CSF2**
29.36	31.94	32.21	32.83
MetricProj	**CSF**	**Bundle**	**PTA**
34.09	41.19	46.66	46.80
ScalableSurface	**EM PPCA**	**SoftInext**	**BALM**
53.88	59.19	61.94	66.34
MDH	**Compressible**	**SPFM**	**Consensus**
70.34	79.18	85.34	94.61

5.1 Evaluation without missing data

In the following, we discuss the results of the ANOVA without taking 'missing data' into account, using the model as in Eq. (5) without terms related to d_n:

$$\begin{aligned} y = \mu + a_i + m_j + s_k + p_l + as_{ik} \\ + ap_{il} + ms_{jk} + mp_{jl} + sp_{kl}. \end{aligned} \tag{8}$$

The results of the ANOVA using Eq. (8) is summarized in TABLE 2. All factors except ms_{jk} and mp_{jl} are statistically significant. As such, we can conclude that all the aforementioned factors have a significant influence on the reconstruction error. Therefore, we will explore the specifics of each factor in the following, starting with 'algorithm'.

TABLE 3 shows the average reconstruction error for each algorithm. The method MultiBody [33] has the lowest average reconstruction error over all experiments followed by KSTA [11] and RIKS [27]. For more detailed insights refer to TABLE 4 showing the 'algorithm'/'animatronic' dependent reconstruction error. As it can be seen, MultiBody [33] does not have the lowest error for all animatronics, as e.g. KSTA [11] has significantly lower error on the Tearing and Articulated deformations. Both of these can roughly be described

TABLE 4
Interaction term $\mu + a_i + s_k + as_{ik}$. This is equivalent to the algorithms average error on each animatronic. Lowest error for each animatronic is marked with bold text. Algorithms are referred to by their alias in TABLE 1. All numbers are given in millimeters.

	Deflation	Tearing	Bending	Stretching	Articulated
MultiBody	***15.20***	24.81	***25.20***	***25.12***	56.45
KSTA	27.60	***20.78***	36.66	29.62	***45.05***
RIKS	24.10	21.37	35.04	32.07	48.49
CSF2	23.55	21.55	36.21	32.33	50.51
MetricProj	27.75	25.93	35.93	33.22	47.63
CSF	34.92	40.93	40.10	39.96	50.03
Bundle	39.36	29.47	43.07	49.96	71.44
PTA	35.75	34.49	51.81	47.93	63.99
ScalableSurface	34.60	47.95	53.82	59.40	73.65
EM PPCA	40.10	59.59	65.28	73.89	57.10
SoftInext	46.60	54.07	64.05	65.49	79.48
BALM	52.51	58.28	74.85	67.76	78.29
MDH	56.87	63.75	69.00	75.02	87.05
Compressible	61.62	71.06	79.66	79.08	104.47
SPFM	54.85	76.19	80.05	89.93	125.68
Consensus	66.96	83.07	83.51	95.62	143.90

TABLE 5
Interaction term $\mu + a_i + p_l + ap_{il}$. Algorithms are referred to by their alias in TABLE 1. All numbers are given in millimeters.

	Zigzag	Half Circle	Line	Flyby	Circle	Tricky
MultiBody	***19.48***	30.88	***28.52***	***29.72***	***15.37***	52.18
KSTA	24.35	29.36	33.56	34.65	26.57	43.17
RIKS	25.68	***26.76***	30.24	37.59	31.81	***41.21***
CSF2	28.22	28.25	28.96	36.58	31.02	43.96
MetricProj	26.48	30.67	32.37	34.88	31.36	48.79
CSF	31.90	40.17	46.39	34.53	34.65	59.49
Bundle	47.30	45.55	39.27	39.68	52.84	55.30
PTA	35.51	42.67	48.34	43.91	49.82	60.53
ScalableSurface	39.64	52.68	41.88	52.64	87.98	48.49
EM PPCA	52.96	54.71	58.29	55.76	76.01	57.43
SoftInext	51.38	58.32	49.13	62.58	89.06	61.17
BALM	62.61	59.87	72.22	56.73	73.06	73.55
MDH	75.10	60.50	71.77	67.89	79.33	67.46
Compressible	73.61	80.78	80.08	83.84	72.49	84.24
SPFM	85.53	86.09	82.53	88.33	82.68	86.88
Consensus	94.70	94.52	94.81	94.35	94.88	94.42

as rigid bodies moving relative to each other, and it would seem KSTA [11] is the best at handling these deformations.

Methods with a physical prior, like MDH [43] and SoftInext [44], seems not to perform very well, as is evident from tables 1, 4 and 5. MDH [43] is designed with an isometry prior, therefore one would expect it to perform well in the bending deformation. Indeed, while its interaction term as_{ik} has its lowest value for the bending deformation, the average reconstruction error is simply too high.

A similar trend can be observed in TABLE 5, which shows the 'algorithm'/'camera path' dependent reconstruction error. While MultiBody [33] has the lowest average error, it is surpassed in the Half Circle and Tricky 'camera path' by RIKS [27]. On the other hand, MultiBody has the lowest error under the Circle path by quite a significant margin.

From this analysis we can conclude that MultiBody performs the best on average, but is surpassed w.r.t. to certain camera paths and animatronic deformations by algorithms such as RIKS [27] and KSTA [11]. This also clearly indicates that one needs to control for both deformation type and camera motion in future NRS*f*M comparisons, as the above conclusion could be changed by choosing the right combination of camera path and deformation. On the other hand, these findings shows that NRS*f*M performance can be optimized by choosing the right camera path (e.g. zigzag) and the right algorithm for the deformation in question.

The camera model has a significant impact on reconstruction error, a trend that can be observed from TABLE 5. Two factors relate to the camera, 'camera path' and 'camera model'. TABLE 8 shows that there is a significant difference in average error w.r.t. camera path. It is interesting to note that the circle path has one of the highest average errors, only surpassed by the tricky camera path. The latter was specifically designed to be challenging, as such, it is surprising to find that the circle and tricky path's average error only differ by 3.08mm. In fact MultiBody [33] seems to be the only method that benefits from the circle type of camera path, as can be seen in TABLE 5. TABLE 6 shows the average error of reconstructions for an orthographic and a perspective camera model. As it can be seen, there is a difference

TABLE 6
Linear term $\mu + m_j$ sorted in ascending numerical order, this is the average error for the given camera model. All numbers are given in millimeters.

Orthographic	Perspective
50.52	57.72

TABLE 7
Linear term $\mu + s_k$ sorted in ascending numerical order, this is the average error for the given animatronic. All numbers are given in millimeters.

Deflation	Tearing	Bending	Stretching	Articulated
40.15	45.83	54.64	56.02	73.95

TABLE 8
Linear term $\mu + p_l$ sorted in ascending numerical order, this is the average error for the given camera path. All numbers are given in millimeters.

Zigzag	Half Circle	Line	Flyby	Circle	Tricky
48.40	51.36	52.40	53.35	58.06	61.14

TABLE 9
ANOVA table for NRS*f*M reconstruction error with missing data. Factors as as defined in (5). All factors are statistically significant at a 0.0005 level except ms_{jk}, mp_{jl} and md_{jn}.

Factor	Sum Sq.	DoF	Mean Sq.	F	p-value
a_i	1.3×10^{5}	8	1.6×10^{4}	90.9	7.7×10^{-108}
m_j	1.4×10^{4}	1	1.4×10^{4}	81.6	1.2×10^{-18}
s_k	7.5×10^{4}	4	1.9×10^{4}	106.5	3.8×10^{-73}
p_l	4.1×10^{4}	5	8.2×10^{3}	47.0	8.8×10^{-43}
d_n	1.6×10^{4}	1	1.6×10^{4}	89.8	2.7×10^{-20}
as_{ik}	1.6×10^{4}	32	5.0×10^{2}	2.9	3.4×10^{-7}
ap_{il}	5.6×10^{4}	40	1.4×10^{3}	8.0	6.4×10^{-37}
ad_{in}	1.1×10^{4}	8	1.3×10^{3}	7.5	1.1×10^{-9}
ms_{jk}	2.6×10^{3}	4	6.5×10^{2}	3.7	0.0052
mp_{jl}	2.5×10^{3}	5	5.1×10^{2}	2.9	0.013
md_{jn}	2.9×10^{2}	1	2.9×10^{2}	1.6	0.2
sp_{kl}	2.7×10^{4}	20	1.4×10^{3}	7.8	6.7×10^{-21}
sd_{kn}	3.6×10^{3}	4	8.9×10^{2}	5.1	0.00048
pd_{ln}	8.1×10^{3}	5	1.6×10^{3}	9.3	1.4×10^{-8}
Error	1.4×10^{5}	824	1.8×10^{2}		
Total	5.7×10^{5}	962			

TABLE 10
Interaction between camera path and missing data; $\mu + p_l + d_n + pd_{ln}$. Numbers are given in milimeters.

	Without Missing	With Missing
Zigzag	43.34	47.47
Half Circle	44.93	52.32
Flyby	46.63	53.53
Line	47.19	53.56
Circle	55.40	60.30
Tricky	54.96	76.56

TABLE 11
Interaction between animatronic and missing data; $\mu + s_k + d_n + sd_{kn}$. Numbers are given in milimeters.

	Without Missing	With Missing
Tearing	40.63	44.59
Deflation	37.20	49.47
Stretching	51.63	56.28
Bending	50.54	64.01
Articulated	63.72	72.10

of 7.20mm, which is significant but not as large as the difference w.r.t. 'algorithm' (TABLE 3) or 'camera path' (TABLE 8). This suggests that, while the error increases the state-of-the-art in NRS*f*M can still operate under a perspective camera model. This is quite interesting as most NRS*f*M are not designed with a perspective camera in mind. It would seem that an orthographic or weak-perspective camera acts a reasonable approximation on the scale of our animatronics.

There is also a significant difference between the average reconstruction error of each animatronic which TABLE 7 shows. Articulated has by far the highest average reconstruction error, making it the most difficult to reconstruct for the current state-of-the-art in NRS*f*M . Since most approaches use low-rank methods, a highly structured motion such as an articulation is difficult to handle with a low-rank prior, especially if points are densely sampled on all joints. On the other hand, deflation seems to be quite easy to handle for most of the state-of-the-art methods.

5.2 Evaluation with Missing Data

As previously mentioned, we are interested in 'missing data' and its effect on NRS*f*M . We, thus, here use Eq. (5), which is use to evaluate the subset of methods capable of handling missing data, as shown in TABLE 1. It should be noted that while MDH [43] is nominally capable of handling missing data, it has not been included in this part of the study. The reason being code provided only reconstructs frames with minimum ratio of visible data, thus our error metric cannot be applied. As such, we have 8 methods in total in this category.

We treat 'missing data' as a categorical factor having two states: with or without missing data. This is because the missing percentage of our occlusion-based missing data is dependent on the 'animatronic', 'camera path' and 'camera model' factors. Additionally, there is a significant sampling bias in the occlusion-based missing data. For example, in-plane motion, like articulated and tearing, rarely get a missing percentage above 25% and more volumetric motion such as deflation rarely go below 40% missing data. This would make it difficult to distinguish between the influence of the 'missing data' factor and the animatronic factor.

The results of the ANOVA is summarized in TABLE 9 and all factors except ms_{jk}, mp_{jl} and md_{jn} are statistically significant. This means that 'missing data' has a significant influence on the reconstruction error. TABLE 12 shows the interaction between 'algorithm' and 'missing data'. As expected, the mean error without missing data is very similar to the averages in TABLE 3 with KSTA [11] having the lowest expected error. However, with missing data, MetricProj [29] actually has a lower average reconstruction error. This is due to its low increase in error of 5.85mm when operating under occlusion-based missing data. In comparison, KSTA [11], CSF2 [10] and CSF [9] are much more unstable with average increases in error of 9.65mm, 18.15mm and 13.49mm respectively. Common for the three methods is that they assume a Discrete Cosine Transform (DCT) as their prior. Indeed, we see a similar increase for ScalableSurface of 16.52mm and this method also uses a DCT basis.

These results suggests that while DCT-based approaches are quite accurate without missing data, they are not very robust when operating under occlusion-based missing data. And, thus, they would likely not be very robust when applied to real-world deformations where occlusion-based missing data is unavoidable. This indicates that, future research should focus on making DCT basis methods more robust or to modify the DCT model to better generalise for 'missing data'. Finally, BALM [30] method exhibit some peculiar behavior as its average error actually decreases by 3.33mm, contrary to expectation.

TABLE 11 shows the average error as an interac-

TABLE 12
Interaction between 'algorithm' and 'missing data'; $\mu + a_i + d_n + ad_{in}$. This is the average error for each algorithm either with or without occlusion-based missing data.

	KSTA	MetricProj	CSF2	CSF	Bundle	ScalableSurface	EM PPCA	BALM
Without Missing	***31.94***	34.09	32.83	41.19	46.66	53.88	61.22	66.34
With Missing	41.59	***39.94***	50.98	54.68	52.98	70.40	64.05	63.01

tion between 'animatronic' and 'missing data', i.e. the average reconstruction error of each animatronic with and without missing data. It is interesting to note that the in-plane deformations, i.e. Tearing, Stretching and Articulated, generally have a smaller increase in error with missing data compared to the more volumetric deformation, i.e. Deflation and Bending, compared to the error without missing data. The increase is respectively 3.96mm, 4.65mm and 8.38mm versus 12.27mm and 13.47mm. The main difference between the two groups is that the ratio of missing data is consistently low for the in-plane deformations. This would suggest that the ratio of missing data has an impact on the reconstruction error.

TABLE 10 shows the average error as interaction between 'camera path' and 'missing data'. The Tricky path has by far the highest average error. This is expected, as the small camera movement ensures that a portion of the tracked points is consistently hidden. As such, while Tricky and Circle were approximately equally difficult without missing data, this is no longer the case with missing data as Circle's average error only increases by 4.9mm. Indeed, all other camera paths have approximately the same increase in error with missing data. These paths also ensure that all observed points are equally visible. So while the camera paths nominally have approximately the same percentile of missing data as the Tricky path, the spatio-temporal distribution is different. These results suggests that the distribution of missing data is as important as the ratio in affecting the reconstruction error. Indeed this is in line with the observations made by Paladini et al. [29].

The aforementioned observations demonstrates the importance of testing against occlusion-based missing data as it contains a spatio-temporal structure of missing data that a randomly removed subset lacks. Many NRS*f*M methods treats missing data as a matrix fill-in problem, meaning recreating missing values from interpolation of spatio-temporally close observations. Thus, it is easy to see that conceptually it is much easier to interpolate random, evenly distributed missing data, compared to the spatio-temporally clustered structure of occlusion-based missing data. It is noted, that KSTA [11] and CSF [9] were both evaluated using random subset missing data in the original works, and was found to approximately have the same reconstruction whether from 0% to 50% missing data. These results are obviously quite different from the conclusion of our study and we hypothesize, that the spatio-temporal structure of our occlusion-based missing is probably the primary cause for this.

6 Discussion and Conclusion

To summarize our findings, we would like to firstly mention that, the algorithm with the lowest error on average without missing data was found to be MultiBody [33]. There is, however, a large variation between the different algorithms performance depending on the factors chosen. As such our study does not conclude that Multibody [33] is definitively better than all other methods in general. As an example, for some camera paths RIKS [27] had lower average error than MultiBody [33]. Also, with missing data MetricProj [62] has the lowest reconstruction error. Other observations include that methods with a DCT basis were found to have a great increase in error with occlusion-based missing data.

Our study also has findings that support and form hypotheses of where future NRS*f*M work could head. In relation to this, it should be mentioned, that even though some of these hypotheses have been stated before elsewhere, it is a strength of this work and our data set that it confirms these. Firstly, it is clear that methods using the weak perspective approximation to the perspective camera model only incur a small penalty for doing so on average. This camera model seems like a good approximation at first, although it should be noted, that our data set does not challenge the algorithms extremely in this regard, with only an average 1.6 fold change in the depth change.

Another main avenue of investigation was the effect of missing data. Here we found, that that this aspect has a large impact on on the reconstruction error. This is somewhat at odds with previous findings, and we speculate that this has to do with our missing data having structure originating from object self occlusion, as opposed to generate missing data with random sampling. In particular, occlusion-based missing data increases the reconstruction error of all methods except BALM [30]. Our study thus indicates this area to be a fruitful area of investigation for NRS*f*M research.

Another observation is that the physically based methods did quite poorly compared to the methods using a statistically based deformation model. This is in a sense counter intuitive, provided that the physical models capture the deformation physics well. This in turn, thus, lead us to the observation that stronger efforts could

be beneficial as far as better physical based deformation models.

As stated, many of these observations, support hypothesis held in the NRS*f*M community, and it strengths them, that we have here provided empirical support for them. On the other hand, this study also helps to validate the suitability of our compiled data set. In regard to which, it should be noted, both deformation types and camera paths have a statistically significant impact on reconstruction error, regardless of the algorithm used. This indicated that our proposed taxonomy and the data set design has value.

All in all, we have here presented a state of the art data set for NRS*f*M evaluation. We have applied 16 different NRS*f*M method to this data set. Methods that span the state of the art of NRS*f*M . This evaluation validates the usability of our proposed, and publicly available data set, and gives several insights into the current state of the art of NRS*f*M , including directions for further research.

REFERENCES

[1] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] H. Aanæs, A. Dahl, and K. Steenstrup Pedersen, "Interesting interest points," *International Journal of Computer Vision*, vol. 97, pp. 18–35, 2012.

[3] M. Salzmann and P. Fua, "Deformable surface 3d reconstruction from monocular images," *Synthesis Lectures on Computer Vision*, vol. 2, no. 1, pp. 1–113, 2010.

[4] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[5] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *International Conference on Computer Vision and Pattern Recognition*, pp. 690–696, June 2000.

[6] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization approach," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[7] I. Akhter, Y. Sheikh, S. Khan, T. Kanade, *et al.*, "Nonrigid structure from motion in trajectory space," in *Neural Information Processing Systems (NIPS 2008)*, 2008.

[8] I. Akhter, Y. Sheikh, S., and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2011.

[9] P. F. U. Gotardo and A. M. Martinez, "Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 2051–2065, 2011.

[10] P. F. U. Gotardo and A. M. Martinez, "Non-rigid structure from motion with complementary rank-3 spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[11] P. F. U. Gotardo and A. M. Martinez, "Kernel non-rigid structure from motion," in *IEEE International Conference on Computer Vision*, 2011.

[12] J. Valmadre and S. Lucey, "General trajectory prior for non-rigid reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] I. Akhter, T. Simon, S. Khan, I. Matthews, and Y. Sheikh, "Bilinear spatiotemporal basis models," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 2, p. 17, 2012.

[14] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh, "Kronecker-markov prior for dynamic 3d reconstruction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2201–2214, 2017.

[15] M. Dawud Ansari, V. Golyanik, and D. Stricker, "Scalable dense monocular surface reconstruction," *International Conference on 3DVision*, 2017.

[16] L. Torresani, A. Hertzmann, and C. Bregler, "Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

[17] M. Brand and R. Bhotika, "Flexible flow for 3d nonrigid tracking and shape recovery," in *International Conference on Computer Vision and Pattern Recognition*, pp. 315–22, December 2001.

[18] A. Del Bue, X. Lladó, and L. Agapito, "Non-rigid face modelling using shape priors," in *IEEE International Workshop on Analysis and Modelling of Faces and Gestures* (S. G. W. Zhao and X. Tang, eds.), vol. 3723 of *Lecture Notes in Computer Science*, pp. 96–107, Springer-Verlag, 2005.

[19] A. Del Bue, "Adaptive non-rigid registration and structure from motion from image trajectories," *International Journal of Computer Vision*, vol. 103, pp. 226–239, June 2013.

[20] S. I. Olsen and A. Bartoli, "Implicit non-rigid structure-from-motion with priors," *Journal of Mathematical Imaging and Vision*, vol. 31, no. 2, pp. 233–244, 2008.

[21] H. Aanæs and F. Kahl, "Estimation of deformable structure and motion," in *In Workshop on Vision and Modelling of Dynamic Scenes, ECCV02*, 2002.

[22] A. Del Bue, F. Smeraldi, and L. Agapito, "Non-rigid structure from motion using ranklet–based tracking and non-linear optimization," *Image and Vision Computing*, vol. 25, pp. 297–310, March 2007.

[23] A. Del Bue, X. Llado, and L. Agapito, "Non-rigid metric shape and motion recovery from uncalibrated images using priors," in *International Conference on Computer Vision and Pattern Recognition*, 2006.

[24] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion," in *International Conference on Computer Vision and Pattern Recognition*, 2008.

[25] S. Brandt, P. K. ad J. Kannala, and A. Heyden, "Uncalibrated non-rigid factorisation with automatic shape basis selection.," in *Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, 2011.

[26] C. Kong and S. Lucey, "Prior-less compressible structure from motion," June 2016.

[27] O. C. Hamsici, P. F. Gotardo, and A. M. Martinez, "Learning spatially-smooth mappings in non-rigid structure from motion," pp. 260–273, Springer, 2012.

[28] G. Wang, H. Tsui, and Q. Wu, "Rotation constrained power factorization for structure from motion of nonrigid objects," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 72–80, 2008.

[29] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Optimal metric projections for deformable and articulated structure-from-motion," *International Journal of Computer Vision (IJCV)*, vol. 96, pp. 252–276, 2012.

[30] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini, "Bilinear modeling via augmented lagrange multipliers (balm)," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, pp. 1496 –1508, August 2012.

[31] Y.-X. Wang, C. M. Lee, L.-F. Cheong, and K.-C. Toh, "Practical matrix completion and corruption recovery using proximal alternating robust subspace minimization," *International Journal of Computer Vision*, vol. 111, no. 3, pp. 315–344, 2015.

[32] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 101–122, 2014.

[33] S. Kumar, Y. Dai, and H. Li, "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion," *Pattern Recognition*, 2017.

[34] M. Lee, J. Cho, and S. Oh, "Procrustean normal distribution for non-rigid structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1388–1400, July 2017.

[35] J. Cho, M. Lee, and S. Oh, "Complex non-rigid 3d shape recovery using a procrustean normal distribution mixture model," *International Journal of Computer Vision*, vol. 117, no. 3, pp. 226–246, 2016.

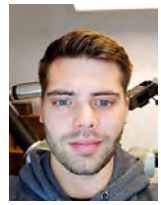
[36] A. Varol, M. Salzmann, E. Tola, and P. Fua, "Template-free monocular reconstruction of deformable surfaces," in *International Conference on Computer Vision*, pp. 1811–1818, 2009.

[37] J. Fayad, L. Agapito, and A. Del Bue, "Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences," in *European Conference on Computer Vision*, 2010.

[38] C. Russell, J. Fayad, and L. Agapito, "Energy based multiple model fitting for non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
[39] M. Lee, J. Cho, and S. Oh, "Consensus of non-rigid reconstructions," pp. 4670–4678, 2016.
[40] J. Taylor, A. D. Jepson, and K. N. Kutulakos, "Non-rigid structure from locally-rigid motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.
[41] A. Chhatkuli, D. Pizarro, and A. Bartoli, "Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity.," in *BMVC*, 2014.
[42] S. Parashar, D. Pizarro, and A. Bartoli, "Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
[43] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli, "Inextensible non-rigid structure-from-motion by second-order cone programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
[44] S. Vicente and L. Agapito, "Soft inextensibility constraints for template-free non-rigid reconstruction," in *European Conference on Computer Vision*, pp. 426–440, 2012.
[45] A. Del Bue and A. Bartoli, "Multiview 3d warps," in *International Conference on Computer Vision*, pp. 675–682, 2011.
[46] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel, "Sequential non-rigid structure from motion using physical priors," IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 979–994, 2016.
[47] A. Agudo and F. Moreno-Noguer, "Force-based representation for non-rigid shape and elastic model estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
[48] L. Torresani, D. Yang, E. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," in *International Conference on Computer Vision and Pattern Recognition*, 2001.
[49] J. Hyeong Hong, C. Zach, and A. Fitzgibbon, "Revisiting the variable projection method for separable nonlinear least squares problems," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
[50] J. Xiao and T. Kanade, "Uncalibrated perspective reconstruction of deformable structures," in *IEEE International Conference on Computer Vision*, pp. 1075–1082, October 2005.
[51] R. Vidal and D. Abretske, "Nonrigid shape and motion from multiple perspective views," in *European Conference on Computer Vision*, pp. 205–218, Springer, 2006.
[52] G. Wang, H.-T. Tsui, and Z. Hu, "Structure and motion of nonrigid object under perspective projection," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 507–515, 2007.
[53] R. Hartley and R. Vidal, "Perspective nonrigid shape and motion recovery," in *European Conference on Computer Vision*, pp. 276–289, 2008.
[54] X. Lladó, A. Del Bue, and L. Agapito, "Euclidean reconstruction of deformable structure using a perspective camera with varying intrinsic parameters," in *Proc. International Conference on Pattern Recognition*, (Hong Kong), 2006.
[55] X. Lladó, A. Del Bue, and L. Agapito, "Non-rigid metric reconstruction from perspective cameras," *Image and Vision Computing*, vol. 28, no. 9, pp. 1339–1353, 2010.
[56] C. M. University, "Cmu graphics lab motion capture database," 2002.
[57] L. Torresani, A. Hertzmann, and C. Bregler, "Learning non-rigid 3D shape from 2D motion," in *Advances in Neural Information Processing Systems 16* (S. Thrun, L. Saul, and B. Schölkopf, eds.), Cambridge, MA: MIT Press, 2004.
[58] A. Del Bue, X. Lladó, and L. Agapito, "Non-rigid face modelling using shape priors," in *AMFG*, pp. 97–108, Springer, 2005.
[59] H. Aanæs, R. Jensen, G. Vogiatzis, E. Tola, and A. Dahl, "Large-scale data for multiple-view stereopsis," *International Journal of Computer Vision*, pp. 1–16, 2016.
[60] Deutsches Institut fr Normung, "VDI 2634: Optical 3-D measuring systems. Optical systems based on area scanning," tech. rep., Deutsches Institut fr Normung, 2012.
[61] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.
[62] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *International Conference on Computer Vision and Pattern Recognition*, 2009.
[63] P. F. Velleman and D. C. Hoaglin, *Applications, basics, and computing of exploratory data analysis*. Duxbury Press, 1981.
[64] D. F. Williamson, R. A. Parker, and J. S. Kendrick, "The box plot: a simple visual method to interpret data," *Annals of internal medicine*, vol. 110, no. 11, pp. 916–921, 1989.
[65] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.
[66] J. C. Gower and G. B. Dijksterhuis, *Procrustes problems*, vol. 30. Oxford University Press on Demand, 2004.
[67] G. A. Seber and A. J. Lee, *Linear regression analysis*, vol. 936. John Wiley & Sons, 2012.

Sebastian Hoppe Nesgaard Jensen is a Ph.D. student employed at the Image Analysis and Computer Graphics Department at the Technical University of Denmark. A technical expert that has worked extensively to build different datasets and with the robotic setup used for data acquisition.

Alessio Del Bue is the head of the Visual Geometry and Modelling (VGM) Lab at Istituto Italiano di Tecnologia (IIT), Genova, Italy. Starting his research on NRSfM in 2004, he contributed to the field with novel non-linear optimization methods, shape priors and studies over the motion manifold of rigid, non-rigid and articulated objects.



Mads Emil Brix Doest is a Ph.D. student employed at the section for Image Analysis and Computer Graphics, at the Technical University of Denmark. His research is focused on optical scanners and appearance acquisition.

Henrik Aanæs is associate professor in computer vision at the Technical University of Denmark, where he is, among others, heading and effort for making large high quality data sets for 3D computer vision. His interests mainly lye in 3D computer vision, and their application, where he has e.g. also worked with NRS*f*M .

APPENDIX B

An Adaptive Robotic System for Doing Pick and Place Operations with Deformable Objects

An Adaptive Robotic System for Doing Pick and Place Operations with Deformable Objects

Troels Bo Jørgensen[1], Sebastian Hoppe Nesgaard Jensen[2], Henrik Aanæs[2], Niels Worsøe Hansen[3] and Norbert Krüger[1]

[1]*Maersk McKinney Møller institute, University of Southern Denmark, 5230 Odense M, Denmark, trjoe@mmmi.sdu.dk*
[2]*DTU Compute, Technical University of Denmark, 2800 Kongens Lyngby, Denmark, snje@dtu.dk, aanes@dtu.dk*
[3]*Danish Meat Research Institute, Danish Technological Institute, 2630 Taastrup, Denmark, nwh@dti.dk*

Abstract

This paper presents a robot system for performing pick and place operations with deformable objects. The system uses a structured light scanner to capture a point cloud of the object to be grasped. This point cloud is then analyzed to determine a pick and place action. Finally, the determined action is executed by the robot to solve the task. The robotic placement strategy contains several free parameters, which should be chosen in a context-specific manner. To determine these parameters we rely on simulation-based optimization of the individual use cases. The entire system is tested extensively in real world trials. First, the reliability of the grasp is evaluated for 7 different types of pork cuts. Then the validity of the simulation-based optimization of the placement strategy is evaluated for 2 of the most different pork cuts, to show the generality of the overall approach.

Keywords: Robotic Manipulation, Deformable Objects, Structured Light Scanner, Vision-based Meat Analysis, Simulation-based Optimization

1. Introduction

Minimizing setup times for industrial robotic systems is an important task for incorporating robots in small batch production, since designing and integrating the system is a relatively large part of the total expense in this type of production. In this paper, we focus on meat handling, where we investigate the possibilities for using robots to execute pick and place operations of meat pieces. The challenge is that there are a lot of different cuts of meat, and special solutions have to be designed for each case. Thus it is important that a procedure is formulated, which can help to design robotic solutions for as many cases as possible in a reasonable amount of time.

We approach this problem from two directions. First, we design a general and adaptable hardware setup for doing pick and place operations with meat. Secondly, we present a simulation-based optimization framework for designing and fine tuning the solution in simulation.

The hardware setup is shown in Fig. 1 and it consists of a 6 axis robot arm, a suction-based gripper tool and a vision system. The gripper tool can be adapted to a specific task and it is designed to cope with the high variation when grasping deformable objects. The vision system is also designed to be a generic solution for detecting and segmenting meat surfaces. It uses stereoscopic structured light for 3D surface reconstruction, which has been shown to generate precise point clouds, even when scanning materials with high levels of subsurface scattering [1]. Furthermore, we apply a generic region growing method to segment the individual meat

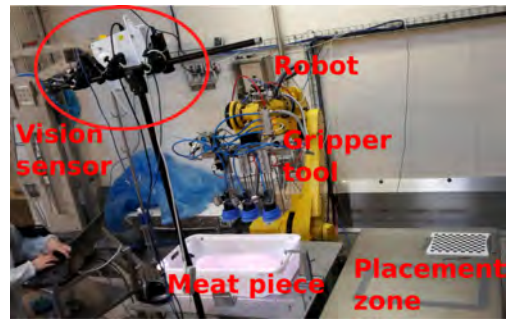


Fig. 1: The physical setup.

surfaces, which we have applied successfully to different cuts of pork. The segmented point cloud is then used to generate a robotic action for lifting and placing the meat. This action is also designed such that it can be adapted to specific use cases.

To adapt the generic solutions to specific use cases, we have designed a simulation tool for modeling the robotic meat handling operations. Furthermore, the simulation framework enables the user to analyze the robustness of the solutions. This is achieved by doing hundreds of simulations with different perturbations of system uncertainties, e.g. the meat size, to ensure the system works even for products with high variation, such as meat. We parameterized the generic solutions, such that they can be tuned for the specific problems using numeric optimization. The optimization is done based on the simulation framework, such that ten thousands of simulations are used to determine good system parameters. After good parameters are found in simulation, they are implemented and evaluated in the real world.

The main contribution of this work lies in combining several technologies, in order to design a robot solution for handling pork in a physical prototype at a Danish slaughterhouse. The individual technologies have been published in various conference proceedings. The gripper and the grasping strategy used in this work was introduced in [2]. This is extended by parameterizing the grasp action and introducing a placement strategy. The simulation framework used was introduced in [3]. In our work, this framework is extended to model the use cases addressed in this paper. Lastly, the optimization approach is based on work presented in [4].

The paper is structured as follows: First we discuss relevant litterature addressing the three main components in Section 2. These components are vision solutions for segmenting meat, robot systems for handling meat and optimization techniques relevant for robotic systems. The overall system is described in Section 3. The vision system for generating and segmenting the point clouds is described in Section 4. In Section 5, we introduce the gripper tool and discuss the procedure for generating a robotic action based on the point cloud. The case specific tuning of parameters is split into two parts. First, we introduce the simulation tool in Section 6 and then the optimization process is described in Section 7. In Section 8, we test the solutions with different cuts of pork. Lastly, we conclude on the results in Section 9.

2. Related Work

In this paper, 3 key topics are addressed. The first topic is vision based segmentation of meat surfaces. The second topic is robotic systems for manipulating deformable objects such as meat. Lastly, optimization based parameter tuning of robotic systems is addressed.

2.1. Vision Systems for Analysing Deformable Objects

3D reconstruction and simulation of deformable objects and has been studied intensively for years. A recent example would be [5] where cloth is handled dynamically by a humanoid robot. Here a control algorithm is fed input data from a Kinect that supplies both color and depth. Similar approaches can be found in [6], [7] and [8]. Common for these is that the object of interest is distinct and easily segmented from its environment. As such they are not directly applicable to our problem domain. This is because we have to handle boxes of meat with multiple pieces of meat in a pile. For this reason, depth data supplied is too inaccurate to properly segment each piece.

One needs to look no further than the DAVIS challenge [9] to see the tremendous progress and challenge of object segmentation. Some researchers have proposed to used convolutional neural network [10][11], others pursue other strategies such as region augmentation via Gaussian mixture models (GMM) [12]. However, while they focus and succeed at segmenting a single primary object, they do not consider a cluttered scenario as our system will have to deal with.

Our contribution will be applying high accuracy depth from structured light and a simple, yet powerful segmentation algorithm to obtain depth data for each piece of meat. The superior accuracy [13] of stereoscopic vision enables us to distinguish individual pieces, something that would likely be impossible with the Kinect.

2.2. Robotic Solutions for Handling Meat

While a huge body of work has addressed pick and place operations for rigid objects, only limited research has addressed these operations for deformable objects, such as meat. One example is [6] who developed a robotic system for handling silicon elements which was used as a more test friendly replacement for meat. They both addressed peg-in-hole operations and laying down operations of deformable objects with their system. In this work, we focus on real world cases and use substantially different equipment to address the grasping challenges of real meat products.

For related tasks such as cutting and separation of meat pieces research has been done in [14], [15] and [16]. Long et al. [14] proposed a system using three robots, one for moving the vision system, one for holding the meat and one for cutting the meat. Furthermore, they developed a simulator for modeling the deformable meat handling operation. Nabil et al. [15] proposed a similar system, but focused more on physically accurate simulation of the use case. Our proposed approach similarly rely on simulation-based analysis of the problem. However, we focus on modeling the interactions between the meat and its surroundings rather than just the interaction with a knife. We also use numeric optimization to tune the solutions in simulation, rather than just using it as a virtual test bed.

The researchers behind GRIBBOT [16] developed an automation solution for separating chicken fillet from a carcass. Their system consists of a vision solution, a 6-axis robot and a gripper tool for grasping and separating the chicken fillet. They also show how incorporating compliance in the gripper tool can make the solution robust to uncertainties from the vision system. Our system contains the same components and we also use compliance in the gripper to handle uncertainties and variation in the meat products. However, we focus on more general solutions for handling multiple tasks.

In terms of placing the deformable meat pieces, a closely related field is draping operations for cloth. To solve this problem, Balaguer et al. [17] proposed to combine reinforcement learning and learning by demonstration to train a robot system to fold a towel. Other researchers have shown how visual servoing can be used to fold cloth [18]. In our work, we also deal with fairly flat objects where draping operations to some level are necessary to achieve a nice placement. The pork bellies handled in our work is more rigid, which makes them easier to place and thus we can utilize simpler placement strategies. However, the individual products vary more and therefore it is necessary with a placement operation that is robust to the product variation. To achieve this, our work focusses more on determining robust placement actions based on optimization.

In terms of robotic solutions, our main contributions are a novel gripper tool and strategies for grasping and placing the meat based on point clouds from the vision system.

2.3. Numeric Optimization of Robotic Systems

Numeric optimization has been applied to several robotic problems to determine stable solutions based on real world trials [19, 20, 21, 22]. However, limited work has addressed simulation-based optimization of robotic solutions, where the systems are tested in simulation rather than the real world. The advantage of simulation-based optimization is that the number of real world trials can be heavily reduced. Besides speeding up the integration process, this also reduces the chance that real products are damaged during the test phase. When handling meat products, this is especially useful since the meat products have to be changed often to avoid contamination and health hazards. Thus testing in simulation can make the test phase substantially cheaper. Furthermore, it is often easier to set-up experiments and adjust various hardware settings in simulation compared to doing it in the real world, as we demonstrated in [4].

Buch et al. [23] proposed to use simulation-based optimization to determine robotic action parameters for executing a peg-in-hole operation. In their work, they only optimize 2 parameters. Thus they are able to use brute-force like methods to determine a good parameter set. Bodenhagen et al. [6] also rely on simulation-based optimization to tune their action for doing peg-in-hole and laying down operations with deformable objects. Their solutions again rely on only 2 and 3 parameters, and thus they are able to use brute-force like techniques. In our work, we rely on more parameters to define the solutions and thus we focus on optimization techniques that can deal with this in a computationally tractable manner.

Wolniakowski et al. [24] focus on optimizing gripper design in simulation. To achieve this gradient descent based methods are used to determine 11 parameters specifying the gripper fingers. In our work, 12 parameters are optimized, so the scope of the problems are similar. However, we focus on using optimization based on function fitting, in particular "RBFopt" [25], since earlier work [26] indicated this technique is more suitable for this type of optimization problem.

One of the robotic problems that have been optimized based on real world trials is maximizing the walking speed of bipedal robots [20, 21]. In both approaches optimization based on function fitting is used to determine the parameters that result in the fastest robots. Similarly Tesch et al. [22] optimize the speed of a snake-like robot. Again they show optimization based on function fitting have the best performance in terms of quickly optimizing their 7 free parameters.

Our main contributions in the field of parameter tuning is a new use case, where we show simulation-based optimization is suitable for designing robot solutions for handling deformable objects in an industrial setting.

3. Method

Robotic handling of meat is a challenge as few prior assumptions can be made in design. For example, we cannot design towards a specific shape and size as is common in contemporary robotics. Additional we do not have prior knowledge on the object's pose. As such the exact geometry and the pose must be acquired during the runtime of the system. One way to accomplish this is through 3D vision technology.

Physically moving the object requires adaptable automation. The 6-axis robot arm is ideal for this purpose as it gives us maximum freedom of movement. Furthermore, the robot arm must be equipped with a gripper that is flexible enough to handle the variation and deformation of the meat. The gripper should also be adaptable to different types of meat cuts. Either in runtime or after a short preparation stage.

Picking and placing are not trivial either, as the object of interested should be placed in a specific pose. The deformable nature of the meat pieces makes this need even more pressing. As such our system is equipped with a sophisticated path planning system for determining an appropriate grasp and placement action based on the vision input.

Our system can be roughly broken down into **Vision**, **Gripper** and **Planning** components. However, this alone is not enough as all components contain parameters that must be tuned to a given problem. A large part of the setup time goes to this tuning process. Therefore we have developed a **Simulation** framework, which can handle a huge chunk of the optimization in a virtual environment.

The entire system is illustrated in two flowcharts. The first (Fig. 2) describes the physical system. The diagram also indicates, which parameters are tuned using simulation-based optimization. The second flowchart (Fig. 3) indicates how the system parameters are optimized in simulation. The optimization happens in an iterative procedure, where different software and hardware parameters are tested to determine which produce the best result.

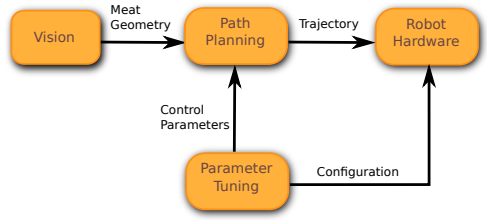


Fig. 2: Diagram of the system architecture. The upper three boxes constitute the runtime system.

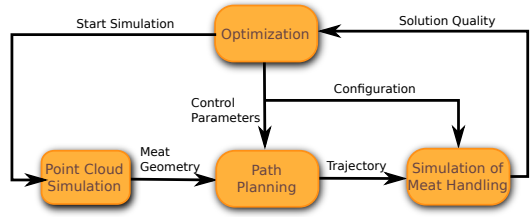


Fig. 3: Diagram of the simulation-based parameter optimization. This process describes how the control parameters and configuration are determined in the parameter tuning block of Fig. 2.

4. Vision

For the robot system to properly locate and handle the meat pieces, it must be supplied with 3D data. By far the most flexible way to achieve this is through vision technology. There has been a huge surge in 3D vision applications due to the wide availability of user friendly real-time scanners such as the Microsoft's Kinect and Intel's Real-Sense. While they are great, they have made a lot of sacrifices to reach real-time performance on a low-cost embedded platform. This means that the accuracy and precision of their 3D data is subpar. For example both versions of the Kinect has an accuracy in the near 1cm range [27].

We instead choose to go another route, by using a similar technology as employed in the above examples, but customized to our needs.

4.1. Structured Light Scanning

Structured light is an active 3D scanning technology that estimates depth via stereo triangulation [28]. The basic idea is the same as with passive stereo vision. By finding the same points projection in a stereo image pair and knowing the relative camera geometry, it is possible to infer the 3D position of that point. The first part is known as the correspondence problem and it can be quite challenging. In passive stereo non-unique and weak texture creates uncertainty which has to be resolved with e.g. statistical priors like spatial smoothness [29].

Instead of relying solely on material appearance, we can project light patterns onto the scene to create artificial texture. By building a certain structure into the projected pattern, the correspondence can be made a lot easier. Hence the name; structured light. There exist many different encoding strategies ranging from

the one-shot, speckle patterns of the Kinect and the Real-Sense to multi-pattern approaches such as Gray Codes [30] and Micro Phase-shifting [31]. First we will go over our hardware setup, afterwards we will discuss the specific structured light method used.

Our scanner consists of three components: two high-definition cameras and a light projector. Depth is estimated via stereo triangulation using the pixel disparities between the camera image pair. This is illustrated in Fig. 4.

In theory, triangulation could also be done between the projector and a camera. However, this requires a projector that has a very well-defined linear gamma curve. Most consumer projectors cannot be used here. By adding a second camera, we ease on the hardware requirement of the projector. This is due to that the projected pattern need no longer be accurately portraited, but that it simply has to be horizontally unique.

As mentioned, we project a series of patterns onto the scene and acquire a series of images from both cameras. Then the idea is to use these patterns to, as the name suggests, encode a continuous phase across the scene. This value can then be used to efficiently solve the correspondence problem. Formally we consider a situation with N projected patterns. Each pixel in each projected pattern should conform to the following spatio-temporal model,

$$I_i(x, y) = \sin\left(2\pi\left[\frac{i}{N} + \frac{\omega \cdot x}{w}\right]\right), \tag{1}$$

where i is the sequence number, ω is the spatial pattern frequency and w is the pattern width. The first term, $\frac{i}{N}$, defines the temporal component of the waveform and the second term, $\frac{\omega \cdot x}{w}$ defines the spatial component. The latter defines a constant, unique phase for each pixel. This is true for both the projected pattern and any image taken of it. We can use the acquired pattern series to estimate $\frac{\omega \cdot x}{w}$ for each pixel. Fig. 4 illustrates the overall process in the method.

We refer the interested to [28] for specific implementation details.

4.2. Segmentation

Of course, a point cloud generated by structured light is not particularly useful in itself. It must be segmented into meaningful parts before the information can be utilized in path planning. Specifically, we want each meat piece as separate segments. We accomplish this via a modified version of the region growing segmentation algorithm available in Point Cloud Lib [32][33]. Our version is shown in Algorithm 1. It grows a region from a

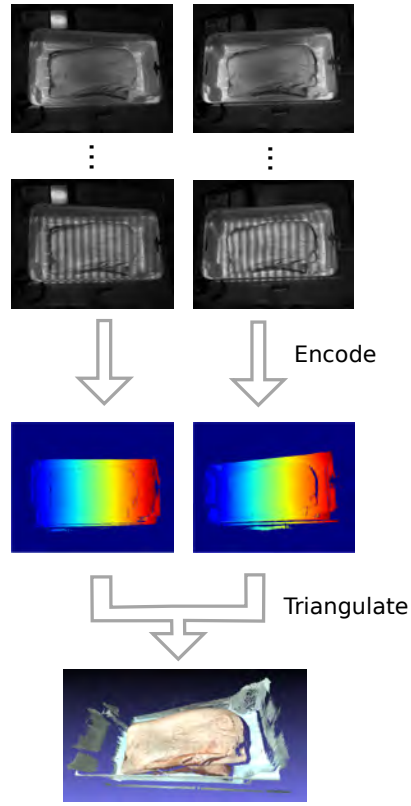


Fig. 4: Illustration of the encoding and triangulation flow of a structured light scan. The first row shows the scene, the second shows the encoded phase as per equation 1 and the final shows the triangulated point cloud.

Data:

$P =$ organized point cloud,
$N =$ organized point normals,
$C =$ organized point curvature,
$c_t =$ curvature threshold,
$\theta_t =$ angle threshold.

Result:

$R =$ list of segmented regions.

$\{w, h\} \leftarrow \text{size}(P)$;
$R \leftarrow \emptyset$;
$A \leftarrow \text{zeros}(w, h)$;
$S \leftarrow$ set of all points in P;
Sort S by ascending order of curvature;
while $S \neq \emptyset$ **do**
  $p_{min} = (x, y) \leftarrow$ *head of* S;
  $S \leftarrow S \setminus p_{min}$;
  if $A(x, y) = 1$ **then**
    continue;
  end
  $S_c \leftarrow \{p_{min}\}$;
  $R_c \leftarrow \emptyset$;
  while $S_c \neq \emptyset$ **do**
    $p_i \leftarrow$ *head of* S_c;
    $S_c \leftarrow S_c \setminus p_i$;
    $R_c \leftarrow R_c \cup p_i$;
    $B \leftarrow$ *8-neighbors of* p_i;
    for p_j *in* B **do**
      $\{x_j, y_j\} \leftarrow p_j$;
      $\{x_i, y_i\} \leftarrow p_i$;
      if $A(x_j, y_j) = 1$ **then**
        continue;
      end
      $a \leftarrow N(x_j, y_j) \cdot N(x_i, y_i)$;
      if $a < \cos\theta_t$ **then**
        continue;
      end
      $R_c \leftarrow R_c \cup p_j$;
      $A(x_j, y_j) \leftarrow 1$;
      if $c(x_j, y_j) < c_t$ **then**
        $S_c \leftarrow S_c \cup p_j$;
      end
    end
  end
  $R \leftarrow R \cup R_c$;
end

Algorithm 1: Segmentation via image space region growth.

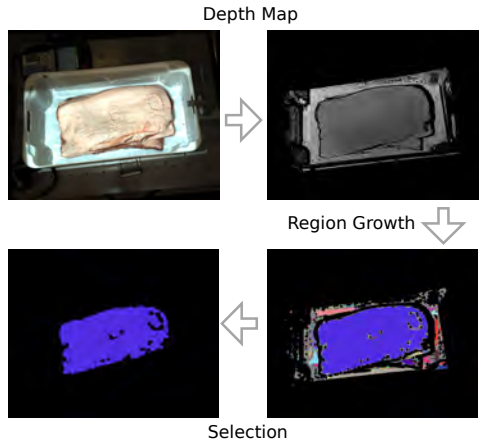


Fig. 5: Flow of the segmentation and selection process of the observed meat pieces.

point of low curvature and terminates at high curvature and change in normal angle. The idea is that an object's surface is relatively smooth and it's edge is characterized by corners or other high curvature forms.

The main difference lies in our algorithm being tailored specifically towards organized point clouds, meaning point clouds that are given in a 2D grid. This is the typical output format of e.g. the Kinect and our structured light scanner. The main performance limiter for a generic point cloud is the search for neighbors. This, along with various control logic, can be greatly sped up by exploiting the grid location of a given point. On an Intel Core i7-4610M it segments a point cloud of size 675x540 in 100ms-150ms.

After segmenting the point cloud, we must determine which is the next meat piece that should be handled. This is achieved by locating the five largest point clouds and selecting the top most point cloud of these. The process in its entirety is shown in Fig. 5.

5. Pick and Place Operations

In this work, we focus on a fairly general pick and place operation where multiple meat pieces are placed in a box and have to be moved to a conveyor belt. Furthermore, the meat should be placed stretched out such that it is ready for post-processing. Automating this task is a challenge as the meat is deformable and each cut varies significantly. To solve the task, two components are required: First, a hardware solution has to be de-

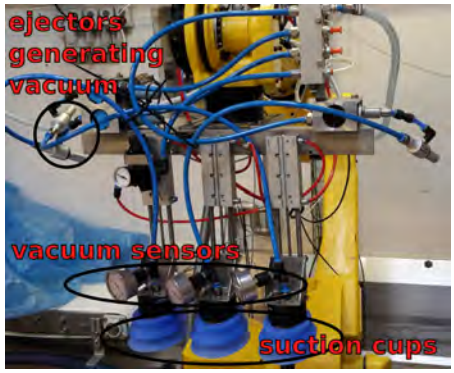


Fig. 6: The suction based gripper relies on ejectors for generating the vacuum, and it contains 3 sensors for measuring the pressure levels at each suction cup.

signed to move the meat. Secondly, a mechanical motion for lifting and placing the meat has to be generated.

As discussed in Section 1, a 6-axis robot with a flexible and adaptable suction based gripper tool is used to lift the meat. The gripper attached to the robot is discussed in Section 5.1.

Besides designing a hardware solution, a robotic motion for lifting and placing the meat also has to be developed. These motions are discussed in detail in Section 5.2 and 5.3 respectively.

Both these motion strategies are determined based on the vision input derived as discussed in Section 4. Furthermore as one might expect, the parameterization of the robotic hardware and motions contain several free parameters which have to be determined. In this work, these parameters are determined using simulation-based optimization as discussed in Section 6 and 7. A full list of the parameters are given in Table 1 and they are explained in detail in this section.

5.1. The Gripper

When designing the gripper tool, one of the key challenges is the high variation between each pick. Multiple aspects contribute to this variation. First of all, the size, shape and deformability of the meat pieces vary even within the same type of meat cut. Furthermore, the placement and deformed state also vary as each meat piece is placed differently in the box. Besides simply being flexible enough to handle one type of meat cut, the gripper should also be adaptable, such that it can be adjusted to handle different cuts. The gripper design used for addressing these challenges can be seen in Fig. 7 and the real gripper is shown in Fig. 6.

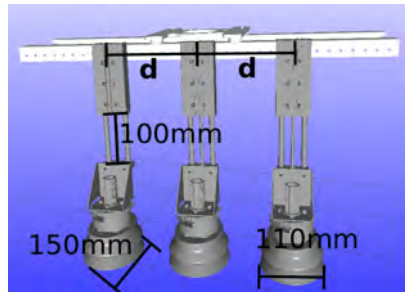


Fig. 7: A cad model of the gripper tool.

To grasp the meat, the gripper relies on suction cups, similar to [2]. The suction cups are flexible, so they can adapt to the local surface variation of the meat pieces. This is necessary to ensure that no air leaks into the vacuum chamber which would result in the suction cup dropping the meat. However, the local surface adaptation is not enough to deal with the larger variations that can occur across an entire meat piece. To address this, the suction cups are placed at the end of air pistons, which act as passive components much like if they were replaced with one-dimensional springs. These air pistons can be compressed a lot more than the suction cups, and enable the tool to adapt to larger deformation.

Besides being flexible, the gripper should also be adaptable such that it can grasp meat cuts of different sizes. To achieve this the distance between the suction cups, d, can be changed to match a particular meat cut. In this work, d is considered a control parameter which is optimized in simulation. A deeper discussion of the gripper design is given in [2].

5.2. The Rolling Grasp

The goal of the grasping strategy is to lift the meat robustly. A key challenge here is that a vacuum can form between the meat piece that is to be lifted and the piece below. If this vacuum becomes too strong, it will result in the grasp failing because the gripper lifts the two pieces sticking together.

To address this challenge a rolling lift was designed where the suction cups are placed close to the edge of the meat and lifted in a rolling motion, as illustrated in Fig. 9. This allows air to flow under the meat which increases the chance that the meat is separated from the piece below. The benefit of using this fairly complex grasp strategy over a simpler approach is demonstrated in [2].

The grasp based on the segmented point cloud of the meat discussed in Section 4, is generated in two stages:

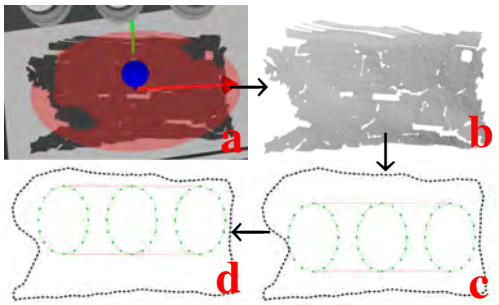


Fig. 8: Placement of the suction cups. a) A PCA is applied to the grey 3D point cloud. The red and transparent ellipsoid represents the eigenvector and eigenvalues of the PCA. The frame is the PCA frame. b) The point cloud is projected onto the PCA frame to generate a 2D point cloud. c) The black dots show the concave hull of the 2D point cloud that is used to represent the meat edge. The green dots represent the initial suction cup placement aligned with the PCA frame. d) The final suction cup placement is determined by minimize a regret score.

First, an acceptable suction cup placement is determined based on the segmented point cloud. This process is illustrated in Fig. 8. Then a robotic trajectory is defined to move the suction cups to the determined positions and lift the meat piece in a rolling motion. This motion is illustrated in Fig. 9.

Two challenges have to be addressed when placing the suction cups. First of all, none of the suction cups should stick outside the meat, if this happens air can flow into the vacuum chamber resulting in the suction cup dropping the meat. The second challenge is that the suction cups should be placed close to the edge of the meat to allow air to flow in and separate the meat piece from the piece below during the lift. To further increase this air flow, a large part of the meat edge should also be close to the suction cups. To address these challenges, the first step is to determine the edge of the meat piece. Secondly, the placement of the suction cups should be based on the edge and determined to address both of the mentioned challenges.

To determine the edge of the meat, the first step is to do a PCA of the segmented point cloud (Fig. 8a). Then the point cloud is projected onto the x,y-plane of the PCA frame (Fig. 8b). Finally, the edge can be determined as a concave hull of the projected 2D points (Fig. 8c). To generate the concave hull, the concave hull algorithm from PCL [33] is used. After the edge is determined, it is re-sampled to a resolution of 10mm to have a uniformly sampled edge model.

The next step is to determine the placement of the suction cups based on the edge model. This placement has to satisfy three conditions. First, the suction cups should be placed within the meat. Secondly, the suction cups should be placed close to the edge. Lastly, a large part of the meat edge should be close to the suction cups. To determine a placement that satisfies these conditions, we pose the problem as a minimization problem where a regret score is minimized. The regret score, R, captures how well the placement satisfies the conditions and it consists of two parts R_{cups} and R_{meat}. R_{cups} ensures that the suction cups are placed close to the edge while still being inside the meat. R_{meat} ensures that a large part of the meat edge is close to the suction cups. For the particular case where three oval suction cups are placed on a rectangular meat piece: R_{cups} favors that the suction cups are placed close to the long edge of the meat, while R_{meat} favors that the suction cups are placed close to the corners of the meat piece.

To control how close the suction cups and the meat edge should be, the control parameter d_{ideal} is introduced. d_{ideal} represents the ideal distance between the suction cups and the meat edge and it should be determined through simulation-based optimization. The regret score and the two subcomponents are given in (2), (3) and (4).

$$R_{\text{cups}} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}(min(\|\mathbf{s}_i - \mathbf{P}\|) - d_{\text{ideal}})^2, & \text{all } \mathbf{s}_i \text{ are inside the meat} \\ 1.0, & \text{otherwise} \end{cases} \tag{2}$$

$$R_{\text{meat}} = \begin{cases} \frac{1}{M}\sum_{j=1}^{M}\sqrt{|min(\|\mathbf{p}_j - \mathbf{S}\|) - d_{\text{ideal}}|}, & \text{all } \mathbf{p}_j \text{ are outside the suction cups} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$R = w \cdot R_{\text{cups}} + (1 - w) \cdot R_{\text{meat}}^4 \tag{4}$$

where $\mathbf{s}_i$ is a point on the suction cups. Each suction cup contains 16 points placed on the periphery, as illustrated by green dots in Fig. 8c and 8d. $\mathbf{P}$ is the meat edge. To determine R_{cups} the smallest distances from the suction cup points to the meat edge are squared, to favor that all the points on the suction cups are close to the edge.

$\mathbf{p}_j$ is a point on the meat edge. $\mathbf{S}$ is the suction cup edges. To determine R_{meat}, the square root of the smallest distances from points on the meat edge to the suction cups are used. This is done to ensure outliers do not dominate the score since some edge points will be far away from the suction cups. This can be seen in Fig. 8d, where there are many points on the meat edge (black dots) that are far away from the suction cups. This way

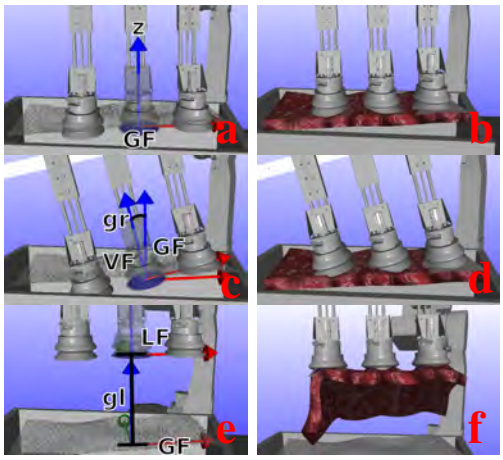


Fig. 9: Rolling lift - planning and simulation. a,b) show how the tool is aligned with the grasp frame after the grasp frame is mapped to the 3D world. c,d,e,f) illustrate the rolling motion that is used to lift the meat. The via frame, VF, and the lift frame, LF, are used to define the motion as indicated.

the score favors many inliers, over being close to every point.

Finally, the regret score is determined based on a tradeoff, w, between the two subcomponents. This tradeoff is a control parameter which should be optimized in simulation.

To determine a good suction cup placement based on the regret score, the minimization algorithm coordinate descent [34] is used. This algorithm moves the suction cups around in the 2D-plane, to find the placement with the lowest regret.

After the suction cup placement is determined, the next step is to determine the actual robot motion. The purpose of this motion is to lift the meat while avoiding a vacuum forming below it. This is achieved by lifting the meat in a rolling motion, such that air can flow in and separate the meat from the surface below. The motion is produced by the robot moving through three frames, which is illustrated in Fig. 9. The first frame is the grasp frame, GF, and it describes where the suction cups should be placed to grasp the meat. After the robot reaches the grasp frame the suction cups are activated to initiate the lift. Then the robot moves to the via frame, VF, which ensures the meat is lifted in a rolling motion. Lastly, the robot moves to the lift frame, LF, which ensures the meat is lifted well above the box.

The three frames are determined based on two control parameters named gr and gl, which should be optimized in simulation. Both gr and gl are illustrated in Fig. 9.

Fig. 10: Human meat placement. The meat is placed on the conveyor belt with a flat front facing the wrapping station.

The grasp frame is determined by reprojecting the optimal suction cup placement back into the 3D-world. The via frame is determined by rotating the grasp frame around the y-axis of the frame, by an angle specified by gr. Lastly, the lift frame is determined by translating the grasp frame in the z-direction by a distance specified by gl.

5.3. The Placement Operation

The goal of the placement strategy is to place the meat, such that it can be wrapped in folio by a wrapping station. To achieve this, the meat should be placed stretched out on the conveyor belt with a flat front facing the wrapping station, as illustrated in Fig. 10. This operation is fairly specific, but the placement criteria itself is common in the meat sector. E.g. it is a requirement if the meat is to be placed in boxes and for various cutting operations.

To enable the robot to place the meat in this manner, the placement strategy is designed to stretch the meat as it collides with the conveyor belt. This stretching is achieved by moving the gripper through two frames as it moves towards the final position over the conveyor belt. As the tool reaches these frames, the meat collides with the conveyor belt which stretches it. If the frames are picked reasonably, the meat is more likely to be placed in the desired fashion. When the gripper reaches the final position, the suction cups release the meat and the robot moves away. The entire placement strategy is illustrated in Fig. 11, and especially 11c shows how the collision with the conveyor belt can stretch the meat.

The three frames that define the robotic motion are named placement frame, PF, first approach frame, $AF1$, and second approach frame $AF2$. The placement frame specifies where the suction cups should be placed when the meat is released.

At the end of the rolling grasp (Fig. 9f) it can be seen that a large part of the meat hangs down to the left and at

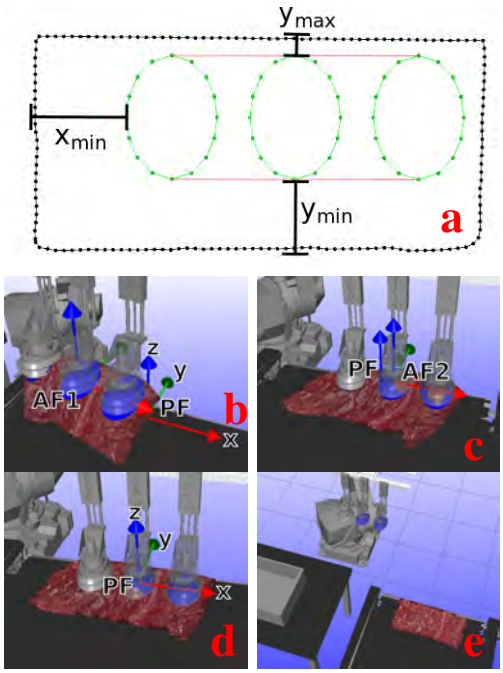


Fig. 11: Placement Action. a) illustrates the 2D placement of the suction cups used during the grasp. The distances x_{min}, y_{min} and y_{max} is used to determine the placement action. b) shows the tool moving to the first approach frame, $AF1$, and c) shows it moving to the second approach frame, $AF2$. d) shows the tool as it reaches the placement frame, PF, and finally in e) the vacuum is turned off and the robot is moved away.

the back of the suction cups. To ensure this is stretched out the first approach frame was introduced. The height of the frame is chosen, such that the corner of the meat to the left and at the back roughly touch the conveyor belt. Furthermore, the frame is moved slightly to the left and further back to ensure the meat is stretched as the robot moves towards the placement frame.

Initial trials using only the first approach frame and the placement frame resulted in the meat being twisted during the placement. The result was similar to the placement shown in Fig. 11c. This twist was corrected by introducing the second approach frame. This frame ensures the meat is dragged a bit too far, such that when it moves back to the placement frame the twist will be reduced as illustrated in Fig. 11d.

Both of the approach frames are dependent on several control parameters, which should be optimized in simulation to ensure a robust placement operation. These parameters are d_{x1}, d_{y1}, d_{z1}, d_{x2}, d_{z2}, θ and b. All parameters except b represent different offsets to the translation and rotations of the approach frames. b specify how much the meat hanging down at the edges of the suction cups should be considered in the translation of the first approach frame.

Mathematically the first approach frame is defined as the placement frame translated by $(x_{AF1}, y_{AF1}, z_{AF1})$ and rotated around the z-axis by θ_{AF1}, these values are given in (5), (6), (7) and (8).

$$x_{AF1} = -(x_{min} \cdot b + d_{x1}) \tag{5}$$

$$z_{AF1} = x_{min} \cdot b + d_{z1} \tag{6}$$

$$y_{AF1} = \begin{cases} -(y_{min} \cdot b + d_{y1}), & y_{min} > y_{max} \\ y_{max} \cdot b + d_{y1}, & \text{otherwise} \end{cases} \tag{7}$$

$$\theta_{AF1} = \begin{cases} -\theta, & y_{min} > y_{max} \\ \theta, & \text{otherwise} \end{cases} \tag{8}$$

where x_{min}, y_{min} and y_{max} are distances between the suction cups and the edge of the meat. These distances are illustrated in Fig. 11a, and they are used to ensure that the stretching of the meat is dependent on how much meat is hanging down at the edges of the suction cups.

The z_{AF1} translation ensures that the meat roughly touches the conveyor belt. The x_{AF1} translation ensures that the meat hanging down to the left of the suction cups is stretched. The y_{AF1} translation ensures the meat is stretched in the y direction as well. Whether the meat should be stretched in the positive or negative y direction depends on where the suction cups are placed on the meat. Finally, initial trials indicated that the meat is slightly rotated when the y translation is introduced. Therefore the rotation θ_{AF1} was added to reduce the other rotation.

The second approach frame, $AF2$, is defined as the placement frame translated $(d_{x2}, 0, d_{z2})$. The x translation is introduced to ensure the meat moves too far, such that it can move back to reduce the twist of the meat (Fig. 11c). The z translation is introduced to ensure the meat is not pushed too hard into the conveyor belt.

6. Simulation

To optimize the robot system in simulation, the first step is to construct a simulation framework for modeling robotic handling of the meat pieces. To achieve this two fundamental steps have to be taken. First of all, a deformable model of the meat piece should be created. Secondly, several models for the interaction between the meat and its surrounding has to be developed.

In this work, a mass-spring model is used to model the deformable meat pieces. This model consists of several particles, which motion is constrained by various springs placed between them. Throughout this paper the particles in the mass-spring model is refered to as meat particles. The model is described in detail in [3]. As discussed in [3], a spring-model was chosen over more complex finite element models, similarly to [35, 15]. The main reason for this is that spring-models tend to to be less computationally expensive. This is favorable since many simulations have to be conducted both to evaluate the robustness of potential solutions and to optimize the overall solution.

During the real scenario, four different mechanical interactions occur. First, a human places the meat in a box containing an arbitrary number of meat pieces. Then the robot pushes the unactivated suction cups into the meat. Next, the suction cups are activated, such that the robot can lift the meat. Finally, the meat is placed on the conveyor belt.

All these interactions are modeled by various constraints, which determine the motion of the meat particles. These constraints all represent the various surfaces the meat comes in contact with, and they come in three formats. The first is the *initial box constraint*, this represents the initial surface that the meat is placed on in the box, and it is discussed in Section 6.1. The second constraint is the *planar constraint*, this model the conveyor belt and the unactivated suction cups, and it is discussed in Section 6.2. The last constraint is a *vacuum constraint* which is used to attach the meat piece to the activated suction cups. This constraint is discussed in Section 6.3.

Besides modeling the motion of the meat particles, the motion of the suction cups also has to be modeled as discussed in Section 6.4. Furthermore, to generate the robotic action discussed in Section 5, a point cloud has to be rendered. This rendering process is discussed in Section 6.5. Lastly, the entire simulation scenario is discussed in Section 6.6.

6.1. Initial Box Constraint

The purpose of this constraint is to model the interaction between the meat piece and the box that the meat arrives in. This constraint should also capture that there might be several meat pieces below the top piece.

To model the uneven surface of a box full of meat a thin plate spline (TPS) [36] is used. This spline guarantees a smooth surface, and yet it can be highly randomized to capture many different initial conditions. The initial box surface, with a meat piece laying on it, is illustrated in Fig. 15.

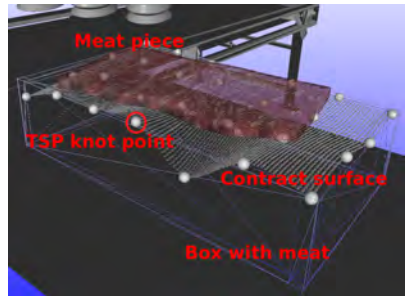


Fig. 12: The meat piece laying on the initial contact surface. The grey point cloud represents the thin plate spline, and the big grey points represent the knot points determining the shape. Furthermore, the meat piece is transparent and the box with the meat is represented as lines, to make the knot points of the thin plate spline fully visible.

The spline is defined based on 4×7 knot points, which determine the shape of the surface. The knot points are placed on a regular grid, which matches the shape of the box with the meat pieces. The height of the knot points is randomized, to roughly model that the meat pieces below are placed randomly.

If a meat particle moves through the thin plate spline, it is considered in contact with the initial box constraint. When this happens, the particles motion is fixed to the point of contact.

This constraint should capture two phenomena. The first is that the meat can not move through the surface of the box. The second is that a vacuum can form between two meat pieces placed in the box.

To model both these aspects, the meat particle is fixed to the point where it comes in contact with the surface. To move it two conditions have to be satisfied. The first condition is that air can flow below the meat particle. This is modeled by requiring that at least one of the neighboring meat particles is free of the initial box constraint. The second condition is that the meat can not move further into the surface. This is modeled by requiring that the meat particle is lifted.

6.2. Planar Constraint

The purpose of this constraint is to model the interactions between the meat piece and a planar surface. In this work, the un-activated suction cups and the conveyor belt is modeled with planar constraints.

The constraint has a planar surface and a 2D boundary shape, for the suction cups the shape is an ellipse, and for the conveyor belt it is a rectangle. In case a meat particle comes in contact with the constraint, a contact

Fig. 13: A planar constraint is used to ensure the meat does not fall through the conveyor belt.

point is added where the particle collides with the surface. In case the particle moves further into the surface, it is moved back to the contact point. In case it moves away from the surface the constraint is removed. When using the constraint to model a suction cup, the contact point moves along the surface of the suction cup. The constraint in action is illustrated in Fig. 13, where it keeps the meat from falling through the conveyor belt.

6.3. Vacuum Constraint

The purpose of this constraint is to model the interactions between the meat and the activated suction cups.

When the suction cups are activated in the real world, the meat is quickly attached to the suction cups. To model this in simulation, a contact volume is used to determine which meat particles are in contact with the suction cups. This volume is an elliptic cylinder that is formed by the surface of the suction cup ±5mm. When a suction cup is activated, all the meat particles within the contact volume is projected onto the surface of the suction cup. The meat particles are then fixed to these projected points until the suction cups are deactivated. The constraint in action can be seen in Fig. 14, where it constrains meat particles to the blue suction cup surface.

6.4. Suction Cups

Besides modeling the interaction between the meat and the suction cups, the motion of the suction cups themselves also has to be modeled. The model should capture the linear motion of the air pistons placed above the suction cups, and the local adaptation of the suction cups themselves.

This is achieved by modeling a suction cup as a planar elliptical mass placed at the end of a linear and angular spring, as illustrated in Fig. 14. The other side of the linear spring is attached to the gripper tool, which position is kinematically determined based on the robotic motion. The forces and torques affecting a suction cup

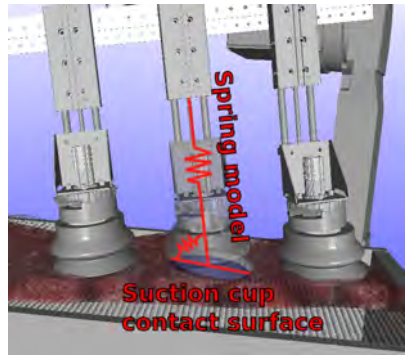


Fig. 14: The spring based suction cup model ensures the blue surface of the suction cup aligns with the meat during grasping. The middle suction cup and the meat are transparent to better visualize the blue surface of the middle suction cup.

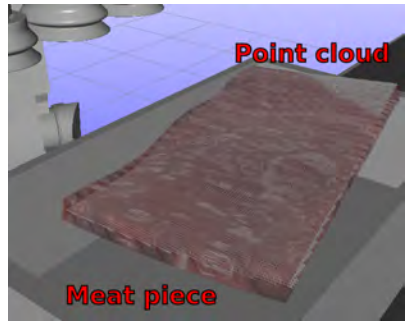


Fig. 15: The simulation of the grey point cloud is done using RobWork [37]. The meat piece is transparent such that the entire point cloud can be seen. Furthermore, the point cloud is slightly translated and rotated to model the effect of uncertainties in the vision system.

are determined based on the meat particles in contact with the suction cup.

6.5. Point Cloud Rendering

Besides modeling the mechanical interactions between the meat piece and its surroundings, a point cloud renderer was also introduced. This step is needed to generate input data for the pick and place strategy, which determine the robot motion based on a segmented point cloud of the meat. To ensure the meat piece is segmented, a new scene only containing the meat piece is generated and then the point cloud is captured in this scene. The RobWork [37] point cloud renderer was used to generate the point cloud, and the final result is illustrated in Fig. 15.

6.6. The Simulation Scenario

The goal of the simulator is to model the use case addressed in this paper. The scenario we address is that meat pieces are dropped into a box. Then a scanner generates a segmented point cloud of the top meat piece, as discussed in Section 4. This point cloud is used to determine a robotic pick and place action for moving the meat, as discussed in Section 5. Then the robotic action is executed, which moves the meat from the box to a conveyor belt.

To simulate this, the meat piece is first placed above the box. The meat is then translated by $(M_x, M_y, 0)$ and rotated by M_R around the z-axis to randomize the initial position. Then it is dropped such that it falls into the box, similarly to if a human dropped it. The constraint in the box is based on a thin plate spline, so after the meat settles, it is placed in a deformed state, similarly to if it was placed on top of another meat piece. Then a point cloud of the meat piece is rendered to generate the robotic pick and place action for moving the meat to the conveyor belt.

After the action is generated, the robotic motion is simulated. First, the suction cups are moved down to the meat. Then they are activated and lifted according to the grasp strategy. Then the suction cups move the meat to the conveyor belt where it is placed. After it is placed the suction cups are lifted, and then the simulation ends. Images from the simulation can be seen in Fig. 9 and 11.

7. Optimization in Simulation

To determine an action that places the meat pieces stretched out on a table, we propose to use simulation-based optimization. The simulation used is discussed in Section 6. The strategy used to place the meat is based on various control parameters, and it is these parameters that should be optimized. The control parameters are discussed in Section 5 and listed in Table 1.

Besides the control parameters, there are also many uncertain parameters in the use case, such as the shape and initial placement of the meat. To ensure the robot solution is robust, it is important to analyze whether the performance of the solution is stable, even when the uncertain parameters are varied. Both the control parameters and the uncertain parameters are discussed in Section 7.1.

In order to optimize the control parameters, it is necessary to quantify the quality of a placement based on the simulations. This score should favor actions that place the meat stretched out on a table. The score for achieving this is discussed in Section 7.2.

Finally, the process for determining a robust solution based on numeric optimization is discussed in Section 7.3. This process is applied to two different cases. In the first the robot has to pick pork bellies and in the second it has to pick pork loins.

7.1. Free Parameters and Uncertainties

Several control parameters that are crucial for generating a stable placement action were selected for optimization. All the parameters are listed in Table 1, the first parameter specifies the gripper design, the next 4 specify the grasp action and the last 7 specify the placement action. Besides just listing the parameters the table also shows the parameter bounds used during optimization of the entire pick and place action. These bounds are selected based on hardware limitations and to ensure that the meat pieces are placed on the conveyor belt. All the control parameters have been described in detail in Section 5.

Besides the control parameters, the system also contains a lot of uncertain parameters, such as meat size and deformability. To ensure the solution can cope with variation in these parameters, the tested actions should be simulated with different perturbations of the uncertain parameters. To achieve this the first step was to determine the most crucial uncertainties. These are listed in Table 2. Furthermore, the bounds the parameters can occur within should be estimated. These bounds are based on data from the production lines at Danish Crown and the values are given in Table 2.

The first 6 uncertain parameters are introduced to capture the variation between the meat pieces and how they are placed in the boxes. M is the weight of the meat piece. S_{def} is a deformability parameter. This parameter model the variation in the deformability of the meat pieces. In reality, this variation occurs due to multiple factors, such as variation in the thickness, fat content and temperature of the meat. In the simulation, the variation is modeled by a scalar multiplied to all the spring constants in the meat model. The base spring constants are chosen to make the simulated meat deform similarly to the real meat pieces, the spring model and the constants are discussed in more detail in [3]. S_{size} is a scaling factor multiplied to the base size of the meat piece, for the pork belly this size is $525 \times 250 \times 20$mm and for the pork loin it is $550 \times 120 \times 75$mm. M_x and M_y are perturbations of the meat piece in the x and y-direction before it is dropped into the box in the simulations. M_R specify how much the meat is rotated around the z-axis before it is dropped.

The following 6 parameters are included to model imperfections in the camera-robot calibration and other

Table 1: Control parameter bounds used during optimization of the placement action.

	Gripper	**Rolling Grasp**				**Placement**			
	d	$\mathbf{d_{ideal}}$	**w**	**gl**	**gr**	$\mathbf{d}_{x1}$, $\mathbf{d}_{y1}$, $\mathbf{d}_{z1}$	θ	**b**	$\mathbf{d}_{x2}$, $\mathbf{d}_{y2}$
min	130mm	20mm	0	100mm	15°	0mm	0°	0	0mm
max	170mm	50mm	1	200mm	25°	100mm	20°	2	100mm

Table 2: Parameter bounds for the uncertain values of the pork belly, which is used to analyze the robustness of the solutions.

	Meat Cutout					**Vision**		**Contact Surface**	
	M	$\mathbf{S_{def}}$	$\mathbf{S_{size}}$	$\mathbf{M_{x,\,y}}$	$\mathbf{M}_R$	$\mathbf{V_{x,\,y,\,z}}$	$\mathbf{V_{R,\,P,\,Y}}$	$\mathbf{S_{offset}}$	$\mathbf{d_{tps}}$x28
min	3.5kg	0.9	0.95	-50mm	-10°	-10mm	-3°	0mm	0.0
max	5.5kg	1.1	1.05	50mm	10°	10mm	3°	200mm	1.0

Table 3: Parameter bounds for the uncertain values of the pork loin, which is used to analyze the robustness of the solutions.

	Meat Cutout					**Vision**		**Contact Surface**	
	M	$\mathbf{S_{def}}$	$\mathbf{S_{size}}$	$\mathbf{M_{x,\,y}}$	$\mathbf{M}_R$	$\mathbf{V_{x,\,y,\,z}}$	$\mathbf{V_{R,\,P,\,Y}}$	$\mathbf{S_{offset}}$	$\mathbf{d_{tps}}$x28
min	2.0kg	0.9	0.9	-50mm	-10°	-10mm	-3°	0mm	0.0
max	3.5kg	1.1	1.1	50mm	10°	10mm	3°	200mm	1.0

uncertainties introduced by the vision system. V_x, V_y and V_z specify perturbations in the x, y and z directions of the point cloud of the meat after it is dropped into the box. V_R, V_P and V_Y specify a roll, pitch and yaw perturbation to the rotation of the point cloud.

The last uncertain parameters are included to model the variation of the box the meat is dropped into. This variation occurs because there can be between 0 and 8 meat pieces below the top piece that is to be grasped. This surface is uneven and in the simulation, it is modeled by a thin plate spline. This spline is specified based on 29 uncertain parameters. First S_{offset} specify the maximum height of any knot point in the thin plate spline. The other 28 d_{tps} parameters are used to specify the height of the individual 28 knot points, while ensuring the points are never placed below the box.

7.2. The Objective Score

To use numeric optimization, an objective score has to be defined. This score should capture the quality of any given set of control parameters, such that the optimal pick and place action can be distinguished from poor actions. In this work, the objective score is determined based on an automated analysis of the simulations. In particular, it is determined by analyzing each meat particle throughout the simulation, which is discussed in Section 6

To capture the quality of a solution the score should address three different issues. First, it should favor actions resulting in the meat being stretched out on the table. Secondly, it should favor actions where the orientation of the meat matches the desired orientation. Lastly, it should favor actions where the internal forces in the meat are limited, to ensure the meat is not damaged in the operation. These issues are addressed by constructing the final objective score from three different scores.

The first two scores ensure that the rotation and deformation of the meat piece match the desired rotation and deformation after it is placed on the conveyor belt. To determine these scores, the first step is to determine the pose of the meat piece after it is moved to the conveyor belt. This is done by using the Kabsch algorithm [38] between the point set representing the desired meat placement and the point set representing the meat piece in the simulation. This returns a pose transformation

from the desired point set to the actual point set in the simulation. The rotation from the pose transformation is then used as the rotational error, E_{rotation}. This error is converted to the **rotation objective** through (9).

$$Q_{\text{rotation}} = \begin{cases} 0 & \text{if } E_{\text{rotation}} > 30^\circ \\ 1 - \frac{E_{\text{rotation}}}{30^\circ} & \text{otherwise} \end{cases} \tag{9}$$

To determine the **deformation objective**, the desired point set is moved onto the final point set using the pose transformation, and then the RMS error between the two point sets are determined. This score is used as the deformation error, $E_{\text{deformation}}$, which is converted into an objective score through (10).

$$Q_{\text{deformation}} = \begin{cases} 0 & \text{if } E_{\text{deformation}} > 50\text{mm} \\ 1 - \frac{E_{\text{deformation}}}{50\text{mm}} & \text{otherwise} \end{cases} \tag{10}$$

The last objective score is the **force objective**. This score favors solutions that produce small internal forces inside the meat pieces. This score is based on the maximal force exerted on any meat particle throughout the simulation. The maximal force, F_{max}, is converted into an objective score through (11).

$$Q_{\text{force}} = \frac{6.0\text{N}}{F_{\text{max}}} \tag{11}$$

Finally, all the objective scores are combined into one score, Q, using the geometric mean as shown in (12). The geometric mean was chosen since it favors solutions where all the objective scores are high. Furthermore, the partial objective scores are all designed to be between 0 and 1, and thus the combined score will also be in this interval. For more detail, on the objective scores, we refer to [3].

$$Q = \sqrt[3]{Q_{\text{rotation}} \cdot Q_{\text{deformation}} \cdot Q_{\text{force}}} \tag{12}$$

7.3. Numeric Optimization

In bounded global numeric optimization, the idea is to determine the parameter set resulting in the highest function evaluation for a multi-dimensional function. This can be expressed by equation (13).

$$\mathbf{x}_{opt} = \underset{\mathbf{x} \in \mathbb{R}^n | \mathbf{x}_{min} \leq \mathbf{x} \leq \mathbf{x}_{max}}{\operatorname{argmax}} \quad f(\mathbf{x}) \tag{13}$$

In this work, $\mathbf{x}$ is the control parameters that define the pick and place action and $\mathbf{x}_{opt}$ define the best action. $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ are the bounds which the control parameters should be optimized within. f is based on the objective score, Q, which is calculated in the simulations. To ensure f favors solutions that are robust to the uncertain parameters, it is determined based on multiple simulations with different uncertain parameter perturbations according to (14).

$$f(\mathbf{x}) = \bar{Q} - 2 \cdot \text{SD}(Q) \tag{14}$$

where $\bar{Q}$ is the average objective score based on multiple simulations. $\text{SD}(Q)$ is the standard deviation of the objective scores.

When computing the score, the variation in Q is achieved by varying the uncertain parameters of the simulation uniformly within the uncertainty bounds. Equation (14) is based on work presented in [4]. In [4], the equation is demonstrated to be effective at determining solutions that work well, even when tested for different perturbations of the uncertain parameters in simulation.

Several tools exist for solving the maximization or optimization problem. In previous work, [26], we showed that *RBFopt* is a powerful optimization algorithm for robotic meat handling and other robotic use-cases where the solutions should be robust to various uncertainties in the system. Thus in this work, we use *RBFopt* to optimize the parameters.

During the optimization, it is infeasible to run a substantial amount of simulations for each parameter set. Therefore, as verified in [26], we propose to do multiple optimization runs where each parameter set is evaluated based on a few simulations. Then for each optimization run, the best parameter set are thoroughly evaluated to determine the very best set. During the optimization, we evaluate the parameter set in simulation based on 10 different perturbations of the uncertain parameters. Furthermore, we do 10 optimization runs with 100 iterations each. After the 10 best parameter sets are determined we evaluate them based on 1000 different perturbations of the uncertain parameters to determine the best parameter set, which is then used as the final solution. This optimization process is done for both use cases, to determine case-specific solutions for both cases.

For the pork belly case, the optimization process is illustrated in Fig. 16. In Fig. 16a, f is plotted throughout the iterations of the optimization runs. Here it can be seen that the objective score increases substantially throughout the optimization. This shows that the pick and place action improve substantially as better and better control parameters are tested. In Fig. 16b, the final solutions of the individual optimization runs are compared, in order to select the very best solution. This solution is at index 6, where f evaluated based on 1000

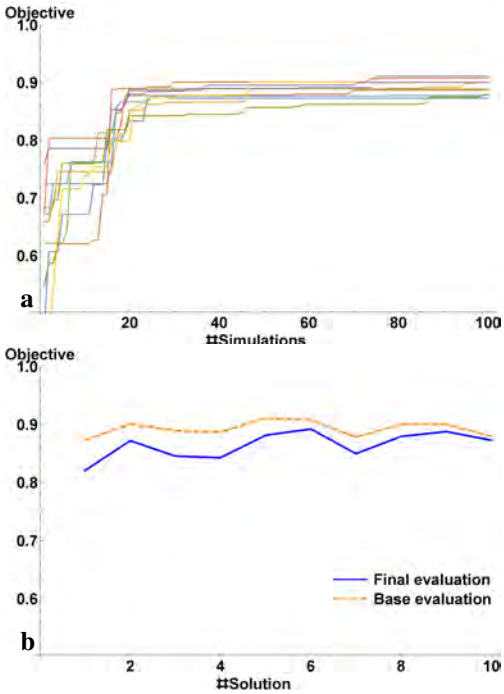


Fig. 16: Optimization of the pork belly pick and place action. a) Each graph shows the best score achieved as the optimization algorithm progress over the 100 iterations. The 10 different graphs represent the 10 optimization runs. b) The 10 resulting optimal solutions are evaluated 1000 times to determine the very best with a more extensive evaluation. The dashed orange line shows the scores of the optimized parameters based on 10 evaluations and the blue line shows the scores of the same parameters based on 1000 evaluations.

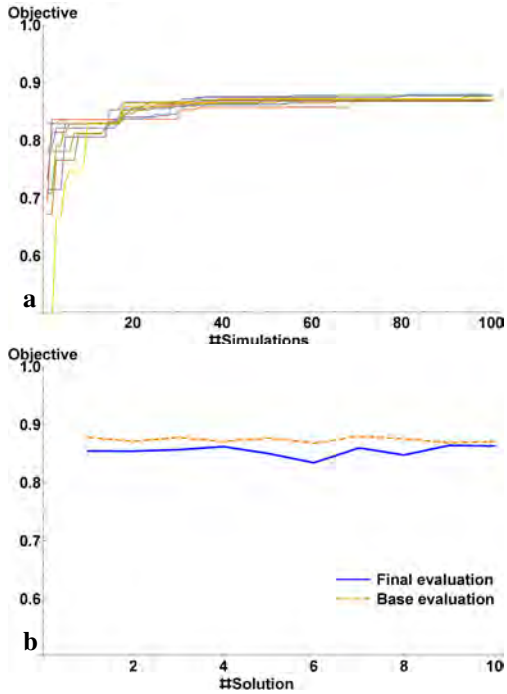


Fig. 17: Optimization of the pork loin pick and place action. The graphs are similar to Fig. 16.

simulations result in 0.892. The two graphs in Fig. 16b represents solution qualities based on 10 and 1000 evaluations. Due to the similarity between the two graphs, it can be seen that the objective scores based on 10 and 1000 evaluations are correlated. However, picking a solution based on 1000 evaluations changes the best pick from solution 5 to solution 6, so the final evaluation improves the choice slightly.

The optimization process for the pork loin case is illustrated in Fig. 17. This case appears easier since the objective scores during optimization converge more quickly. The best solution is at index 9, where the objective score evaluated based on 1000 simulations result in 0.864. Furthermore, the performance of the optimized parameter sets is more similar compared to the pork belly case. Again the best solution changes from solution 7 to 9 when 1000 simulations are used, so again the final evaluation improves the choice slightly.

The optimal parameter sets for the two cases are shown in Table 4.

8. Real World Evaluation

In this section, the real world evaluation of the robot solutions proposed in this paper is discussed. First, we discuss how the grasp strategy was fine-tuned and evaluated on a physical prototype at a Danish slaughterhouse. Then we discuss the evaluation of the placement strategy, which was optimized in simulation. The optimized solutions are evaluated for pork bellies and pork loins.

8.1. Grasping Different Pork Cuts

The first part of the experiments was done to determine a reliable grasp strategy for the physical prototype. This was difficult to optimize in simulation due to many subtle effects playing a role in the success of each grasp. To capture all these effects in simulation would be computationally intractable.

Table 4: Control parameters used during real world trials. Pork belly and Pork loin refer to the optimal parameter sets for the two cases.

	Gripper	**Rolling Grasp**				**Placement**						
	d	$\mathbf{d}_{ideal}$	**w**	**gl**	**gr**	$\mathbf{d}_{x1}$	$\mathbf{d}_{y1}$	$\mathbf{d}_{z1}$	θ	**b**	$\mathbf{d}_{x2}$	$\mathbf{d}_{y2}$
Pork belly	170mm	22.7mm	0.78	199mm	16.5°	9.6mm	0.3mm	65.9mm	0.2°	1.33	14.4mm	0.3mm
Pork loin	170mm	42.7mm	0.99	102mm	15.3°	8.5mm	0.7mm	100.0mm	0.0°	1.28	15.6mm	1.8mm
Default	130mm	0mm	0.5	150mm	20°	0mm	0mm	0mm	0°	0	0mm	0mm

Fig. 18: The pork cuts tested during grasp evaluation. a) pork loin, b) pork back, c) single ribbed narrow belly, d) undercut narrow belly, e) single ribbed heavy belly, f) undercut heavy belly, g) single ribbed narrow belly with skin.

During the real world trials of the grasp strategy, seven different cuts of pork were grasped. These are all shown in Fig. 18, the backs and loins are thicker than the bellies and therefore more rigid. The bellies are wider and thinner and therefore tend to be quite flexible. The heavy bellies are overall larger than the narrow bellies. The undercut bellies tend to be the least rigid since some of the meat structure on the meat side is removed.

During the grasps, several types of failures occurred, to better analyze the solutions we have split the grasp results into 4 categories. These categories are 3 different failure types and success, S. All the categories are illustrated in Fig. 19. The first failure type is failure before the lift, FBL. This refers to failures where the gripper tool is unable to establish vacuum before lifting the meat. The second failure type is failure after the lift, FAL. This refers to the suction cups losing vacuum after the meat is lifted and separated from the piece below. The last failure type is failure due to multiple lifts, FML. This refers to failures where two meat pieces or a meat piece and the box stick together. This can cause the gripper to lift both objects, which is undesirable.

After some initial trials and fine tuning of the rolling grasp strategy, we evaluated it on all 7 cuts of pork. The success rate and the failure causes of the grasps are shown in Fig. 20. For most cuts between 40 and 50 trials were done, but for pork bellies with skin (Fig. 18g) we only did 15 since this was clearly easier than all other cases.

The results show that the undercut pork bellies are significantly more difficult to lift than the single ribbed bellies. This is because they contain less structure and thus are more flexible. During the lift, this extra flexibility makes it more likely that the meat deforms at the

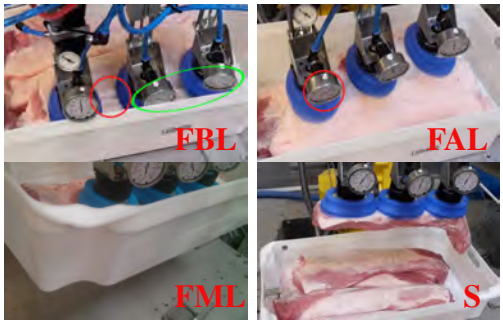


Fig. 19: Failure and success categories. FBL) the green circle highlights the vacuum gauges which show vacuum is never established. The reason for this is the large bulges highlighted by the red circle. FAL) the lift starts well, but vacuum is lost at the vacuum gauge highlighted by the red circle. FML) the box is lifted with the meat. S) a successful lift of the pork loin.

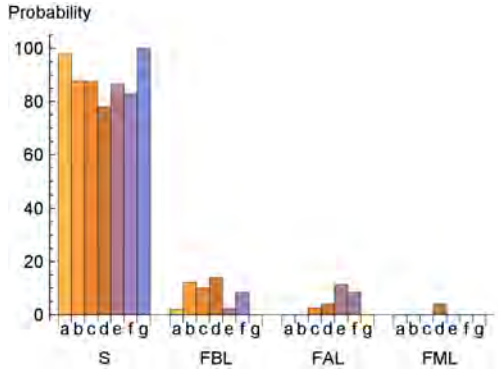


Fig. 20: Success and failure rates of the grasp strategy. a, b, c, d, e, f and g refers to the cuts listed in Fig. 18. S is successful grasp, FBL is failure before lift, FAL is failure after lift and FML is failure due to lifting multiple pieces.

Fig. 21: Example of the image data collected for our analysis.

suction cups and allow air to flow in.

It can also be seen that it is only undercut bellies that fail due to the gripper lifting multiple objects. This is again due to the lack of meat structure which makes it more likely that a vacuum is formed between two meat pieces such that they stick together. Besides the single ribbed pork bellies the system also handles pork backs well, and as seen from the failure types pork backs are never dropped during the lift. This is because this cut is thicker and more rigid, thus the meat is less prone to deform and allow air to flow into the suction cups. However, pork backs are also more narrow and thus it is more likely that the suction cups are slightly misplaced before the grasp. The system also handles pork loins well and since they deform even less than the pork backs, the suction cups almost always create and maintain a stable vacuum.

After the grasp strategy was tested, the next step was to optimize the entire pick and place action in simulation. Since pork bellies are the most common cut at the "Danish Crown" slaughterhouse we decided one of the test cases should be a single ribbed heavy belly (Fig. 18e). The reason for picking this particular belly cut is that the vacuum gripper is more likely to work on single ribbed cuts. Furthermore, since it is wider it is more difficult to control during the placement which makes it more interesting from a scientific perspective.

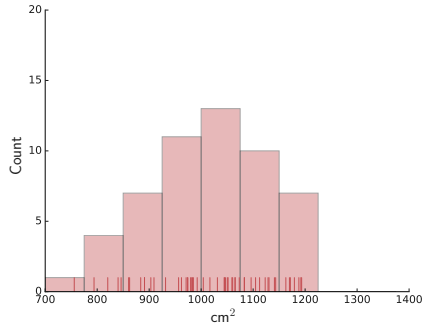
Besides the pork belly we also picked the pork loins as a test case, the reason for this is that the loins are the cut that differs most from the bellies. Thus this is the best cut for illustrating the versatility of the system.

8.2. Placement Quality

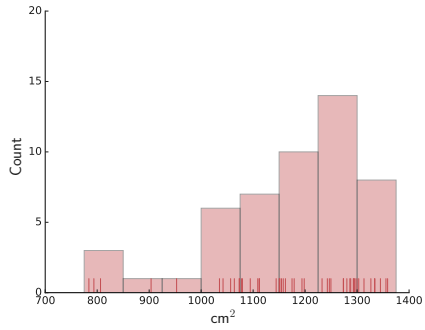
After the rolling grasp strategy was fine-tuned and tested in real world trials (see Section 8.1) the next step was to optimize the parameters relevant for the placement in simulation. This was done as discussed in Section 7. After the optimal parameter set was found, the next step was to evaluate it in the real world and determine whether it leads to better performance. We evaluated the default parameter set and the optimized pork belly parameter set from Table 4 for picking and placing pork bellies in the real world.

We evaluate the quality of each strategy via running a series of trail grasps. We allow the robot to pick the meat and place it as intended on a table. Then we acquire an image of the meat which we can use to quantify the results. The idea is that the ideal pose should be the meat lying completely flat without any folds on the delivery table. This means that the meat's visible surface will be maximised. So by taking an image of the meat in it's delivered pose and quantifying it's surface area, we can quantity the quality of delivery.

After the robot has transported the meat, we take a photo. We also ensure that a calibration artifact



(a) Unoptimized



(b) Optimized

Fig. 22: Distribution of meat area after delivery for the two strategies.

(checkerboard) is present near the meat. By using the artifact we can deduce the meat's physical size from its image space size as well as its physical location. Fig. 21 shows an example of one such photography.

We have this experiment for pork belly cutouts and fig. 22 shows the resulting distribution of the meat's visible surface area after delivery for two strategies: unoptimized and optimized via the previosly described simulation framework. The optimized strategy shows a consistenly higher mean of surface area compared to the unoptimized strategy. The mean being 151cm^2 higher. We can conclude that optimized strategy is better at maximising the visible area and thus at placing the meat in an optimal flat pose. Qualitative inspection supports this conclusion as the one side of the pork belly is consistently folded for the unoptimized strategy. Fig 23 shows an example of this along with a succesful example from the optimized strategy. As such we can see a clear improvement in quality by employing parameters obtained

(a) Unoptimized (b) Optimized

Fig. 23: Examples of pork belly pose after delivery. Unoptimized grasping consistently folds the meat, whereas the optimized strategy delivers a consistently flat pose.

Fig. 24: Top shows a bad placement. Bottom shows a satisfactory pose.

from the simulation-based optimization.

As shown by the results in Fig. 20, the system is fairly generic and capable of grasping many variations in cutout shape and size. To illustrate that the placement strategy is also generic, we tested it for the pork loins as well. This was again done by optimizing the placement strategy in simulation and then testing the solution in real world trials. The optimized parameter set is listed as Pork loin in Table 4. Using this parameter set, we achieved a satisfactory placement for approximately 98% of the pick and place operations. A failure and success case is shown in Fig. 24.

9. Conclusion and Future Work

In this work, we have presented a generic solution for doing pick and place operations with meat pieces. Furthermore, we have presented a simulation-based optimization procedure for tuning the generic solution to specific use cases. Finally, the resulting solutions have been evaluated in the real world to validate the approach.

To enable the robot action, the first step was to design a vision system for detecting the meat. The vision system developed for this work is able to generate precise point clouds of the 7 pork cuts tested. Furthermore, a

segmentation algorithm was designed, which is able to segment the top surface of all 7 pork cuts.

To move the meat a robot and a suction based gripper tool was used. The robot motion for moving the meat is based on the segmented point cloud from the vision system. Furthermore, it is based on a rolling lift which allows air to flow below the meat piece to avoid it sticking to the surface below. The placement strategy is designed as a simple draping like motion to place the meat piece stretched out on a table.

The entire pick and place action was optimized in simulation to determine the most robust action for placing the meat flat on a table. The resulting solution was tested in the real world. This solution was compared to a non-optimized solution and it is shown that the optimized solution improves the performance by stretching the meat more in the real world as well.

To show the generality of the entire approach, we also optimized it for moving pork loins. For this case, we achieved a success rate of 98% for placing the pork loins nicely on a table.

In future work, we intend to extend the optimization framework to model more grippers and manipulation tools. This would enable the framework to optimize solutions for a much broader range of problems within the food sector. Furthermore, if new gripper tools are able to handle sacks or cloth, it would be possible to evaluate the system in substantially different domains and show the broad applicability of the overall approach.

The vision solution used in this work is already fairly generic. However, for it to work optimally it requires static background lighting, which cannot always be garuanteed. A possible solution to this problem would be the light concentration technique of [39], which vastly increases the SNR of the projected pattern thus making the noise from the background illumination irrelevant. Another feasible solution would be to increase acquisition speed via better hardware synchronization. It should be possible to each speeds of 10-20 point clouds pr. second [40]. Such speed would make most background lighting appear approximately constant.

Even though there are some limitations to the presented system, the system and the individual technologies can still help speed up the design and integration of automation systems for handling meat pieces. This is especially beneficial when automating small batch production, where the design and integration cost is a relatively large part of the overall production cost.

Acknowledgement

The financial support from the The Danish Innovation Foundation through the strategic platform "MADE-Platform for Future Production" and from the EU project Xperience (FP7-ICT-270273) is gratefully acknowledged.

References

[1] S. Jensen, J. Wilm, H. Aanæs, An Error Analysis of Structured Light Scanning of Biological Tissue, Springer, 2017, pp. 135–145. doi:10.1007/978-3-319-59126-1_12.

[2] T. B. Jørgensen, M. M. Pedersen, N. W. Hansen, B. R. Hansen, N. Kruger, A flexible suction based grasp tool and associated grasp strategies for handling meat, International Conference on Mechatronics and Robotics Engineering (accepted 2017).

[3] T. B. Jørgensen, P. H. S. Holm, H. G. Petersen, N. Krüger, Intelligent Robotics and Applications: 8th International Conference, ICIRA 2015, Portsmouth, UK, August 24-27, 2015, Proceedings, Part II, Springer International Publishing, Cham, 2015, pp. 431–444.

[4] T. B. Jørgensen, K. Debrabant, N. Krüger, Robust optimization of robotic pick and place operations for deformable objects through simulation, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 3863–3870.

[5] D. Kruse, R. J. Radke, J. T. Wen, Human-robot collaborative handling of highly deformable materials, in: American Control Conference (ACC), 2017, IEEE, 2017, pp. 1511–1516.

[6] L. Bodenhagen, A. R. Fugl, A. Jordt, M. Willatzen, K. A. Andersen, M. M. Olsen, R. Koch, H. G. Petersen, N. Krüger, An adaptable robot vision system performing manipulation actions with flexible objects, IEEE transactions on automation science and engineering 11 (2014) 749–765.

[7] J. Schulman, A. Lee, J. Ho, P. Abbeel, Tracking deformable objects with point clouds, in: Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, 2013, pp. 1130–1137.

[8] Y. Li, Y. Wang, M. Case, S.-F. Chang, P. K. Allen, Real-time pose estimation of deformable objects using a volumetric approach, in: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, IEEE, 2014, pp. 1046–1052.

[9] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, L. Van Gool, The 2017 davis challenge on video object segmentation, arXiv:1704.00675 (2017).

[10] P. Voigtlaender, B. Leibe, Online adaptation of convolutional neural networks for video object segmentation, in: BMVC, 2017.

[11] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, L. Van Gool, One-shot video object segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2017.

[12] Y. J. Koh, C.-S. Kim, Primary object segmentation in videos based on region augmentation and reduction, 2017. URL: http://openaccess.thecvf.com/content_cvpr_2017/papers/Koh_Primary_Object_Segmentation_CVPR_2017_paper.pdf.

[13] E. R. Eiríksson, J. Wilm, D. B. Pedersen, H. Aanæs, Precision and accuracy parameters in structured light 3-d scanning., International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 40 (2015).

[14] P. Long, W. Khalil, P. Martinet, Robotic deformable object cutting: From simulation to experimental validation, 2014.

[15] E. Nabil, B. Belhassen-Chedli, G. Grigore, Soft material modeling for robotic task formulation and control in the muscle separation process, Robotics and Computer-Integrated Manufacturing 32 (2015) 37–53.

[16] E. Misimi, E. R. Øye, A. Eilertsen, J. R. Mathiassen, O. B. Åsebø, T. Gjerstad, J. Buljo, Ø. Skotheim, Gribbot–robotic 3d vision-guided harvesting of chicken fillets, Computers and Electronics in Agriculture 121 (2016) 84–100.

[17] B. Balaguer, S. Carpin, Combining imitation and reinforcement learning to fold deformable planar objects., in: IROS, IEEE, 2011, pp. 1405–1412. URL: http://dblp.uni-trier.de/db/conf/iros/iros2011.html#BalaguerC11.

[18] G. T. Zoumponos, N. A. Aspragathos, A fuzzy strategy for the robotic folding of fabrics with machine vision feedback, Industrial Robot: An International Journal 37 (2010) 302–308.

[19] F. Berkenkamp, A. Krause, A. P. Schoellig, Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics, arXiv preprint arXiv:1602.04450 (2016).

[20] T. Hemker, M. Stelzer, O. von Stryk, H. Sakamoto, Efficient walking speed optimization of a humanoid robot, The International Journal of Robotics Research 28 (2009) 303–314.

[21] R. Calandra, A. Seyfarth, J. Peters, M. P. Deisenroth, Bayesian optimization for learning gaits under uncertainty, Annals of Mathematics and Artificial Intelligence 76 (2016) 5–23.

[22] M. Tesch, J. Schneider, H. Choset, Using response surfaces and expected improvement to optimize snake robot gait parameters, in: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, IEEE, 2011, pp. 1069–1074.

[23] J. P. Buch, J. S. Laursen, L. C. Sørensen, L.-P. Ellekilde, D. Kraft, U. P. Schultz, H. G. Petersen, Applying simulation and a domain-specific language for an adaptive action library, in: International Conference on Simulation, Modeling, and Programming for Autonomous Robots, Springer, 2014, pp. 86–97.

[24] A. Wolniakowski, J. A. Jorgensen, K. Miatliuk, H. G. Petersen, N. Kruger, Task and context sensitive optimization of gripper design using dynamic grasp simulation, in: Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on, IEEE, 2015, pp. 29–34.

[25] A. Costa, G. Nannicini, Rbfopt: an open-source library for black-box optimization with costly function evaluations, Optimization Online (2014).

[26] T. B. Jørgensen, A. Wolniakowski, H. G. Petersen, K. Debrabant, N. Kruger, Robust optimization with applications to design of context specific robot solutions, Robotics and Computer Integrated Manufacturing (Submitted 2017).

[27] H. Gonzalez-Jorge, P. Rodríguez-González, J. Martínez-Sánchez, D. González-Aguilera, P. Arias, M. Gesto, L. Díaz-Vilariño, Metrological comparison between kinect i and kinect ii sensors, Measurement 70 (2015) 21–26.

[28] J. Geng, Structured-light 3d surface imaging: a tutorial, Advances in Optics and Photonics 3 (2011) 128–160.

[29] M. F. Tappen, W. T. Freeman, Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters, in: null, IEEE, 2003, p. 900.

[30] J. Posdamer, M. Altschuler, Surface measurement by space-encoded projected beam systems, Computer Graphics and Image Processing 18 (1982) 1–17.

[31] M. Gupta, S. K. Nayar, Micro Phase Shifting, Proc. IEEE CVPR (2012) 813–820.

[32] R. B. Rusu, S. Cousins, Region growing segmentation, 2017. URL: http://pointclouds.org/documentation/tutorials/region_growing_segmentation.php.

[33] R. B. Rusu, S. Cousins, 3D is here: Point Cloud Library (PCL), in: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011.

[34] I. Loshchilov, M. Schoenauer, M. Sebag, Adaptive coordinate descent, 2011.

[35] J. Mesit, R. Guha, S. Chaudhry, 3d soft body simulation using mass-spring system with internal pressure force and simplified implicit integration, Journal of Computers 2 (2007) 34–43.

[36] D. Eberly, Thin plate splines, Geometric Tools Inc 2002 (2002) 116.

[37] L.-P. Ellekilde, J. A. Jorgensen, Robwork: A flexible toolbox for robotics research and education, in: Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK), VDE, 2010, pp. 1–7.

[38] W. Kabsch, A solution for the best rotation to relate two sets of vectors, Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography 32 (1976) 922–923.

[39] M. Gupta, Q. Yin, S. K. Nayar, Structured light in sunlight, in: The IEEE International Conference on Computer Vision (ICCV), 2013.

[40] J. Wilm, O. V. Olesen, R. Larsen, Slstudio: Open-source framework for real-time structured light, Proceedings of the 4th International Conference on Image Processing Theory, Tools and Application (ipta 2014) (2014) 7002001.

APPENDIX C

An Error Analysis of Structured Light Scanning of Biological Tissue

An Error Analysis of Structured Light Scanning of Biological Tissue

Sebastian Nesgaard Jensen, Jakob Wilm and Henrik Aanæs
{snje, jakw, aanes}@dtu.dk

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads B321, Kongens Lyngby, Denmark

Abstract. This paper presents an error analysis and correction model for four structured light methods applied to three common types of biological tissue; skin, fat and muscle.
Despite its many advantages, structured light is based on the assumption of direct reflection at the object surface only. This assumption is violated by most biological material e.g. human skin, which exhibits subsurface light reflection. In this study, we find that in general, structured light scans of biological tissue deviate significantly from the ground truth. We show that a large portion of this error can be predicted with a simple, stochastic linear model based on the scan geometry. As such, scans can be corrected without introducing any specially designed pattern strategy or hardware. We can effectively reduce the error in a structured light scanner applied to biological tissue by as much as factor of two or three.

Keywords: 3D Reconstruction, Error Modeling, Structured Light

1 Introduction

Structured light has proven to be very useful for 3D scene acquisition. This is due to its high speed, precision and versatility. As such a wide array of related techniques have been developed in the past decades, facilitating everything from high precision metrology to real-time guidance of automation [8].

Structured light uses a calibrated camera-projector pair as shown in Fig. 1. A series of time multiplexed patterns is projected onto the scene, which can be used for matching and triangulation with the camera. This active approach makes correspondence searching much simpler than passive stereo approaches, and is applicable to scenes with poor texturing. A very important application for structured light is 3D scanning of biological materials, especially human tissue. Examples include head tracking for medical motion correction [22], vision guided surgery [18][23], medical diagnostics [4][1][28] and automation in agriculture and farming [21][25][7]. While the progress in the field has been impressive, one must understand that many target materials are quite problematic. Indeed, they violate the inherent assumption of direct, diffuse surface reflection that most structured light methods are built on. The Fresnel equations predict that when light transitions from one media to another a portion is directly reflected and another is transmitted into the media itself. In the media the light is scattered one or multiple times

until it is absorbed or retransmitted into the environment. The proportion between reflected and refracted light is determined by the specific media's optical properties. For example only 5-7% of human skin reflectance is direct, the remainder is emitted via subsurface scattering [14]. It is therefore of paramount importance that the effect of this violation on structured light is understood and quantified.

In this study, we show that in general, a structured light scan of biological tissue deviates significantly from reference measurements, even with patterns designed specifically to reduce these effects. A large portion of the error can be predicted with a simple, stochastic linear model based on the incident ray geometry. Scans can then be corrected without the need for advanced pattern strategies or special hardware. We can effectively reduce the error in any structured light scanner applied to biological tissue by as much as factor of two or three.

Our study focuses on three types of biological tissue (fat, muscle and skin) with an emphasis on human applications. However we are using porcine materials as a substitute due to its availability and optical similarity to human tissue [26][27]. Through empirical study we quantify the error induced in structured light by the biological material's optical properties. This results in a linear error model based on the view geometry fitted to each method, material combination that can be used to predict and correct for the deterministic scan error.

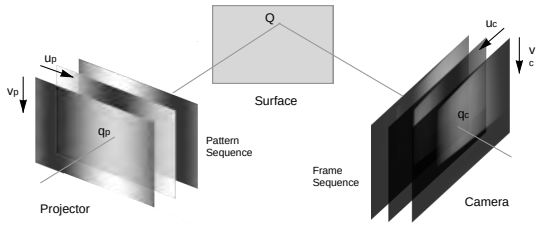


Fig. 1. The structured light principle: a number of patterns are projected onto the scene, and images are captured by a camera. Correspondences are determined by different encoding algorithms, and used to triangulate points on the object surface. In this example, 3-step phase shifting patterns are shown.

2 Related Work

The issue of global lighting effects in the context of structured light has been recognized by many authors, e.g. in the acquisition of a human face reflectance field [6].

In order to reduce these effects, hardware modifications such as polarization have been used [2]. Recent attempts have been to design structured light encoding patterns such that they are less susceptible to global lighting effects. The underlying observation is, that with high-frequent patterns, global lighting effects can be considered constant,

and invariant to a spatial shift of the pattern. This allows for efficient separation of the observed light intensities into direct and global light [20]. In modulated phase shifting [3], structured light patterns are modulated by means of carrier patterns, such that they become high-frequent in both spatial dimensions, thereby improving their separation power. Micro Phase Shifting [10] makes use of sinusoidal patterns in a narrow high-frequency band, promising robustness to global lighting effects and stable phase unwrapping with an optimal number of patterns. It should be noted, that the decoding process in conventional phase shifting methods (e.g. [13]) also implicitly performs direct/global light separation. This is true in particular for high frequency scene coding patterns. Since lower frequency phase unwrapping patterns are affected differently by global lighting effects, this can lead to gross outliers. Hence, the advantage of micro phase shifting is not in higher accuracy, but rather in improved robustness (fewer outliers), and more efficient use of information in the encoding patterns.

A newer approach is unstructured light [5], in which the pattern frequency can be high in both dimensions. However the number of patterns is not ideal, and the matching procedure rather inefficient. For binary encoding methods, exclusively high or low-frequency pattern schemes can be considered robust against different global illumination effects [9].

An approach to compensate for the measurement error in isotropic semi-transparent material caused by subsurface scattering was presented in [16]. Similarly to our approach, this work empirically determines the measurement error and explains it by means of a single variable (the projected light angle), albeit only with a single verification object and structured light method. In [15], a Monte-Carlo simulation of the measurement situation was presented, which gives some insight into the error forming process.

In [11], an analytical derivation of the measurement error is given for the phase shifting method. This error model predicts the error to decrease with increased spatial frequency of the pattern. The model does not however take into account the loss of amplitude at higher frequency patterns, which increases noise in the measurement data. Furthermore it requires precise knowledge about the scanned material's optical properties (extinction coefficient, phase function and index of refraction), all of which can be difficult to find or estimate.

Computer simulations of structured light scans were performed in [19] to benchmark encoding methods with respect to various parameters, and were found to have similar robustness with respect to subsurface effects.

To our knowledge, no study has thus far quantified the amount of error in scans of biological tissue, or provided a means of correcting for it.

3 Statistical Error Model

Our principle assumption is that the error is composed of a deterministic part, which once determined can be subtracted from future scans, in order to improve the accuracy. Previous work gives some hints as to which parameters to include in a statistical error model [16][11].

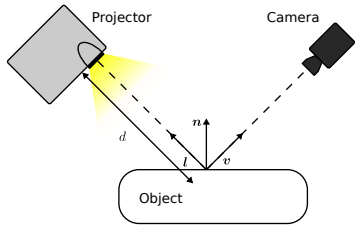


Fig. 2. The structured light scan geometry with the parameters of our error model. The surface normal is n, view direction v, light direction l and the projector-surface distance is d.

Considering the scan geometry, as shown in Fig. 2, we include three variables in our error model: the view angle (given by $n \cdot v$), the light angle (given by $n \cdot l$) and the distance from projector to object, d. We then formulate the following error model:

$$y = \begin{bmatrix} 1 & n \cdot v & n \cdot l & d \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}, \tag{1}$$

where

y is the predicted error in mm,
β_i is a weight,
n, v, l and d are shown in Fig. 2.

We also tried including many other variables, including reflected light to view angle and coding direction to normal vector angles. These variables are inspired by the analytical error model of Holroyd [11], but did not explain sufficient variance to justify their inclusion in our model. We also fitted Holroyd's error model directly, but our linear model provided more explanatory power.

4 Experiments

In order to gather data for the error quantification we scanned surfaces made of one of three porcine tissue types; fat, muscle or skin. All samples were raw and unprocessed with 8 samples of each type. The samples were placed individually in the scan volume and spanned many view and light angles. Their distance to the projector also varied from approximately 200 mm to 400 mm. Each scan produced approximately $5 \cdot 10^5$ data points resulting in millions for each tissue type.

In optical metrology it is common practice to prepare optically challenging surface with a spray [12]. This makes the surface optically diffuse while preserving the geometry. The method was used to acquire a ground truth surface to which each scan was compared. Specifically, after each scan the object was sprayed and covered with a thin

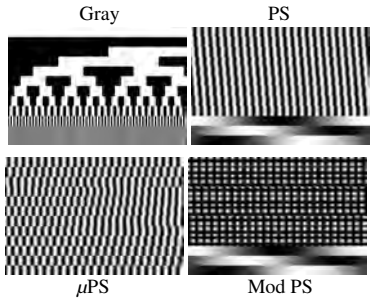


Fig. 3. Structured light patterns used in our experiments. In each case, 12 patterns were used.

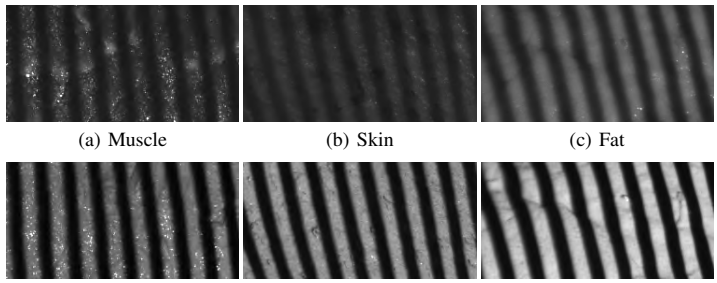


Fig. 4. Fine grained binary structured light pattern projected onto various types of tissues. The effect of subsurface scattering is clearly seen the pattern becomes blurred without chalk coating.

layer of chalk. Then the reference scan was obtained. While we cannot assume that the chalk coated surfaces to be perfect, we consider them ground truth as they provide very clear contrast with virtually no global illumination. In order to verify that this procedure does not alter surface geometry, we applied two separate layers of chalk to a sample object, and compared the scan result after each layer. The mean signed distance was 0.037 mm, indicating that chalk spraying the surfaces does not significantly bias the result. As can be seen in Figure 4 the effect of chalk spraying is relatively pronounced, increasing reflectance and counteracting the pattern blurring caused by subsurface scattering.

In our experiments, we used four different structured light methods:

– Binary Gray coding [24]: one completely lit and one completely dark image were used to define the binary threshold individually in each camera pixel. The remaining patterns were used to encode $2^{10} = 1024$ individual lines on the object surface.

- N-step phase shifting was used with 9 shifts of a high-frequency sinusoid of frequency $1/76\,\mathrm{px}^{-1}$, corresponding to approximately $1/10\,\mathrm{mm}$ on the object surface. Three additional patterns were used for phase-unwrapping [13].
- Micro phase shifting [10] using frequencies in the band $[1/80.00 - -1/70.00]\,\mathrm{px}^{-1}$. These frequencies corresponds to a spatial frequency on the object surface of approximately $1/10\,\mathrm{mm}$. Slightly different from [10], the specific values were determined using a derivative free non-linear pattern search.
- Modulated phase shifting [3] with three shifts of a sinusoid of frequency $1/76\,\mathrm{px}^{-1}$ ($1/10\,\mathrm{mm}$ on the object surface). Each of these sinusoids was modulated in the orthogonal direction using a sinusoidal carrier with the same frequency. Three additional patterns were used for phase-unwrapping.

For the sake of brevity these will henceforth be referred respectively to as; Gray, PS, Micro PS and Mod PS. The former two are standard methods of structured light and can be expected to perform very similar to many derived methods. The latter two are state-of-the-art and have been specifically designed to mitigate the effects of global illumination, as described in Sec. 2. A pattern budget[1] of 12 was settled on for each method as it provided a reasonable balance in acquisition time and accuracy. For all phase-shifting methods, pattern frequency was set so that each period would be approximately 10 mm on the object surface. The remaining frequencies needed in micro phase-shifting were determined using simplex optimization as suggested in the original paper [10]. Fig. 3 shows the pattern sequences used in our experiments.

For every sample, we defined a binary mask within which all possible surface points were reconstructed. This ensured that the exact same surface region was used in the evaluation of each method.

The error of each surface point was quantified by determining its signed distance to the corresponding point in the chalk sprayed reference. For Gray code scans we define the corresponding points as being the pair with the smallest absolute normal distance. With the other methods, we compared points using their position in the pixel grid.

5 Results and Discussion

The parameters obtained after fitting the error model to our data are seen in Table 1, 2 and 3. These shows the estimated parameters as well as the RMS of data compared to the chalk coated reference before and after correction (respectively $\mathrm{RMS}_{\mathrm{raw}}$ and $\mathrm{RMS}_{\mathrm{cor}}$) in units of mm. The two latter were evaluated through a process of leave-one-out k-fold cross validation with 5 partitions. In addition we have also estimated the degree of variance explained, R^2, as well as the P-values for the statistical significance of our model against a constant model. All model dependencies were subject to an analysis of variance (ANOVA) [17].

In general the model provides a significant reduction in RMS for all methods with the greatest effect for muscle and skin. It is interesting to note that R^2 is in general relatively low; at best 13% and at worst 0.8%. Such measure might dispute model's validity, but the statistical test versus a constant model proves otherwise. In all cases we

[1] Pattern budget is the number of projected patterns allowed in a single scan.

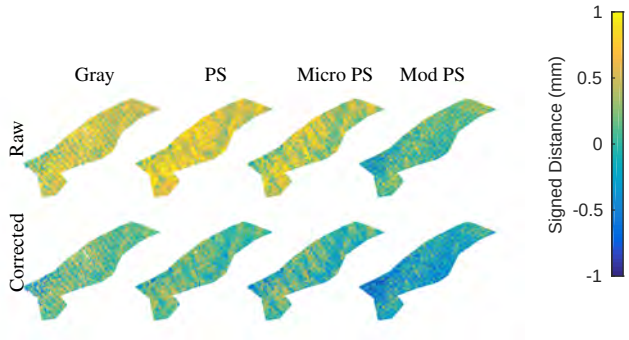


Fig. 5. Signed distance (sd) between scan and reference on a single sample of muscle. Top row: before applying the linear correction model. Bottom row: after correction.

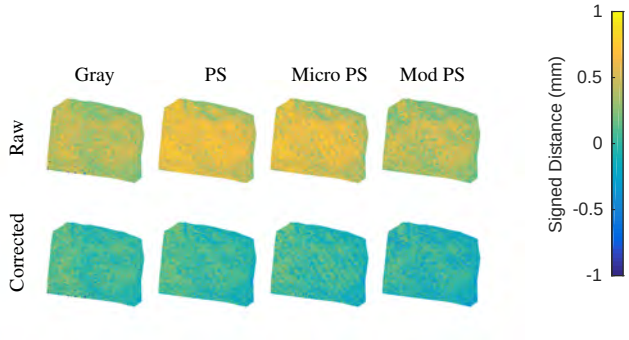


Fig. 6. Signed distance (sd) between scan and reference on a single sample of skin. Top row: before applying the linear correction model. Bottom row: after correction.

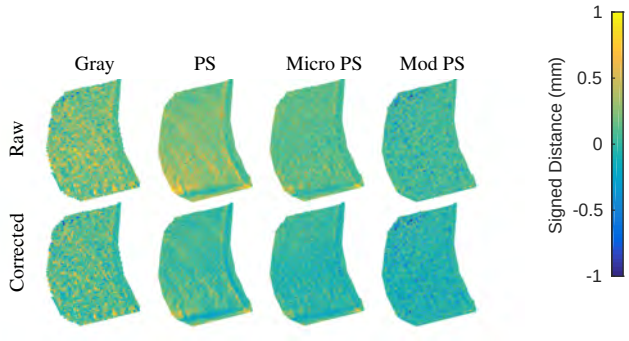


Fig. 7. Signed distance (sd) between scan and reference on a single sample of fat. Top row: before applying the linear correction model. Bottom row: after correction.

Table 1. Muscle model estimate and regression quality.

	β_0	β_1	β_2	β_3	RMS_{raw}	RMS_{cor}	R^2	P
Gray	0.13	0.15	-0.026	2.3×10^{-4}	**0.42**	0.27	0.0082	0
Phase Shifting	0.25	0.47	-0.18	-2.5×10^{-5}	0.5	**0.21**	0.06	0
Micro PS	0.21	0.36	-0.12	-4.1×10^{-6}	0.45	0.23	0.034	0
Modulated PS	0.27	0.077	0.053	-9.7×10^{-5}	0.42	0.26	0.0037	0

Table 2. Skin model estimate and regression quality.

	β_0	β_1	β_2	β_3	RMS_{raw}	RMS_{cor}	R^2	P
Gray	-0.48	0.018	0.43	1.3×10^{-3}	**0.4**	0.19	0.069	0
Phase Shifting	0.27	0.28	0.26	-5.9×10^{-4}	0.54	**0.17**	0.13	0
Micro PS	0.45	0.27	0.21	-1.0×10^{-3}	0.52	0.19	0.13	0
Modulated PS	0.34	0.1	0.27	-6.7×10^{-4}	0.46	0.22	0.054	0

Table 3. Fat model estimate and regression quality.

	β_0	β_1	β_2	β_3	RMS_{raw}	RMS_{cor}	R^2	P
Gray	-0.12	0.13	0.039	2.0×10^{-4}	0.26	0.24	0.016	0
Phase Shifting	-0.18	0.31	-0.11	3.9×10^{-4}	0.22	**0.16**	0.084	0
Micro PS	-0.13	0.2	-0.043	3.0×10^{-4}	**0.2**	0.16	0.043	0
Modulated PS	-0.06	0.15	-0.029	1.6×10^{-4}	0.2	0.17	0.018	0

can conclude that our model is statistical significant within almost a 100% confidence interval, as indicated by the P-values tested against a constant model. While this might seem improbably low, bear in mind that the models was estimated using millions of points which assists in obtaining a statistical significant results. The model estimate itself is rather stable, yielding almost the same error measure for every iteration in the cross validation. This is to be expected due to the high number of training samples and the low dimensionality of the model.

It is seen that most methods have a positive intercept, meaning that regardless of measurement conditions the surface seems to be further away from the camera. The phase shifting methods are especially affected by this bias. This effect is further amplified under ideal scanning conditions, where view and light angle are approximately perpendicular to the measured surface. Since β_1 and β_2 are in most cases positive it will further add to positive surface bias. It is also interesting to note that for phase shifting methods distance adds a negative weight. This means that distant measurement will effectively have less of a positive bias than close ones. The worst bias can be observed in standard phase-shifting applied to skin were error can climb to approximately 0.75 mm.

This positive trend can be illustrated by visualizing the per point error as a heat map upon an obtained point cloud, an interesting trend can be observed. Fig. 5, 6 and 7 shows the signed error on a single sample visually before and after applying the correction model. All have a positive bias which is very strong for muscle and skin. This alludes to a general trend, subsurface scattering causes the estimated surface to lie further away

from the scanner. This is intuitively correct as subsurface scattering is caused by light entering the material for a bit before it is reflected.

In all cases the application of the proper linear model reduces the error's RMS significantly. With a relatively low reduction for fat and a high reduction for skin and muscle. Skin seems to be especially interesting for application as it has the highest error RMS and also receives the largest reduction from error prediction. The remaining unmodeled variance can probably be attributed to variance in chalk thickness, material inhomogeneity and slight vibrations in the recording environment.

6 Conclusion

Structured light is greatly affected by the optical properties of biological materials such as subsurface scattering. By comparing structured light scans of a biological object with scans of the same object covered with a thin chalk layer, we have successfully quantified the resulting error. Our study shows a general positive bias resulting in a surface that lies further away from the scanner than an identical diffuse surface. Due to this positive bias, the RMS of the error can be as high as 0.54 mm. We described the error by fitting a stochastic linear model based on view geometry to the obtained data. Using it, a large portion of the error can be predicted and compensated for. For instance, applying this model to phase-shifting scans of skin reduces error RMS from 0.54 mm to 0.17 mm.

As opposed to the solutions to global illumination proposed in [10][3] our approach requires no specially designed pattern strategy or hardware. It can simply be applied directly to the obtained geometry. Additionally our methodology can be applied to any given structured light method and subsurface scattering material. From a pragmatic view, one must conclude that standard phase-shifting is the superior choice for scanning biological tissue. Not because it shows the lowest error, but rather because the error can be predicted well and compensated using our method.

Bibliography

[1] ACKERMAN, J. D., KELLER, K., AND FUCHS, H. Surface reconstruction of abdominal organs using laparoscopic structured light for augmented reality. *Proc. SPIE 4661* (2002), 39–46.

[2] CHEN, T., LENSCH, H. P. A., FUCHS, C., AND SEIDEL, H. P. Polarization and phase-shifting for 3D scanning of translucent objects. *Proc. IEEE CVPR* (2007).

[3] CHEN, T., SEIDEL, H.-P., AND LENSCH, H. P. Modulated phase-shifting for 3D scanning. *Proc. IEEE CVPR* (2008), 1–8.

[4] CLANCY, N. T., LIN, J., ARYA, S., HANNA, G. B., AND ELSON, D. S. Dual multispectral and 3d structured light laparoscope. *Proc. SPIE 9316* (2015), 93160C.

[5] COUTURE, V., MARTIN, N., AND ROY, S. Unstructured light scanning robust to indirect illumination and depth discontinuities. *Int. Journal on Computer Vision 108*, 3 (2014), 204–221.

[6] DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. Acquiring the reflectance field of a human face. *Proc. SIGGRAPH* (2000), 145–156.

[7] FENG, Q. C., CHENG, W., ZHOU, J. J., AND WANG, X. Design of structured-light vision system for tomato harvesting robot. *Int. Journal of Agricultural and Biological Engineering 7*, 2 (2014), 19–26.

[8] GENG, J. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics 160*, 2 (2011), 128–160.

[9] GUPTA, M., AGRAWAL, A., VEERARAGHAVAN, A., AND NARASIMHAN, S. G. A Practical Approach to 3D Scanning in the Presence of Interreflections, Subsurface Scattering and Defocus. *Int. Journal on Computer Vision 102*, 1-3 (aug 2012), 33–55.

[10] GUPTA, M., AND NAYAR, S. K. Micro Phase Shifting. *Proc. IEEE CVPR* (2012), 813–820.

[11] HOLROYD, M., AND LAWRENCE, J. An Analysis of Using High-Frequency Sinusoidal Illumination to Measure the 3D Shape of Translucent Objects. *Proc. IEEE CVPR* (2011), 2985–2991.

[12] HUANG, Z., NI, J., AND SHIH, A. J. Quantitative evaluation of powder spray effects on stereovision measurements. *Measurement Science and Technology 19*, 2 (2008).

[13] HUNTLEY, J. M., AND SALDNER, H. Temporal phase-unwrapping algorithm for automated interferogram analysis. *Applied Optics 32*, 17 (1993), 3047–3052.

[14] KRISHNASWAMY, A., AND BARANOSKI, G. A biophysically-based spectral model of light interaction with human skin. *Computer Graphics Forum 23*, 3 (2004), 331–340.

[15] LUTZKE, P., HEIST, S., KÜHMSTEDT, P., KOWARSCHIK, R., AND NOTNI, G. Monte Carlo simulation of three-dimensional measurements of translucent objects. *Optical Engineering 54*, 8 (2015).

[16] LUTZKE, P., KÜHMSTEDT, P., AND NOTNI, G. Measuring error compensation on three-dimensional scans of translucent objects. *Optical Engineering 50*, 6 (2011), 063601.

[17] MADSEN, H., AND THYREGOD, P. *Introduction to general and generalized linear models*. CRC Press, Taylor & Francis Group, 2011.

[18] MAURICE, X., ALBITAR, C., DOIGNON, C., AND DE MATHELIN, M. A structured light-based laparoscope with real-time organs' surface reconstruction for minimally invasive surgery. *Proc. Int. Conf. of IEEE EMBS 2012* (2012), 5769–5772.

[19] MEDEIROS, E., DORAISWAMY, H., BERGER, M., AND SILVA, C. T. Using Physically Based Rendering to Benchmark Structured Light Scanners. *Pacific Graphics 33*, 7 (2014).

[20] NAYAR, S. K., KRISHNAN, G., GROSSBERG, M. D., AND RASKAR, R. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. on Graphics 25*, 3 (2006), 935.

[21] NGUYEN, T., SLAUGHTER, D. C., MAX, N., MALOOF, J. N., AND SINHA, N. Structured light-based 3d reconstruction system for plants. *Sensors 15*, 8 (2015), 18587–18612.

[22] OLESEN, O. V., PAULSEN, R. R., HØJGAARD, L., ROED, B., AND LARSEN, R. Motion tracking for medical imaging: a nonvisible structured light tracking approach. *IEEE Trans. Medical Imaging 31*, 1 (2012), 79–87.

[23] PAQUIT, V., PRICE, J. R., SEULIN, R., MERIAUDEAU, F., FARAHI, R. H., TOBIN, K. W., AND FERRELL, T. L. Near-infrared imaging and structured light ranging for automatic catheter insertion. *Proc. SPIE 6141* (2006).

[24] POSDAMER, J., AND ALTSCHULER, M. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing 18* (1982), 1–17.

[25] ROSELL-POLO, J. R., CHEEIN, F. A., GREGORIO, E., ANDJAR, D., PUIGDOMNECH, L., MASIP, J., AND ESCOL, A. Chapter three - advances in structured light sensors applications in precision agriculture and livestock farming. In *Advances in Agronomy*, vol. 133. Academic Press, 2015, pp. 71 – 112.

[26] TFAILI, S., GOBINET, C., JOSSE, G., ANGIBOUST, J.-F., MANFAIT, M., AND PIOT, O. Confocal raman microspectroscopy for skin characterization: a comparative study between human skin and pig skin. *Analyst 137*, 16 (2012), 3673–3682.

[27] WEIGMANN, H.-J., SCHANZER, S., PATZELT, A., BAHABAN, V., DURAT, F., STERRY, W., AND LADEMANN, J. Comparison of human and porcine skin for characterization of sunscreens. *Journal of Biomedical Optics 14*, 2 (2009), Article No.: 024027.

[28] WISSEL, T., STÜBER, P., WAGNER, B., BRUDER, R., SCHWEIKARD, A., AND ERNST, F. Enriching 3d optical surface scans with prior knowledge: tissue thickness computation by exploiting local neighborhoods. *Int. Journal of Computer Assisted Radiology and Surgery* (2015).

APPENDIX D

Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering

Research Article Vol. 56, No. 27 / September 20 2017 / Applied Optics 7679

applied optics

Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering

Jonathan Dyssel Stets,[1,†] Alessandro Dal Corso,[1,†] Jannik Boll Nielsen,[1] Rasmus Ahrenkiel Lyngby,[1] Sebastian Hoppe Nesgaard Jensen,[1] Jakob Wilm,[1] Mads Brix Doest,[1] Carsten Gundlach,[2] Eythor Runar Eiriksson,[1] Knut Conradsen,[1] Anders Bjorholm Dahl,[1] Jakob Andreas Bærentzen,[1] Jeppe Revall Frisvad,[1,*] and Henrik Aanæs[1]

[1]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Richard Petersens Plads, 2800 Kongens Lyngby, Denmark
[2]Department of Physics, Technical University of Denmark, Fysikvej, 2800 Kongens Lyngby, Denmark
*Corresponding author: jerf@dtu.dk

Received 23 May 2017; revised 15 August 2017; accepted 15 August 2017; posted 16 August 2017 (Doc. ID 295986); published 19 September 2017

Transparent objects require acquisition modalities that are very different from the ones used for objects with more diffuse reflectance properties. Digitizing a scene where objects must be acquired with different modalities requires scene reassembly after reconstruction of the object surfaces. This reassembly of a scene that was picked apart for scanning seems unexplored. We contribute with a multimodal digitization pipeline for scenes that require this step of reassembly. Our pipeline includes measurement of bidirectional reflectance distribution functions and high dynamic range imaging of the lighting environment. This enables pixelwise comparison of photographs of the real scene with renderings of the digital version of the scene. Such quantitative evaluation is useful for verifying acquired material appearance and reconstructed surface geometry, which is an important aspect of digital content creation. It is also useful for identifying and improving issues in the different steps of the pipeline. In this work, we use it to improve reconstruction, apply analysis by synthesis to estimate optical properties, and to develop our method for scene reassembly.

OCIS codes: (150.4232) Multisensor methods; (150.6910) Three-dimensional sensing; (150.1488) Calibration; (160.4760) Optical properties; (290.1483) BSDF, BRDF, and BTDF; (330.1690) Color.

https://doi.org/10.1364/AO.56.007679

1. INTRODUCTION

Several research communities work on techniques for optical acquisition of physical objects and their appearance parameters [1–5]. Thus, we are now able to acquire nearly any type of object and perform a computer graphics rendering of nearly any type of scene. The range of applications is broad and includes movie production [2], cultural heritage preservation [3], 3D printing [4], and industrial inspection [5]. A gap left by these multiple endeavors is a coherent scheme for acquiring a scene consisting of several objects that have very different appearance parameters, together with the reassembly of a digital replica of such a scene. Our objective is to fill this gap for the combination of transparent and opaque objects, as many real-world scenarios exhibit this combination. An example is a living room, like the one rendered in Fig. 1 (right). We propose a pipeline for acquiring and reassembling digital scenes from this type of heterogeneous real-world scene. In addition, our pipeline closes the loop by rendering calibrated images of the digital scene that are commensurable with photographs of the original physical scene (see Fig. 1, left). This allows for validation and fine-tuning of appearance parameters. The quantitative evaluation we get from pixelwise comparison of rendered images with photographs is a great improvement with respect to validation of the acquired digital representation of the physical objects.

When addressing the problem of acquiring a heterogeneous scene, there is an infinite variety of scenes and object types to choose from. So, to make our task feasible, we focus on scenes that combine glassware and non-transparent materials, more specifically, a white tablecloth and cardboard with a checkerboard pattern. We made these choices, as glass requires a different acquisition modality, the tablecloth bidirectional reflectance distribution function (BRDF) is spatially uniform but not necessarily simple, and the cardboard has simple two-color variation. The latter is particularly useful for observing how light

Fig. 1. To the left, we compare rendered images (top) with photographs (bottom). More views are available in Appendix A. The scenes to the left were digitized using our pipeline and include both glass objects and non-transparent objects (tablecloth and backdrop). To the right, we exemplify the use of our pipeline for virtual product placement using our digitized glass objects, with estimated optical properties and artifact-reduced removal of markers.

refracts through the glass. The chosen case is also of particular interest, since glass is present in many intended applications of optical 3D acquisition. Considering the highly multidisciplinary nature of our work, we have released our dataset (http://eco3d.compute.dtu.dk/pages/transparency). This facilitates further investigation by other researchers of the different steps of our pipeline with the possibility of a quantitative feedback at the end of the process.

A. Related Work and Contributions

Researchers occasionally compare renderings with photographs to provide a qualitative verification of a presented rendering technique. The work by Phong [6], Goral *et al.* [7], and Takagi *et al.* [8] are early examples of this trend. A procedure to bring a rendered image close to a photograph was first presented by Meyer *et al.* [9]. In this work, likeness of images was evaluated perceptually by human observers. Pixelwise comparison of photographs with rendered images is surprisingly uncommon. The few examples we have found are by Rushmeier *et al.* [10], Karner and Prantl [11], Pattanaik *et al.* [12], and Jones and Reinhart [13,14]. These examples build on the rendering framework described by Greenberg *et al.* [15]. Employing such a framework for more complex scenes is a long and tedious process [16]. The key issue is that a scene specification is expected as an input.

Several problems arise as a result of not having correspondence between the physical and the digital scene. Misalignment due to inaccurate scene and viewing geometry and inaccurate orientation of the lighting environment are some of the essential problems identified in previous work [17,18]. One way to deal with this problem is to calculate error for image patches when evaluating results [13,19,20]. As opposed to this, our digitization pipeline (Fig. 2) provides both reference photographs and correspondingly calibrated scene and viewing geometry so that pixelwise comparison becomes meaningful.

Pixelwise comparison of rendered images with photographs is not only useful for quantifying the photorealism of a rendering in terms of error measurements. We find it particularly useful for improving the digitization pipeline. The fact that our pipeline enables quantitative evaluation led us to more specific contributions in its different steps. These contributions are mostly in the reassembly and are as follows. (a) A cross-modality marker-based placement approach, enabling accurate placement of objects scanned with one modality into scenes scanned with another modality. (b) A soft object deformation technique dealing with surface intersections after object placement, which is critical for scenes containing transparent or translucent objects. (c) A micropolygon labeling approach for assigning BRDFs to acquired geometry. (d) A color calibration scheme enabling use

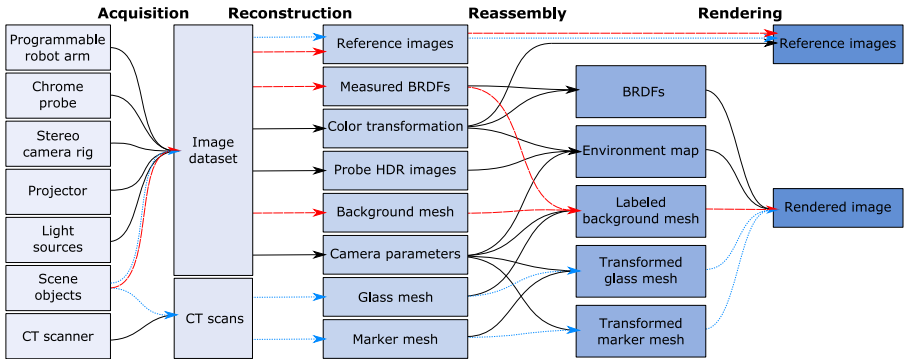


Fig. 2. Overview of our digitization pipeline in four main stages: acquisition, reconstruction, reassembly, and rendering. A video presentation of our pipeline is available in supplementary Visualization 1. Colored arrows show the path through the pipeline of transparent objects (dotted blue) and non-transparent objects (dashed red).

of spectral optical properties for calculating reflectance, transmittance, and absorption. (e) Perspective unwrapping of mirror probe images to improve precision when the environment is not very distant. (f) Use of analysis by synthesis for fine-tuning physics-based optical properties.

Digitization is most often unimodal and tailored toward objects with a specific type of surface reflectance behavior [1]. While unimodal techniques are becoming more versatile [21–23], objects with a transparent material, such as glass, still pose challenging problems. Their reflectance behavior is so different that they require an entirely different modality, such as computed tomography (CT) [24]. The transparent object must then be removed from the scene to be scanned elsewhere. In the meantime, the surrounding scene can be scanned with a more common technique. However, as the transparent object takes most of its appearance from its surroundings, it must be repositioned in the surrounding scene (physically and digitally) if we are to take reference images for comparison with rendered images. The purpose of our scene reassembly is to address this type of issue.

Our digitization technique is multimodal. Currently, such techniques seem to exist only in the context of sensor fusion [25–27]. Here, the goal is to optimize reconstruction by fusing data from different sensor modalities with complementary characteristics. Even so, the different modalities see the same object and thus work for materials with a similar reflectance behavior. The challenge is then mostly in registration of the scans. In their final remarks and suggestions for future work, Weinmann and Klein [1] discuss possible ways of combining multiple techniques tailored to different types of surface reflectance. Our pipeline is a different way to take a step in this direction.

In summary, our work makes it possible to perform multimodal digitization and scene reassembly in such a way that rendered images of the reassembled scene can be quantitatively compared to photographs of the original. This enables us to provide the first empirically founded investigation of the appearance accuracy of objects digitized using a non-optical scanner.

2. DIGITIZATION PIPELINE

We divide our pipeline into four stages: (1) acquisition, (2) reconstruction, (3) reassembly, and (4) rendering. Figure 2 provides an overview. As illustrated, transparent objects (dotted blue arrows) and non-transparent objects (dashed red arrows) take different paths through the pipeline. The acquisition stage includes structured light scanning of non-transparent objects, CT scanning of transparent objects, gonioreflectometric reflectance measurements, and photographic capture of environment, color chart, and scene reference images. Figure 3 provides details of our workflow in these acquisition steps (except the simpler captures of environment and color chart). The second stage includes reconstruction of surface meshes, material BRDFs, and color space. The third stage is reassembly of the digital scene consisting of geometric objects, material appearance properties, and environment map. The fourth and final stage is rendering and comparison with reference images.

Our acquisition stage requires an elaborate hardware setup. We assemble the physical scene in a black light-proof enclosure. This has five LED light tubes for scene lighting, which we capture by high dynamic range (HDR) imaging of a light probe. To acquire non-transparent geometry inside this enclosure, we use a structured light scanner consisting of a toe-in stereo camera rig and a light projector mounted on a robotic arm [28,29]. We chose a converging camera configuration (toe-in) to increase the overlap of the fields of view so that we get a denser point cloud per stereo view. Together with an LED-based illumination arc, we also use this camera rig with exact control for measuring isotropic BRDFs. For transparent objects, we use a CT scanner. In the following subsections, we describe the individual steps of the pipeline with a focus on details required for reproducibility and on non-standard techniques that we introduce.

A. Camera Calibration and Settings

The camera system is calibrated using a standard technique [30]. Our calibration board is an 11 by 12 black-and-white checkerboard. For the intrinsic calibration (Pass 1 of Fig. 3, left), we include a large variety of views to estimate good lens distortion coefficients. To facilitate stereo calibration, we also ensure that both cameras have the calibration board fully in view. For extrinsic calibration (Pass 2 of Fig. 3, left), we balance good coverage of the scene and good coverage of the calibration board. Since we cannot change the camera system while collecting data, we chose a small aperture to ensure that background

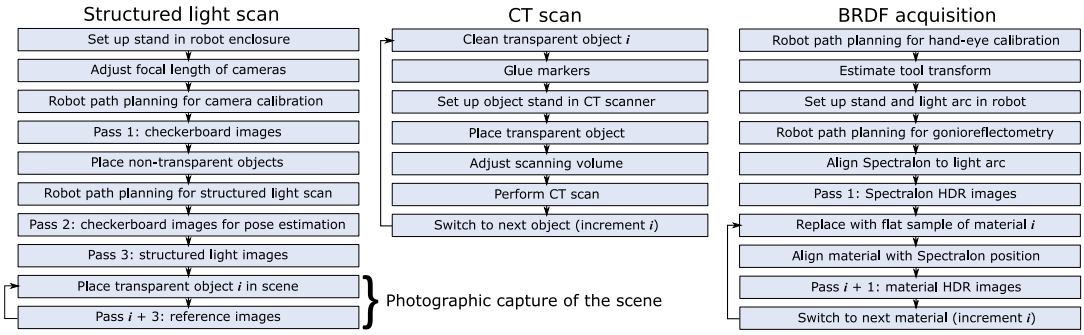


Fig. 3. Our workflow for scanning the geometry of non-transparent objects and collecting reference images (left), for scanning the geometry of transparent objects (middle), and for measuring material reflectance properties (right).

and projected structured light patterns are always in focus from all views. The full setup is in a dark room environment to eliminate external light, so we use a long shutter time (600 ms) to obtain sufficient exposure. A slight noise component is present in the images, but this is considered negligible. Finally, we use the estimated distortion coefficients to remove distortion from all images in the dataset so that subsequent algorithms may assume a pinhole camera model.

To avoid any compression or manipulation of the images by the camera software, in particular automatic color correction, we read the raw sensor data directly. We use bilinear interpolation to reconstruct RGB images from the raw Bayer pattern images. By doing this, we obtain a consistent RGB color space. Moreover, the raw sensor data are linear and correlate directly with radiometric quantities, which allows for better BRDF and environment map estimation in later stages of our pipeline.

We capture radiometrically relevant parts of our dataset in HDR by stacking multiple exposures [31]. More specifically, we stack 11 exposures at one-stop intervals ranging from 1 to 2048 ms. For the other parts of the dataset, we capture a single image at an exposure time of 600 ms.

B. Surface Reconstruction from Structured Light

We use a standard Gray code structured light approach to generate raw point clouds for a scene [32,33]. With camera parameters from the calibration, we transform these point clouds into the same world coordinate system.

To reconstruct one connected triangle mesh from the point clouds, we merge them into a single point cloud and perform screened Poisson reconstruction with trimming and an octree depth of nine [34]. This technique requires point normals, so before the merging, we generate normals for each point cloud as follows. We resample the point cloud down to 100,000 vertices via Poisson disk sampling [35] and then compute normals via planar fitting to a nearest neighborhood of 500 points (∼16 mm radius). We then reorient all the normals according to the location of one of the cameras and transfer them back onto the original point cloud. This procedure ensures smooth continuous normals, necessary for a good performance of the mesh reconstruction algorithm. As we rely on smoothing, we cannot reconstruct features in the mesh with the same physical size as the alignment error accumulated from structured light and calibration. The aim of the chosen constants was to preserve features by striking a balance between too noisy and too smooth. The operability of the pipeline is, however, not sensitive to the choice of these constants.

C. Material BRDF Reconstruction

We assume that all non-transparent materials in the scene are opaque and isotropic, so we model their reflectance properties by BRDFs. To acquire a BRDF, we combine traditional canonical gonioreflectometric sampling [36] with a BRDF interpolation (reconstruction) technique [37]. We follow the workflow outlined in Fig. 3 (right). A light arc illuminates material samples from 11 unique inclinations, evenly distributed from 7.5° up to 90° with 7.5° steps. We place a flat material sample at the center of the circle partly traced by the light arc. Using the cameras mounted on the robot, we then measure radiance reflected by the sample across one octant of a sphere. The center of this sphere coincides with that of the light arc, while its radius is slightly larger to avoid collision between the robot and the arc. The robot moves in steps of 7.5° and captures 11 HDR images of the sample per step, one for each light direction. In total, this yields 2,783 HDR images per material. We avoid tangential and zenith viewing directions (90° and 0°, respectively). In the former case, no reflected radiance should be visible, while in the latter the light arc occludes the view of the sample.

The 2,783 observations are too few to faithfully represent the BRDF of a material in a photorealistic rendering. We need an interpolation scheme to fill the entire $(90 \times 90 \times 180)$ Mitsubishi Electric Research Laboratories (MERL) format BRDF look-up table [38]. The reconstruction method by Nielsen *et al.* [37] is our interpolation scheme. First, we use each of the 100 BRDFs in the MERL dataset [38] as sample points in a $90 \cdot 90 \cdot 180 = 1,458,000$ dimensional space. The nonlinear mapping of Nielsen *et al.* [37] is then applied to each of the samples. The mapped samples are ordered as rows of a matrix $\mathbf{X} \in \mathbb{R}^{m \times d}$ where m is the number of BRDF samples, and d is the dimension of the space. The zero-mean matrix is computed as $\mathbf{X} - \bar{\mathbf{x}}$, with $\bar{\mathbf{x}}$ being the sample mean. From this, the singular value decomposition $\mathbf{X} - \bar{\mathbf{x}} = \mathbf{U}\Sigma\mathbf{V}^T$ is used to compute the eigenvectors and eigenvalues of the covariance matrix of $\mathbf{X} - \bar{\mathbf{x}}$, which are given as the columns of $\mathbf{V}$ and the diagonal elements of Σ, respectively. This is effectively a principal component analysis (PCA), where the eigenvectors are the principal components. A matrix composed of the scaled principal components as columns are computed as $\mathbf{Q} = \mathbf{V}\Sigma$.

Now, the full BRDF can be reconstructed from this principal component space by projection. Let $x' \in \mathbb{R}^n$ be n BRDF observations measured for a given material. Then, let $\bar{\mathbf{x}}' \in \mathbb{R}^n$ be the mean values and $\mathbf{Q}' \in \mathbb{R}^{n \times k}$ be the scaled eigenvectors corresponding to the direction pairs of those n observations. A vector $\mathbf{c}$ that spans the full space can be constructed by finding the linear combinations of principal components that best approximate the n observations. We do this by solving the linear least-squares optimization problem given by

$$\begin{aligned}\mathbf{c} &= \underset{\mathbf{c}}{\operatorname{argmin}} \|(\mathbf{x}' - \bar{\mathbf{x}})' - \mathbf{Q}'\mathbf{c}\|^2 + \eta\|\mathbf{c}\|^2 \\ &= (\mathbf{Q}'^T\mathbf{Q}' + \eta\mathbf{I})^{-1}\mathbf{Q}'^T(\mathbf{x}' - \bar{\mathbf{x}}').\end{aligned}$$

Note that by adding a penalty η to the norm of $\mathbf{c}$, this effectively becomes a Tikhonov regularized least squares. Now, the full, mapped BRDF is reconstructed as $\mathbf{x} = \mathbf{Q}\mathbf{c} + \bar{\mathbf{x}}$. The inverse of the nonlinear mapping applied to $\mathbf{X}$ is applied to $\mathbf{x}$ to get the actual, unmapped BRDF of the material. The described approach is applied to every single non-transparent material in the scene in order to obtain models of their reflectance properties.

This approach assumes that the MERL database encompasses the class of materials present in the scene. Effectively, this is a practical compromise between dense, unbiased, canonical BRDF sampling and fast, inferred BRDF sampling. This enables us to obtain high confidence BRDFs in a matter of a few hours.

D. Surface Reconstruction from CT

In our dataset, we have three glass objects: a sphere, a teapot (pot and lid), and a bowl (bowl and lid), for a total of five pieces. All objects have spherical plastic markers glued onto

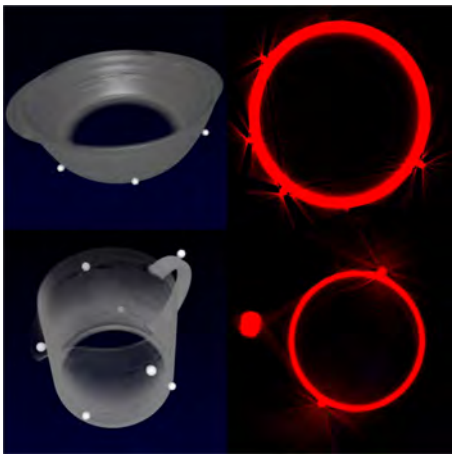


Fig. 4. CT scans of the bowl (top row) and the teapot (bottom row) with markers glued onto them. In the left column, visualized using a 1D transfer function. Note the different density of the markers. In the right column, a slice scaled to display streak artifacts.

their outer surfaces. We CT scan each glass piece to obtain x-ray radiographs and use the CT PRO 3D reconstruction software from Nikon Metrology to obtain a volumetric image for each piece. The resolution of the reconstructed volume is up to 1000^3 voxels. Due to beam hardening, high-density differences between materials lead to streak artifacts [39], especially around our markers and at the tops and bottoms of the objects (see Fig. 4). We account for these artifacts in the volumetric segmentation.

From a CT scan, we generate two triangular meshes with vertex normals: one for the glass object and one the plastic markers. Figure 5 provides an overview of our procedure. We start with the markers, which appear as elements of higher density in the scan. We preprocess the scan by clamping all the values under a certain threshold to zero and then create a mesh using dual contouring [40]. Generating the glass mesh is more cumbersome. We also use dual contouring in this case, but because of the streak artifacts (Fig. 4), it is not possible to isolate the glass mesh via a threshold. Instead, we use a lower threshold that removes only noise, then estimate the marker positions and use these to remove the markers from the glass mesh.

To estimate marker positions, we determine a series of center/radius pairs $(\mathbf{c}_i, r_i)$ by fitting a multi-sphere model to the marker mesh vertices using a tuned random sample consensus (RANSAC) algorithm [41]. We then carve a hole by excluding all the triangles that are inside a sphere with center $\mathbf{c}_i$ and radius $(1+\epsilon)r_i$, where ϵ is usually in the 0.5–0.75 range. We store the marker positions $\mathbf{c}_i$ so that we can use them to transform from the local coordinate system of the glass object to the world coordinate system (see Section 2.F).

After removing the markers, the glass meshes still have aliasing artifacts. To deal with this issue, we first decimate the mesh down to 1% of the original vertices via quadric edge collapse. The holes are then easy to close by identifying the edge loops surrounding each hole and filling these with triangles. We then

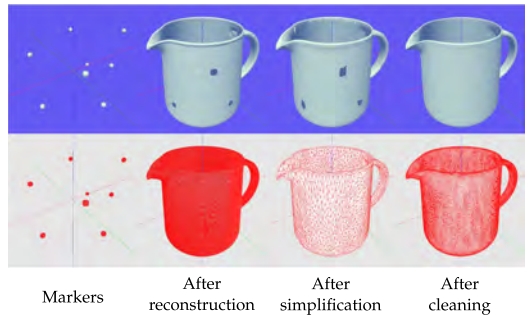


Fig. 5. Reconstruction from CT with stages illustrated using Phong shading (top row) and wireframe shading (bottom row). After estimating the marker mesh (first column) and fitting spheres to the markers, we reconstruct the object mesh (second column). To eliminate noise, we first simplify the mesh (third column) and then close the holes and apply our subdivision-decimation loop to get the final object mesh (fourth column).

introduce a subdivision-decimation loop with alternating $\sqrt{3}$-subdivision [42] and decimation to 33% of the original vertices. We perform this subdivision-decimation operation four times to obtain a cleaned mesh. The decimation removes unwanted high-frequency features from the mesh. Thus, we generate smooth meshes at the cost of some geometric precision. We are again trying to strike a balance between reconstruction error and too much smoothing. In Section 4, we compare our method with a different cleaning procedure that better preserves geometry.

E. Scene Reassembly for Non-transparent Objects

Two operations are necessary to prepare the background mesh for rendering: labeling and deformation. In the labeling, our objective is to identify BRDFs and label each face of the mesh with a BRDF. Assuming a scene with a small number of known BRDFs, we apply edge detection and watershed on the images of the scene to segment BRDF boundaries. Shadows, specular highlights, and different viewing angles of the scene complicate fully automatic BRDF identification. Our approach gets us most of the way, but we manually correct any residual misclassification. Figure 6 shows a label image produced by our labeling technique.

The label images can be used in multi-view projective texturing of the background mesh. However, we would like to precompute the view and label selection instead of doing

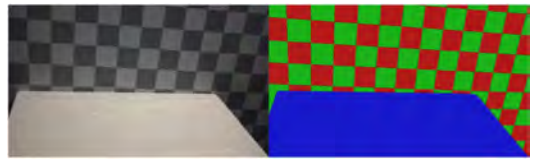


Fig. 6. Labeling of the image to the left results in the label image to the right. Each color in the label image represents a label that we assign a BRDF to. The black edges between labels indicate areas where we apply a nearest neighbor method.

Fig. 7. Subdividing the mesh dissolves unwanted boundary sawtooth artifacts that originate from the BRDF labeling.

it millions and millions of times while rendering. To avoid *uv*-unwrapping of the mesh for storing precomputed labels, we take an approach inspired by micropolygon rendering [43]. We project each vertex of a face onto the label images of the scene and select the face BRDF according to the image label that most of the face vertices were projected to. If a vertex projects to an unknown label, we resolve it by a nearest neighbor search. Since faces around material boundaries overlap multiple materials, we get sawtooth artifacts. We dissolve these by subdividing the mesh until the rendered triangles are smaller than the surface area observed in a pixel; see Fig. 7.

When applying physically based rendering, we observed intersections between background scene and glass meshes. This could be due to small errors in reconstruction and positioning, or perhaps the harder glass objects press down the tablecloth when placed for reference imaging. It causes significant visual artifacts, since the rendering exposes all surfaces of a transparent object. To eliminate these artifacts, we accommodate the hard object (glass) by deforming the soft object (tablecloth); see Fig. 8. To deform the soft object, we need a "down" direction in which to push the vertices. We first find contact vertices. These are vertices in each mesh that are close to any vertex of the other mesh. We consider vertices close if the distance between them is less than 7% of the bounding box diagonal of the hard object. Using least squares regression, we fit a contact plane to the contact vertices of the soft object. We set the sign of the contact plane normal so that the upper half-space contains the center of the hard object bounding box. Projection of a contact vertex to the normal of the contact plane then measures the height of the vertex. For each soft object contact vertex $\mathbf{x}$, we find the nearest hard object contact vertices and push $\mathbf{x}$ down below the lowest one of these.

F. Scene Reassembly for Transparent Objects

To reposition the glass objects in the scene, we rigidly transform the meshes reconstructed from CT to the world coordinate system of the background mesh. We obtain this transformation by matching markers in the stereo images with the marker coordinates $\mathbf{c}_i$ computed during reconstruction from CT (see Section 2.D).

Fig. 8. Deformation of background mesh, where we push the background vertices down to avoid mesh intersection.

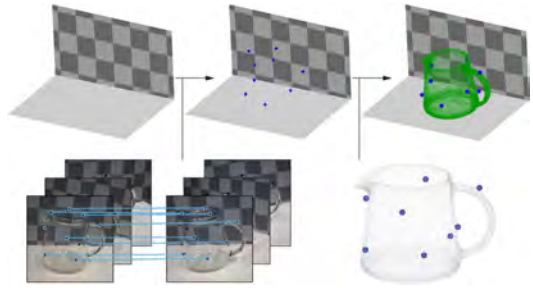


Fig. 9. Repositioning a CT scanned object in the background scene. We identify and match the markers in the stereo image pairs and calculate their corresponding 3D points. Pairing these with marker coordinates from the CT scans, we transform the CT scanned piece of an object into the world coordinate system.

To find the markers, we employ a size invariant circle Hough transform [44]. This works well for our dataset, where the markers show high contrast against their surroundings. We match markers in the left and the right images via Sampson distance [45]. Using this technique, markers on the same epipolar line lead to false positives, so we manually inspect the result. We also manually discard detected markers that are visible through the glass, as the refraction would lead to incorrect positioning. Markers in both stereo images with no match are discarded. The result is a set of matched markers in image coordinates, as seen in Fig. 9 (bottom left). We then triangulate the matched markers from the stereo views and gather them in clusters of 3D points. We remove outliers via their distances from the cluster centers, and for each cluster we select the point with the lowest reprojection error. An example of the points and clustering is shown in Fig. 9 (top middle).

We manually pair the 3D marker coordinates from the images with the marker coordinates $\mathbf{c}_i$ from the CT scans. We perform Procrustes analysis [46] on the two point sets, excluding reflection, since we assume a rigid transformation applied to each vertex of the mesh. The bowl and the teapot are composed of multiple pieces. For these objects, we compute the transformation individually for each piece. The result of the object transformed into the scene is shown in Fig. 9 (top right). We found that in order to have low error in the transformation, the chosen markers should sample the surface evenly and be visible from most views.

G. Color Calibration

Images are quantitatively comparable only if they live in the same color space. Thus, we must ensure that our radiometry-dependent data, namely reference images, environment map, and BRDFs, are in the same color space. We do this by imaging a color chart of precisely known colors. More specifically, we use second-degree root-polynomial color correction [47] based on a 24-patch ColorChecker Classic from X-Rite. This provides a matrix that transforms from camera RGB to XYZ, where we assume illuminant D50 when specifying the XYZ values of the colorchecker. With the assumption of illuminant D50, we can transform colors to the CIE L∗a∗b∗ color

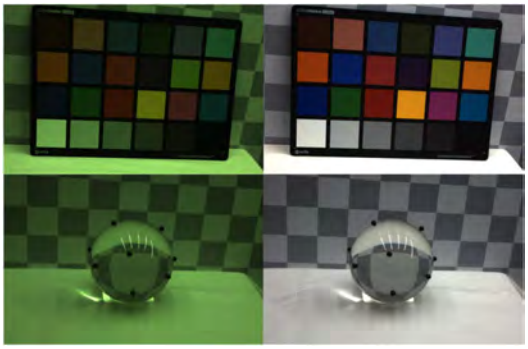


Fig. 10. Color calibration: raw images (left) and color corrected images (right). The camera sensor is particularly sensitive to green.

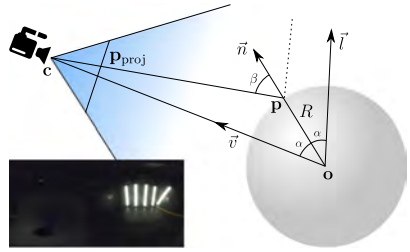


Fig. 11. Unwrapping of a spherical probe. We know the sphere radius R from specification, the camera position $\mathbf{c}$ through calibration, and the sphere center $\mathbf{o}$ by triangulation. Radiance at $\mathbf{p}_{\text{proj}}$ in our image then corresponds to the environment map direction $\vec{l}$. The result for the robot enclosure is in the lower left corner in latitude–longitude panoramic format (here tone-mapped).

space and then compute color differences using the ΔE_{00} metric [48]. We use this to refine our result by minimizing ΔE_{00} using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [49]. The result is in Fig. 10. The average color difference is $\Delta E_{00} = 1.97 \pm 1.21$, which is larger than 1 JND (just noticeable difference) [50], but we find it acceptable.

Since we work with glass objects (and chrome; see Section 2.H), we need refractive indices to determine reflectance, transmittance, and absorption properties. Refractive indices can be found per wavelength in tables of research papers. To use such spectral optical properties together with our trichromatic image data, we integrate them to CIE RGB using the CIE RGB color-matching functions listed by Stockman and Sharpe [51]. It is important to normalize these functions [52] and to use RGB rather than XYZ [53]. This is because a refractive index is not a color, but rather a quantity that in trichromatic representation should resemble a sparse sampling of the spectrum. Thus, as recommended by other authors [54], we choose CIE RGB as our rendering color space. After transforming our image data from camera RGB to XYZ, we therefore convert them to CIE RGB [55]. As a final step, we apply Bradford chromatic adaptation [50], adapting to the originally assumed illuminant D50, so that renderings and reference images get closer to real-life appearance.

H. Environment Lighting

To capture the lighting observed in the reference images, we use a method similar to the mirror probe technique [56]. However, we use a pinhole camera model for probe image unwrapping instead of the standard orthographic model. Our pipeline enables this, as we have a calibrated camera and know its position relative to the photographed mirror probe. With the pinhole model, we obtain a more precise estimate of the environment lighting. The environment map is generated from HDR images and stored in latitude–longitude panoramic format [50]. We use a polished grade G100 chrome bearing ball as mirror probe.

An environment map represents an infinite area light and maps a direction to a texture element (a texel). To do unwrapping, we map each texel direction $\vec{l}$ to the corresponding pixel position $\mathbf{p}_{\text{proj}}$ in a light probe image. Given the configuration illustrated in Fig. 11, we have

$$\vec{v} = \frac{\mathbf{c}-\mathbf{o}}{\|\mathbf{c}-\mathbf{o}\|}, \quad \vec{n} = \frac{\vec{v}+\vec{l}}{\|\vec{v}+\vec{l}\|}, \quad \mathbf{p} = \mathbf{o} + R\vec{n}, \quad \mathbf{p}_{\text{proj}} = \mathbf{M}[\mathbf{p}^T \quad 1]^T,$$

where camera matrix $\mathbf{M}$ and camera position $\mathbf{c}$ are available from our calibration. The radius of the sphere R is available from the bearing ball specification, and we find the center of the sphere $\mathbf{o}$ by manually annotating the sphere and then triangulating it. We assume that the distance to the actual light along $\vec{l}$ is equal to the distance between camera and sphere $\|\mathbf{c}-\mathbf{o}\|$. This assumption works well in practice, leading to an error smaller than the uncertainty of $\mathbf{o}$ caused by the triangulation. With the original orthographic camera model, we can reconstruct the lighting for all directions except one ($-\vec{v}$). In our model, we cannot reconstruct the lighting for a set of directions ($\vec{n}\cdot\vec{v} \le R/\|\mathbf{c}-\mathbf{o}\|$), so we set them to black. Since we do our unwrapping in world space, we can combine contributions from multiple camera views with no need to align them afterwards.

The environment map is color corrected according to Section 2.G, which enables us to correct for the angularly dependent reflectance of chrome. The correction is to divide by Fresnel reflectance, which we compute during unwrapping. As input for Fresnel's equations, we use the angle β between $\mathbf{c}-\mathbf{p}$ and $\vec{n}$ and the complex refractive index of chrome [57] converted from spectrum to CIE RGB. The result is shown in the inset of Fig. 11.

I. Rendering

We render images using progressive unidirectional path tracing [58,59] implemented in OptiX [60]. The captured HDR environment map is the sole light source in our scene [56]. When rendering non-specular materials, we importance-sample the environment map to get direct illumination and use sampling of a cosine-weighted hemisphere to get indirect illumination. From our labeling, we have one BRDF attached to each triangle in our scene. For non-transparent objects, we use our measured BRDFs tabulated in the MERL format [38]. To terminate paths probabilistically, we use Russian roulette based on the

bihemispherical reflectance of each measured BRDF. This reflectance is calculated in a preprocessing step using Monte Carlo integration. We deal with transparent objects in the usual way, setting reflectance and transmittance according to Fresnel's equations of reflection and Bouguer's law of exponential attenuation. Given their small surface, we were unable to estimate a BRDF for the markers. Instead, we render them as glass with all refracted rays being absorbed.

3. ANALYSIS BY SYNTHESIS

The ability to render images comparable to photographs enables us to use our pipeline for improving parameter estimates through analysis by synthesis. As an example, we need a scaling factor for our HDR environment map, as it measures relative radiance [31]. We estimate this factor by taking ratios of references and renderings with the background scene alone. Another example is estimating real and imaginary parts of glass refractive indices. As analysis by synthesis is fundamentally ill posed [61], we take our outset in physics-based initial guesses such as Schott K5 crown glass (sphere and teapot) and soda lime glass (bowl). Spectral refractive indices for these glasses were obtained from an online database (http://refractiveindex.info) and converted to CIE RGB. All parameters were estimated using views different from the ones in our comparisons of renderings with references.

As an example of our analysis by synthesis, we plot the evolution of the root-mean-squared error (RMSE) for different renderings of the glass bowl in Fig. 12. For each rendering, we vary a trichromatic component of the absorption coefficient (which directly relates to the imaginary part of the refractive index). We identify a distinct minimum in the error for each channel, with a slightly larger uncertainty in the red channel. The minimum values in this figure were used in our renderings of the glass bowl. We apply the same analysis to the teapot and the sphere.

Given an initial guess for a parameter, we can employ standard optimization algorithms, defining the RMSE between the reference and the rendering as a cost function to minimize. To reduce rendering times, the evaluation of the cost function can be calculated on a downsampled image or limited to a specific patch of the images. Various general optimization algorithms exist for minimizing expensive cost functions [62].

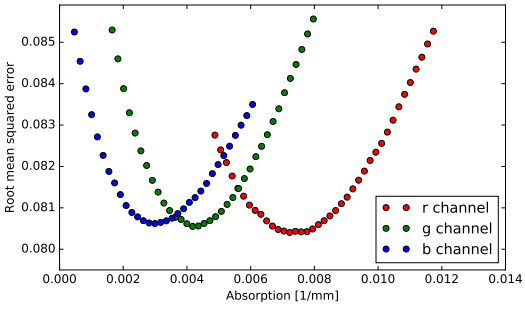


Fig. 12. Analysis by synthesis to estimate absorption of the glass bowl. We run renderings in low resolution and change the absorption in each color channel one at the time. In the case of the bowl, the blue channel is the most sensitive one.

4. RESULTS

Our scenes consist of a backdrop, a stand, and a glass object (with markers) placed on the stand. The backdrop is a 30 by 20 white-and-gray checkerboard print on 120 cm by 80 cm semi-matte cardboard, and the stand is a tabletop with a white cloth. An example scene is depicted in Fig. 13. We implemented our reconstruction and reassembly procedures as a modular software pipeline and computed all rendered images using our path tracer. As illustrated in Fig. 2 and mentioned in Section 2.G, we color correct both rendered images and reference images to have a meaningful perceptual comparison. Figure 14 compares markers in a reference image with rendered markers to validate our marker positioning. For the teapot, the average distance between the markers from stereo and the transformed markers from CT is 0.43 mm.

Figure 15 presents pixelwise comparisons of reference images and rendered images. The error images allow us to spot subtle differences not easily noticed in a perceptual comparison, such as the slight misalignments in geometry and highlights. As reference photographs were not captured in HDR, we clamp the renderings correspondingly. This means that areas of strong light intensity, such as highlights and intense caustics, appear black in the error images.

Fig. 13. Scene with checkerboard backdrop, lighting, glass teapot, and stand with table cloth observed by two cameras mounted on a 6-axis industrial robot arm.

Fig. 14. Markers rendered in blue and added to the reference image to validate marker positions by looking at pixel offsets.

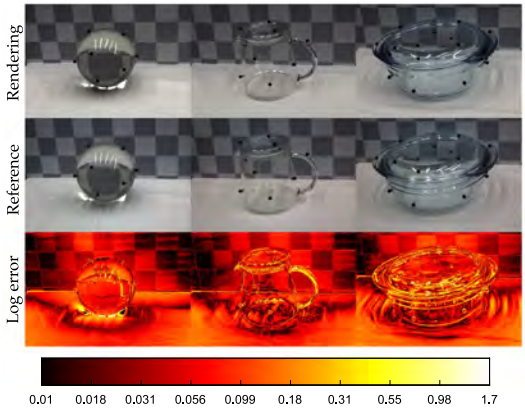


Fig. 15. Pixelwise error for three rendering-reference pairs. Error is the ℓ^2-norm of 32-bit per channel RGB images, visualized using a base 10 logarithmic scale.

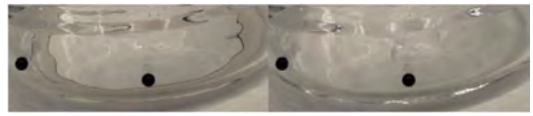


Fig. 17. Zoom-in of Figs. 16(b) and 16(c) to emphasize the effect of our background deformation.

Fig. 18. Zoom-in of Fig. 16(c) and 16(d) to emphasize the effect of our perspective unwrapping of the environment map.

Figure 16 exemplifies the impact on error images of some of our contributions. In Fig. 16(a), we reposition only the glass object in the background scene and apply color correction (Sections 2.F and 2.G). This means that we use Lambertian materials (with bihemispherical reflectances from the measured BRDFs), an orthographic unwrapping model of the environment map, and no chrome reflectance correction or analysis by synthesis optimization. We compare to the reference image in Fig. 16(g), with error images as in Fig. 15. Figure 16(b) shows the impact of using measured BRDFs (Section 2.C), resulting in a more accurate representation of the folds of the cloth in the background scene (top image) and an overall reduction of the error (bottom image). In Fig. 16(c), we add deformation of the background mesh (Section 2.E), which ensures that the background mesh does not poke through the glass surface (see a close-up in Fig. 17). Additionally, we can see how this improves the error on the lid of the bowl, because of refraction of light in the glass. The next step, Fig. 16(d), shows the impact of our modified environment map unwrapping (Section 2.H) against the standard orthographic unwrapping rotated according to our camera parameters. A close-up is available in Fig. 18. Our modified unwrapping provides a better shape and alignment of highlights and caustics. Partially due to the assumption of infinitely distant environment light, some alignment artifacts persist. In Fig. 16(e), we show the effect of correcting for chrome reflectance in our environment map reconstruction. Quantitatively, this changes the distribution of the error (bottom image). On the cloth, the exposure increases, exposing the caustics misalignment. On the backdrop, the error reduces. Interestingly, the structural similarity index (SSIM) improves while the RMSE worsens. Finally, in Fig. 16(f), we use analysis by synthesis to adjust glass absorption. This improves the glass appearance, but it also leads to slight color changes in other parts of the scene due to indirect light paths. Because of this global influence, the analysis by synthesis introduces slightly too much absorption to compensate for the slightly too bright tablecloth.

As an example of how our pipeline can be used to validate existing algorithms, we investigate the case of glass object reconstruction. In Fig. 19, we compare two different reconstruction

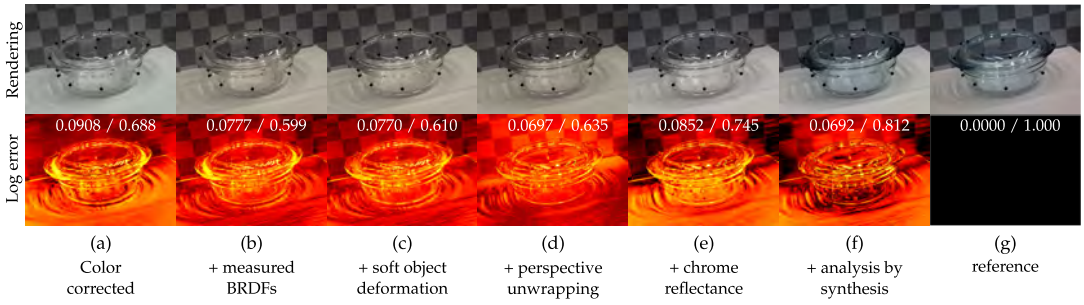


Fig. 16. Qualitative (top) and quantitative (bottom) step-by-step evaluation of our reassembly techniques. The log error images have the same format as in Fig. 15 and the reference photograph is in the rightmost column (g). In each column, we provide root-mean-squared error and structural similarity index (RMSE/SSIM). Both measures attain their best scores in our final result (f).

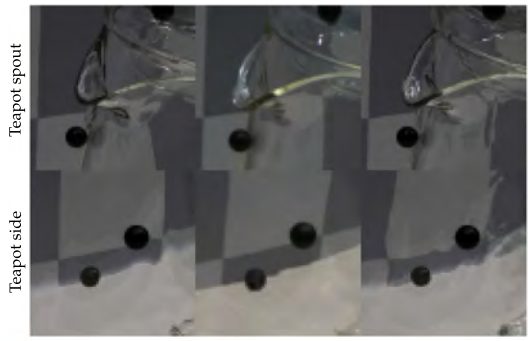


Fig. 19. Trade-off in mesh reconstruction. If we smooth more, we get less distortion in the refractions, but less precision in the mesh geometry. From left to right: Rendering with smoothing, reference image, rendering without smoothing.

methods with a focus on two parts of the teapot scene. Smooth reconstruction refers to the procedure described in Section 2.D. The other procedure is simply to decimate the reconstructed mesh to 2.5% of the original vertices and apply Taubin smoothing [63]. This removes the high frequencies of the noise, but much noise is still present in the midranges, leading to wobbly refractions. Our method in Section 2.D reduces far more noise, but this is at the cost of greater changes to the overall shape. We note that a refractive object with a simple geometry is very hard to reconstruct automatically if fidelity and almost no noise are both required.

5. DISCUSSION

Since our pipeline enables us to compare renderings with photographs, we can identify problems in acquisition, reconstruction, and rendering that would otherwise have been hard to find. Camera calibration issues, for example, reveal themselves as error lines along edges (visible in Fig. 20). Color calibration issues reveal themselves as color shift. Such issues led us to more careful camera calibration procedures and the choice of root-polynomial color correction. Qualitative comparisons revealed artifacts in surface reconstruction, mesh intersections calling for deformation, misplacement of highlights, color shift due to chrome reflectance, and missing absorption in renderings (Figs. 16–19). Quantitative comparisons confirmed

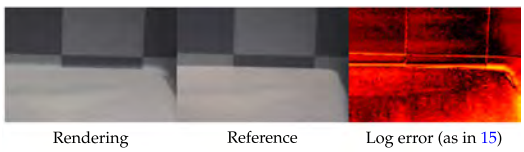


Fig. 20. Material transitions: error lines along checker edges and along the boundary between tablecloth and backdrop.

Fig. 21. Effect of separating markers from glass (refracted light close to marker) and of not accounting for subsurface scattering (dark areas close to caustics).

improvement due to perspective unwrapping of light probe images and led to analysis by synthesis.

The comparison with reference photographs before and after deformation (Fig. 17) to some extent validates our soft object deformation technique. Further validation would be desirable, but it is difficult to come up with a different experiment. Some kind of soft, durable memory foam with a scannable surface would be required, as the soft object would otherwise change shape again once the hard object is removed. Our validation supports only that the cloth appearance (as observed through glass) is represented more faithfully after deformation.

We found analysis by synthesis useful for estimating parameters with an outset in physics-based initial guesses. The results in Fig. 12 show that we can estimate optical properties for a given material and use them in a different setting (right part of Fig. 1). The precision of the estimation varies with the impact of the property on the overall error, and the estimated parameters may compensate for unrelated errors. In this regard, specific scene configurations could be used to favor estimation of a particular parameter. The most important limitation of our method is that we describe materials as large patches of isotropic BRDFs. In our renderings, this assumption works well for the checkerboard backdrop but not for the cloth, where we have both subsurface scattering effects and probably anisotropy due to the weave structure of the cloth. Figure 21 reveals that the rendered image is too dark in areas surrounding caustics. As seen in the light refracted through the sphere in the vicinity of the marker, our processing of the glass object to separate glass from markers causes some imprecision in the geometry. We believe this mainly influences the shape of the caustic. The bleeding of the caustic to areas that are much darker in the rendered images looks like backscattering from the table beneath the cloth. We refer to this as a kind of subsurface scattering.

Another limitation is seen at the transition between non-connected elements. It is visible in the renderings at the boundary between the cloth and the backdrop (see Fig. 20). The problem derives from the fact that the cloth and the backdrop were too close to each other during dataset acquisition. This resulted in the Poisson mesh reconstruction interpreting them as a continuous object instead of two separate ones. The problems around markers (Fig. 21) are also due to transition of materials. The marker removal and whole closing in the glass surface reconstruction interrupts the original shape of the surface. Furthermore, the markers are glued onto the glass surface, and the glue is not considered in the reconstruction and renderings. The marker glue problem is magnified by the glass refraction.

6. CONCLUSION

We have proposed a pipeline for multimodal scene digitization. Our work addresses the entire process from acquisition of the original objects, through reassembly of the digital scene, to accurate modeling of camera and environment. While the pipeline required several non-trivial steps, the benefits are correspondingly great, since we can perform pixelwise comparisons between rendered images and photographs of the corresponding physical scene. This means that we have the means to quantitatively assess the accuracy of an acquired model based on comparison with empirical evidence. We believe this kind of quantitative assessment has not previously been possible for transparent objects. In applications such as cultural heritage preservation and industrial inspection, where the accuracy of a digitization is important, such comparison with empirical evidence is crucial.

To the best of our knowledge, our work is also the first work to quantify the photorealism of a heterogeneous scene requiring multimodal acquisition.

Our dataset is publicly available so that others can test new techniques for the different steps of the pipeline with quantitative feedback based on photorealistic rendering. The fact that one can use off-the-shelf rendering techniques for improving the different steps of a multimodal digitization pipeline is perhaps the most important benefit of our work. An application of the full pipeline is the virtual product placement in Fig. 1. Another important application is the estimation of radiometric properties through analysis by synthesis. The ability to accurately estimate optical properties through computation rather than measurement, which might require specialized equipment, is likely to greatly simplify the digitization of radiometrically complex objects. In this paper, we estimated absorption and refractive indices of transparent objects, but analysis by synthesis could be equally useful for other materials with non-trivial BRDFs. This is another key benefit of our work that we believe is well worth exploring in the future.

APPENDIX A

Figure 22.

Fig. 22. Comparison of renderings and photographs as in Fig. 1 (left), but with more views.

Funding. Innovation Fund Denmark (IFD) (3067-00001B, 5163-00001B, 5163-00003B, 75-2014-1).

†These authors contributed equally for this work.

REFERENCES

1. M. Weinmann and R. Klein, "Advances in geometry and reflectance acquisition (course notes)," in *SIGGRAPH Asia 2015 Courses* (ACM, 2015).
2. P. Debevec, "The light stages and their applications to photoreal digitial actors," in *SIGGRAPH Asia 2012 Technical Briefs* (2012).
3. L. Gomes, O. R. P. Bellon, and L. Silva, "3D reconstruction methods for digital preservation of cultural heritage: a survey," Pattern Recogn. Lett. **50**, 3–14 (2014).
4. L. Zhang, H. Dong, and A. E. Saddik, "From 3D sensing to printing: a survey," ACM Trans. Multimedia Comput. Commun. Appl. **12**, 27 (2016).
5. J. B. Nielsen, E. R. Eiriksson, R. L. Kristensen, J. Wilm, J. R. Frisvad, K. Conradsen, and H. Aanæs, "Quality assurance based on descriptive and parsimonious appearance models," in *Workshop on Material Appearance Modeling (MAM)* (The Eurographics Association, 2015), pp. 21–24.
6. B. T. Phong, "Illumination for computer generated pictures," Commun. ACM **18**, 311–317 (1975).
7. C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, "Modeling the interaction of light between diffuse surfaces," SIGGRAPH Comput. Graph. **18**, 213–222 (1984).
8. A. Takagi, H. Takaoka, T. Oshima, and Y. Ogata, "Accurate rendering technique based on colorimetric conception," SIGGRAPH Comput. Graph. **24**, 263–272 (1990).
9. G. W. Meyer, H. E. Rushmeier, M. F. Cohen, D. P. Greenberg, and K. E. Torrance, "An experimental evaluation of computer graphics imagery," ACM Trans. Graph. **5**, 30–50 (1986).
10. H. Rushmeier, G. Ward, C. Piatko, P. Sanders, and B. Rust, "Comparing real and synthetic images: some ideas about metrics," in *Conference: Proceedings of the Eurographics Workshop Rendering Techniques* (Springer, 1995), pp. 82–91.
11. K. F. Karner and M. Prantl, "A concept for evaluating the accuracy of computer generated images," in *Spring Conference on Computer Graphics (SCCG)* (1996).
12. S. N. Pattanaik, J. A. Ferwerda, K. E. Torrance, and D. P. Greenberg, "Validation of global illumination solutions through CCD camera measurements," in *Color Imaging Conference (CIC)* (1997), pp. 250–253.
13. N. L. Jones and C. F. Reinhart, "Parallel multiple-bounce irradiance caching," Comput. Graph. Forum **35**, 57–66 (2016).
14. N. L. Jones and C. F. Reinhart, "Experimental validation of ray tracing as a means of image-based visual discomfort prediction," Build. Environ. **113**, 131–150 (2017).
15. D. P. Greenberg, K. E. Torrance, P. Shirley, J. Arvo, J. A. Ferwerda, S. Pattanaik, E. Lafortune, B. Walter, S.-C. Foo, and B. Trumbore, "A framework for realistic image synthesis," in *SIGGRAPH 97* (ACM/Addison-Wesley, 1997), pp. 477–494.
16. F. Drago and K. Myszkowski, "Validation proposal for global illumination and rendering techniques," Comput. Graph. **25**, 511–518 (2001).
17. C. Ulbricht, A. Wilkie, and W. Purgathofer, "Verification of physically based rendering algorithms," Comput. Graph. Forum **25**, 237–255 (2006).
18. J. Meseth, G. Müller, R. Klein, F. Röder, and M. Arnold, "Verification of rendering quality from measured BTFs," in *Applied Perception in Graphics and Visualization (APGV)* (ACM, 2006), pp. 127–134.
19. A. I. Ruppertsberg and M. Bloj, "Rendering complex scenes for psychophysics using RADIANCE: how accurate can you get?" J. Opt. Soc. Am. A **23**, 759–768 (2006).
20. A. Dal Corso, J. R. Frisvad, T. K. Kjeldsen, and J. A. Bærentzen, "Interactive appearance prediction for cloudy beverages," in *Workshop on Material Appearance Modeling (MAM)* (The Eurographics Association, 2016), pp. 1–4.

21. B. Tunwattanapong, G. Fyffe, P. Graham, J. Busch, X. Yu, A. Ghosh, and P. Debevec, "Acquiring reflectance and shape from continuous spherical harmonic illumination," ACM Trans. Graph. **32**, 109 (2013).
22. T. Nöll, J. Köhler, G. Reis, and D. Stricker, "Fully automatic, omnidirectional acquisition of geometry and appearance in the context of cultural heritage preservation," J. Comput. Cult. Herit. **8**, 2 (2015).
23. H. Wu, Z. Wang, and K. Zhou, "Simultaneous localization and appearance estimation with a consumer RGB-D camera," IEEE Trans. Vis. Comput. Graph. **22**, 2012–2023 (2016).
24. I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich, "Transparent and specular object reconstruction," Comput. Graph. Forum **29**, 2400–2426 (2010).
25. A. Kolb, J. Zhu, and R. Yang, "Sensor fusion," in *Digital Representation of the Real World*, M. A. Magnor, O. Grau, O. Sorkine-Hornung, and C. Theobalt, eds. (CRC Press, 2015), Chap. 9, pp. 133–150.
26. V. Bhateja, H. Patel, A. Krishn, A. Sahu, and A. Lay-Ekuakille, "Multimodal medical image sensor fusion framework using cascade of wavelet and contourlet transform domains," IEEE Sens. J. **15**, 6783–6790 (2015).
27. A. Pamart, O. Guillon, J.-M. Vallet, and L. De Luca, "Toward a multimodal photogrammetric acquisition and processing methodology for monitoring conservation and restoration studies," in *Eurographics Workshop on Graphics and Cultural Heritage* (The Eurographics Association, 2016), pp. 207–210.
28. H. Aanæs and A. B. Dahl, "Accuracy in robot generated image data sets," in *Scandinavian Conference on Image Analysis (SCIA)*, Lecture Notes in Computer Science (Springer, 2015), Vol. **9127**, pp. 472–479.
29. H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," Int. J. Comput. Vis. **120**, 153–168 (2016).
30. Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 1330–1334 (2000).
31. P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *SIGGRAPH 97* (ACM/Addison-Wesley, 1997), pp. 369–378.
32. J. L. Posdamer and M. Altschuler, "Surface measurement by space-encoded projected beam systems," Comput. Graph. Image Process. **18**, 1–17 (1982).
33. J. Geng, "Structured-light 3D surface imaging: a tutorial," Adv. Opt. Photon. **3**, 128–160 (2011).
34. M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," ACM Trans. Graph. **32**, 29 (2013).
35. M. Corsini, P. Cignoni, and R. Scopigno, "Efficient and flexible sampling with blue noise properties of triangular meshes," IEEE Trans. Visualization Comput. Graph. **18**, 914–924 (2012).
36. J. F. Murray-Coleman and A. M. Smith, "The automated measurement of BRDFs and their application to luminaire modeling," J. Illum. Eng. Soc. **19**, 87–99 (1990).
37. J. B. Nielsen, H. W. Jensen, and R. Ramamoorthi, "On optimal, minimal BRDF sampling for reflectance acquisition," ACM Trans. Graph. **34**, 186 (2015).
38. W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," ACM Trans. Graph. **22**, 759–769 (2003).
39. J. F. Barrett and N. Keat, "Artifacts in CT: recognition and avoidance," RadioGraphics **24**, 1679–1691 (2004).
40. T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of Hermite data," ACM Trans. Graph. **21**, 339–346 (2002).
41. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM **24**, 381–395 (1981).
42. L. Kobbelt, "$\sqrt{3}$-subdivision," in *SIGGRAPH 2000* (ACM/Addison-Wesley, 2000), pp. 103–112.
43. R. L. Cook, "The Reyes image rendering architecture," ACM SIGGRAPH Comput. Graph. **21**, 95–102 (1987).
44. T. Atherton and D. Kerbyson, "Size invariant circle detection," Image Vis. Comput. **17**, 795–803 (1999).
45. P. D. Sampson, "Fitting conic sections to "very scattered" data: an iterative refinement of the Bookstein algorithm," Comput. Graph. Image Process. **18**, 97–108 (1982).
46. J. C. Gower, "Generalized procrustes analysis," Psychometrika **40**, 33–51 (1975).
47. G. D. Finlayson, M. Mackiewicz, and A. Hurlbert, "Color correction using root-polynomial regression," IEEE Trans. Image Process. **24**, 1460–1470 (2015).
48. G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: implementation notes, supplementary test data, and mathematical observations," Color Res. Appl. **30**, 21–30 (2005).
49. J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. (Springer, 2006).
50. E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski, *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, 2nd ed. (Morgan Kaufmann/Elsevier, 2010).
51. A. Stockman and L. T. Sharpe, "The spectral sensitivities of the middle- and long-wavelength-sensitive cones derived from measurements in observers of known genotype," Vis. Res. **40**, 1711–1737 (2000).
52. J. R. Frisvad, N. J. Christensen, and H. W. Jensen, "Computing the scattering properties of participating media using Lorenz-Mie theory," ACM Trans. Graph. **26**, 60 (2007).
53. C. Ulbricht and A. Wilkie, "A problem with the use of XYZ colour space for photorealistic rendering computations," in *Colour in Graphics, Imaging, and Vision (CGIV)* (2006), pp. 435–437.
54. J. Meng, F. Simon, J. Hanika, and C. Dachsbacher, "Physically meaningful rendering using tristimulus colours," Comput. Graph. Forum **34**, 31–40 (2015).
55. H. S. Fairman, M. H. Brill, and H. Hemmendinger, "How the CIE 1931 color-matching functions were derived from Wright-Guild data," Color Res. Appl. **22**, 11–23 (1997).
56. P. Debevec, "Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography," in *SIGGRAPH 98* (ACM, 1998), pp. 189–198.
57. A. D. Rakić, A. B. Djurišić, J. M. Elazar, and M. L. Majewski, "Optical properties of metallic films for vertical-cavity optoelectronic devices," Appl. Opt. **37**, 5271–5283 (1998).
58. J. T. Kajiya, "The rendering equation," ACM SIGGRAPH Comput. Graph. **20**, 143–150 (1986).
59. M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*, 3rd ed. (Morgan Kaufmann/Elsevier, 2017).
60. S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich, "OptiX: a general purpose ray tracing engine," ACM Trans. Graph. **29**, 66 (2010).
61. M. Hejrati and D. Ramanan, "Analysis by synthesis: 3D object recognition by object reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 2449–2456.
62. D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," J. Global Optim. **13**, 455–492 (1998).
63. G. Taubin, "A signal processing approach to fair surface design," in *SIGGRAPH 95* (ACM, 1995), pp. 351–358.

APPENDIX E

Our 3D Vision Data-Sets in the Making

Our 3D Vision Data-Sets in the Making

H. Aanæs[1] K. Conradsen[1] A. Dal Corso[1] A. B. Dahl[1] A. Del Bue[2] M. Doest[1]
J. R. Frisvad[1] S. H. N. Jensen[1] J. B. Nielsen[1] J. D. Stets[1]
G. Vogiatzis[3]
[1] Technical University of Denmark [2] Istituto Italiano di Tecnologia [3] Aston University, UK

1. Introduction

Over the previous years, we have at the Section for Image Analysis and Computer Graphics at the Technical University of Denmark been working on generating high quality data sets for computer vision via our lab setup using a 6-axis industrial robot. This has provided a new data set aimed at feature matching [1, 4], and two data sets aimed at multiple view stereo [14, 16]. The resulting data sets are publicly available via `http://roboimagedata.compute.dtu.dk/`.

The evaluation of computer vision algorithms on these data sets has provided useful insights on realistic scenarios by setting a rigorous framework for evaluation. The results of these efforts have been well received by the community and the hardware and software platform associated with the robot is now well developed. We are currently in the process of making three new data sets aimed at 3D vision, with a special focus on the more challenging aspects, such as radiometry and the modelling of non-rigid objects. The construction of these data sets all leverage on our robotic setup's ability to produce ground truth camera and surface geometry, as briefly outlined in Section 2, and there is a great deal of commonality in the making of the data sets.

This abstract describes our current ongoing work on this data set construction for 3D vision. The data sets include:

1. A direct extension of our large multiple view stereo (MVS) data set [14], where we are now including transparent and semi transparent objects into the scenes, Section 3. A challenge in doing this is getting the ground truth geometry of the transparent objects.

2. A data set addressing the radiometric challenges in 3D vision as presented in Section 4 where we aim at extending our MVS data set by explicitly measure the bidirectional reflectance distribution function (BRDF) of the surfaces. This will have the additional feature to finally give a data set for evaluating photometric stereo with a ground truth.

3. An extension of our data set on feature matching to evaluate these algorithm with non-rigid objects, (Section 5) where we use actuators to make stop motion 3D data sets. This data set will also evaluate Non-rigid Structure from Motion (NRSfM) with realistic objects.

Figure 1. Photos of the 6-axis industrial robot mounted with two cameras and a projector. Cameras allow for MVS, and in conjunction with the projector SL provides ground truth point clouds.

2. Brief System Overview

Our experimental setup, cf. [1], is built around a 6-axis ABB IRB 1600 industrial robot, providing a flexible, precise, and highly repeatable camera pose. The robot is mounted with two Point Grey Grasshopper3 3376×2704 8-bit RGB cameras and a projector (for previously published datasets the cameras were 1600×1200 8-bit Point Grey Scorpion cameras). From each position ground truth surface point clouds are obtained using structured light (SL), and stereo images with a 32 cm baseline are captured with the camera pair. Five individually controlled 6500K LED tube lights allow for soft natural illumination of scenes from varying directions. Figure 1 shows the robot.

Previous evaluations of our system [14] have shown that the ground truth samples obtained through SL have good accuracy with a surface standard deviation of 0.14 mm. We expect similar or better performance in this data set. Positioning repeatability of the robot is very high, with a standard deviation of 0.0031 mm over two months.

Additional instruments used for generating the data include a CT (Computed Tomography) scanner for ground truth geometry of transparent objects (described in Section 3) and an illumination arch for controlled directional light-

Figure 2. Preliminary images from our data set. In the first row, three glass objects (sphere, bowl, teapot) with markers placed on. On the second row, three calibration and rendering tools part of the pipeline: a black and white checkerboard (coordinate estimation), an X-Rite ColorChecker® (color balance compensation) and a chrome sphere (environment light evaluation).

ing (described in Section 4).

3. Transparent Objects

Our goal is to extend our original MVS dataset to account for transparent objects where the focus is on reconstruction of geometry and appearance. Usually, the radiometric behavior of the objects used in 3D reconstructions is assumed diffuse and opaque. This leads to a number of simplifications that we cannot apply to transparent objects. In the case of transparent objects, refraction and reflection cause distortion effects that complicate reconstruction.

Previous methods acquire data sets useful for image-based rendering of a transparent object [18, 11]. However, these methods do not produce an actual triangle mesh and require special rendering techniques for reconstruction of the appearance of the transparent object. A survey on methods that do provide a triangle mesh is available [13]. In this survey, they note that CT scanning of refractive objects like glass is costly but straight forward. Thus, we use CT scanning to obtain ground truth geometry. Another way is to acquire shape and pose of a transparent object from motion [3]. In any case, there seems to be no data set, like the one we propose, which is useful for multiple view reconstruction of transparent objects.

3.1. Data

Our data set contains a set of multiple view HDR images of three glass objects with different radiometric properties (top row of Figure 2). We use a solid sphere, a bowl with lid (composed of two parts) and a teapot with multiple thin glass layers (composed of three parts). The walls of the bowl and the teapot have different thickness. A diffuse backdrop is provided for the objects. We have made this as a gradient checkerboard, so that one half of the squares varies in color from left to right, and the other half varies in color from top to bottom. In this way, we can see how light reflects, refracts and scatters through the objects. The refractive index of the glass objects will be estimated directly from the scanned images, or, if this is unsuccessful, by the use of a refractometer. We marked the objects with small black plastic spheres, in order to easily determine their position relative to the scene. In our data set, we also provide high-resolution triangle meshes generated from CT scans. We use these scans as ground truth for either geometrical reconstruction algorithms or physically based rendering algorithms for appearance modelling.

Our current data set creation procedure is as follows. First, we choose a sequence of camera positions and orientations for our industrial robot. The robot enables us to reproduce a given set of positions and orientations with a very high precision. Then, we capture a first set of images placing a black and white checkerboard in the scene. This is done to obtain the camera positions relative to the scanned objects and calculate camera parameters for the setup. Secondly, we scan a commercial color checker, which allows us to compensate for color channel alterations in the final images. Finally, we scan a chrome sphere to get an HDR environment map of the surroundings. We use the resulting map as a light source in our rendering algorithms [5], so we can simulate the resulting scene with high precision. After these three calibration steps, we can finally scan the glass objects using the same pre-defined path used for the calibration images.

Once compiled, we are planning to use this data set to verify that the radiometric models [9] properly describe the radiometric properties of the scene. To do this we plan to feed the ground truth of our data into a custom-built renderer based on the NVIDIA OptiX library [20], and see how well it reproduces the images. If successful, we have a validated computational model, which in principle we 'just' have to invert to do 3D reconstruction of transparent objects. Following this we plan at applying state of the art 3D reconstruction algorithms and quantify how far the state of the art has come toward solving this central 3D vision reconstruction problem.

4. BRDF measurements and Photometric Stereo

The radiometric behaviour of an object plays a crucial role in MVS. Often this behaviour has been ignored or at most assumed Lambertian. This allows for acceptable reconstructions of geometry, but often poor recovery of the reflectance. For more accurate MVS and reflectance capture, the BRDF of an object should be taken into account and this is a problem that receives a growing amount of

attention [24, 15]. Within the field of photometric stereo, the reflectance of an object is the key element in recovering surface normals and thereby indirectly the object's geometry. Also here, assumptions about reflectance are made, these include e.g. Lambertian behaviour [27] or isotropic BRDFs [12].

For both of the above areas, a multi-view data set having ground-truth reflectance behaviour would be of great value, and does, to our knowledge, not currently exist. We are therefore now working on a MVS data set where not only the ground-truth geometry is given, but also a densely sampled BRDF ground-truth for all materials in the scene. In the following, we will elaborate on the details of how this data set will be acquired and what it will include.

4.1. Concept

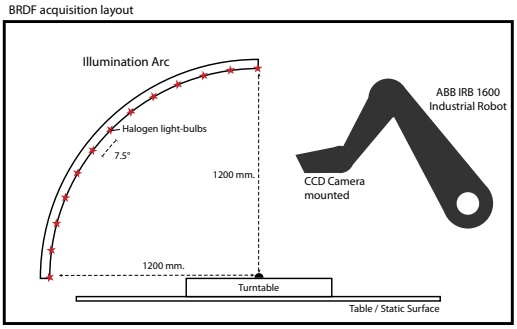


Figure 3. Schematic of BRDF capturing setup. Setup includes a 6-axis industrial robot holding a CCD (stereo) camera for view, and an arc in conjunction with a turntable for illumination.

Capturing the reflectance of a material generally requires four degrees of freedom: polar and azimuthal angle of illumination-direction, and polar and azimuthal angle of view-direction, $\rho(\omega_i, \phi_i, \omega_v, \phi_v)$. Utilizing our lab-facility's 6-axis industrial robot, mounted with a stereo-camera setup, all view directions (ω_v, ϕ_v) can effectively be captured. For illumination directions (ω_i, ϕ_i), we utilize an illumination arc and a rotation-table. The arc holds a range of halogen light-bulbs and is capable of covering the polar angle ϕ_i in 7.5° intervals. The rotation-table turns the target sample with a resolution of $< 1^\circ$, thus densely covering θ_i. Figure 3 shows a schematic of the BRDF capturing setup, and Figure 4 is a photo of an actual acquisition scene.

Using the above described setup, we intend to densely sample the BRDFs of a collection objects whose surfaces consist of one or a few, isotropic, BRDFs. The BRDFs of each material will be stored in the 3-dimensional Rusinkiewicz frame for isotropic BRDFs [21], as also done in the MERL database[17], although with a coarser resolution of 7.5° in each dimension. In conjunction with the densely sampled BRDFs, stereo images of scenes containing the sampled objects will be acquired for a wide range of directions. Objects will be of relatively low geometric complexity, and scenes will consist of one or more of the objects.

Figure 4. Capturing the BRDF of an object with known geometry. All illumination directions and view-directions are covered for each type material present on the object.

5. Non-Rigid Structure from Motion

Evaluating Non-rigid feature matching and NRSfM algorithms[1] in a quantitative manner has in the literature proven to be problematic. Deformations are inherently a dynamic process and subject to the physical properties of the objects in consideration. Thus, evaluating deformation modelling algorithms require a reasonable number of different objects and set of motions. Also, given the dynamic deformation objects might change their topology (e.g. stretching and tearing) and easily self-occluded some parts of the shape. For this reason, many approaches have provided several models that fit specific types of deformation, but that cannot comprise all of them. For this reason understanding the real performance of methods on realistic deformations is necessary to push forward advancements in this field.

The central problem of producing reference ground truth has been approached from many different angles. Several works compare their methods using synthetically generated images, as the true 3D geometry is readily available[26, 22, 19, 10]. Another popular approach is using MOCAP data, mainly human motion, for generating both test video sequence with 3D reference points [7, 26, 2, 10, 25]. Both falls short, as the former often lacks the complexity found in real-life scenes and the latter provides only a sparse set of reference points that are likely not to be possible to detect from images because of occlusions. As stated in [22, 8], there is a lack of and a need for a real-life NRSfM sequence with a dense 3D reference.

[1] A review on NRSfM methods, updated to 2010, can be found here: [23]

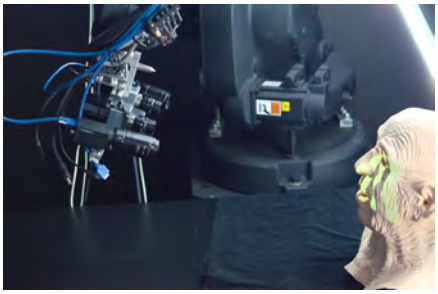


Figure 5. Robot arm carrying cameras for capturing stop motion frame and structured light data. A Gray code pattern is currently being projected onto the object.

We seek to remedy this situation by providing a video recording of real objects with dense 3D ground truth for each frame. It will be accomplished using a stop motion like animation techniques and structured light 3D scanning, combined in our unique recording setup.

5.1. Concept

We wish to simulate motion in a manner similar to stop motion animated films. Here a rigid object is moved into a certain pose, an image is taken, the object is slightly changed with a deformation, another image is taken etc. The result is a sequence that, when played at an interactive frame rate, provides the illusion of motion. We will apply the same principle here, in generating a benchmarking data set for NRSfM with ground truth.

Now one may ask, why not just record the motion using ordinary video format? After all, stop motion techniques does not properly reproduce motion blur artifacts that are present in standard recorded video sequences. Our approach has several significant advantages that greatly outweigh the loss of motion blur. Most importantly, we can obtain a 3D ground truth for each frame. After adjusting the object into its current frame position and acquiring an image for the stop motion sequence, we will perform a 3D scan using structured light. Utilizing gray code patterns we obtain a dense ground truth so obtaining both the image frame and a 3D reference for benchmarking and validation.

Another advantage is that we can obtain data from multiple views by acquiring images at different angles thus providing data for evaluating multi-view NRSfM (e.g. [6]). Furthermore, this procedure provides a great degree of control over both camera movement and object pose. As each frame is recorded independently, time in between becomes a non-issue.

Figure 6. Actuators for manipulating the geometry of the mask. The image of the mask has been superimposed on an image of the actuators, illustrating their functionality.

5.2. Implementation

Such data could be acquired by pure manual effort, however that would be extremely time consuming and error-prone. As such, a robotics solution is currently being developed with a the data acquisition procedure that is predictably and reproducibly implemented. In detail, a robotic arm move the camera and the projector needed for data acquisition and structured light scan. From this the view position can be determined with high precision and reproducibility. Figure 5 illustrates this setup.

Additionally, object deformation will also be automated and Figure 6 shows an example with an object where a mask resembling a human face is put on top of two actuators. Manipulating the actuators deforms the mask geometry, simulating facial movement. Similar results can be obtained with cloth, paper and other deformable materials.

6. Concluding Remarks

We have here presented our ongoing work on making high quality data sets for evaluating and developing methods for 3D vision. A motivation for doing this is that we see a need for this, especially with respect to making data sets that are large enough, so that it is possible to reasonably determine if differences in performance are a statistical fluke, or are in fact statistically significant.

By presenting our ongoing work in this forum, we hope to get valuable and constructive feedback on how these data sets in the making could be adapted to serve the needs of the computer vision communities as best possible.

References

[1] H. Aanæs, A. Dahl, and K. Steenstrup Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012.

[2] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1442–1456, 2011.

[3] M. Ben-Ezra and S. Nayar. What does motion reveal about transparency? In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1025–1032, 2003.

[4] A. Dahl, H. Aanæs, and K. Pedersen. Finding the best feature detector-descriptor combination. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 318–325, 2011.

[5] P. Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of ACM SIGGRAPH 98*, pages 189–198, 1998.

[6] A. Del Bue and L. Agapito. Stereo non-rigid factorization. *International Journal of Computer Vision*, 66(2):193–207, February 2006.

[7] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, volume 6314 of *Lecture Notes in Computer Science*, pages 297–310. Springer, 2010.

[8] K. Fragkiadaki, M. Salas, P. Arbelaez, and J. Malik. Grouping-based low-rank trajectory completion and 3D reconstruction. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 55–63. Curran Associates, Inc., 2014.

[9] A. S. Glassner. Surface physics for ray tracing. In A. S. Glassner, editor, *An Introduction to Ray Tracing*, chapter 4, pages 121–160. Academic Press Ltd., London, UK, 1989.

[10] P. F. U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 802–809. IEEE, 2011.

[11] T. Hawkins, P. Einarsson, and P. E. Debevec. A dual light stage. *Rendering Techniques 2005 (Proceedings of EGSR 2005)*, pages 91–98, 2005.

[12] M. Holroyd, J. Lawrence, G. Humphreys, and T. Zickler. A photometric approach for estimating normals and tangents. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, 27(5):133, 2008.

[13] I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. *Computer Graphics Forum*, 29(8):2400–2426, 2010.

[14] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 406–413, 2014.

[15] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo beyond Lambert. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I:171–178. IEEE, 2003.

[16] S. Kim, H. Aanæs, A. Dahl, K. Conradsen, R. Jensen, and S. Kim. Multiple view stereo by reflectance modeling. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012.

[17] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):759–769, 2003.

[18] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and rendering of transparent and refractive objects. pages 267–278, 2002.

[19] S. I. Olsen and A. Bartoli. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision*, 31(2-3):233–244, 2008.

[20] S. G. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, and M. Stich. OptiX: a general purpose ray tracing engine. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, 29(4):66:1–66:13, July 2010.

[21] S. Rusinkiewicz. A new change of variables for efficient BRDF representation. In *Rendering Techniques (Proceedings of EGWR 1998)*, June 1998.

[22] C. Russell, J. Fayad, and L. Agapito. Dense non-rigid structure from motion. In *Proceedings of 3DIMPVT 2012*, pages 509–516. IEEE, 2012.

[23] M. Salzmann and P. Fua. Deformable surface 3d reconstruction from monocular images. *Synthesis Lectures on Computer Vision*, 2(1):1–113, 2010.

[24] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 519–528. IEEE, 2006.

[25] L. Tao and B. J. Matuszewski. Non-rigid structure from motion with diffusion maps prior. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 1530–1537. IEEE, 2013.

[26] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 2761–2768, 2010.

[27] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):191139, 1980.

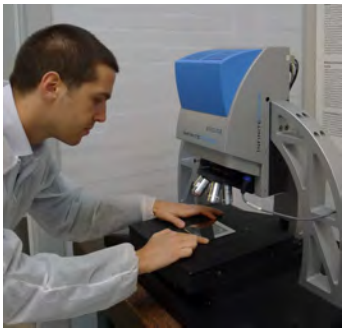
APPENDIX F

Contact area measurements on structured surfaces

euspen's S.I.G. Meeting:
Structured & Freeform Surfaces
9-10 November 2016
Copenhagen, DK

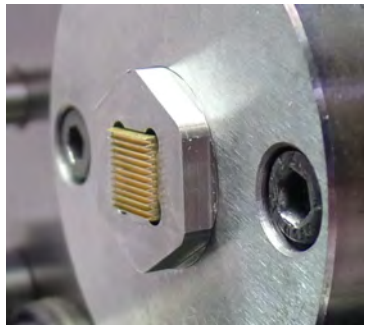
Contact area measurements on structured surfaces

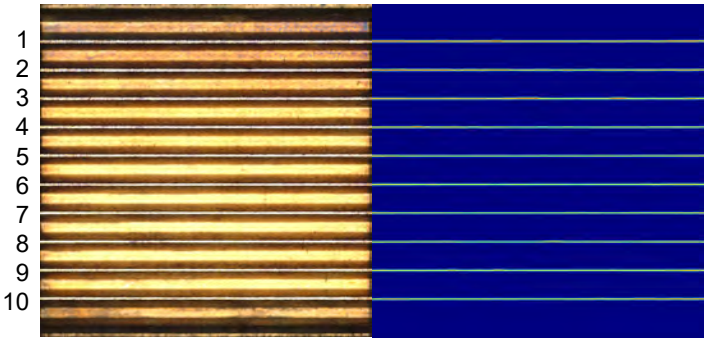
Ömer C. Kücükyildiz, Sebastian H. N. Jensen, Leonardo De Chiffre
Department of Mechanical Engineering and Department of Applied Mathematics and Computer Science, Technical University of Denmark



In connection with the use of brass specimens featuring structured surfaces in a tribology test, an algorithm was developed for automatic measurement of the contact area by optical means.

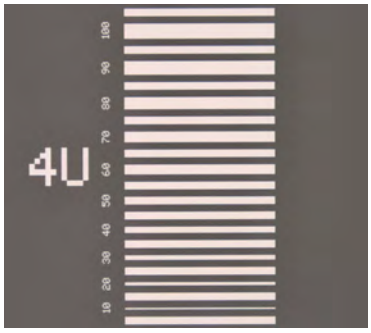
The contact area of the specimen after deformation is visible on a digital photograph as 10 parallel bands in adequate contrast to the background.

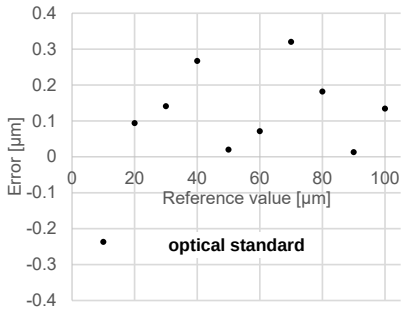


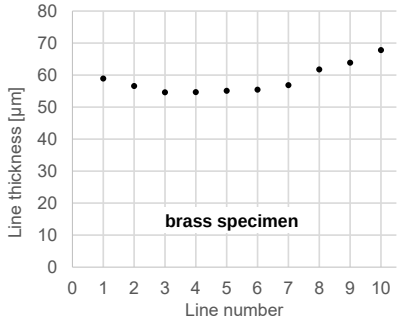


An approach was developed that automatically performs a pixel segmentation based on local image gradient extrema, leading to an accurate band-edge segmentation. For each band, a fine-grained line width is estimated through the distance transform in conjunction with non-max suppression, which can be used to estimate the desired area statistics.

During this study, the traceability of the method was established through an optical standard from NPL. Measuring line-widths in the range 10-100 µm, errors less than 0.4µm were obtained.







The measurement uncertainty for a single band is calculated as

$$U = k \cdot \sqrt{u^2{}_{ref} + u^2{}_{rep} + u^2{}_{p} + u^2{}_{e}}$$ where

u_{ref} : reference uncertainty
u_{rep} : uncertainty from repeatability of the measurements on the reference
u_{p} : specimen line-width uncertainty
u_{e} : uncertainty coming from the coefficient of thermal expansion

A general expanded uncertainty (k=2) of 0.5µm was estimated for single band measurement on the brass specimen.

The method was applied to quantify the single bands' width and it was observed on the specific item that the bands are slightly wider at the edges, indicating a higher deformation.

Based on the study, it is concluded that the method for automatic measurement of contact areas provides traceable measurements for the investigated dimensional range.

DTU Mechanical Engineering
Department of Mechanical Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

APPENDIX G

VirtualTable: a projection augmented reality game

VirtualTable: a projection augmented reality game

A. Dal Corso M. Olsen K. H. Steenstrup J. Wilm S. Jensen R. Paulsen E. Eiríksson
J. Nielsen J. R. Frisvad G. Einarsson H. M. Kjer

Department of Applied Mathematics and Computer Science, Technical University of Denmark

Figure 1: *(left) Our setup using projector (red frustum) and a Kinect camera (green frustum). (middle,right) Pictures of the gameplay.*

VirtualTable is a projection augmented reality installation where users are engaged in an interactive tower defense game.The installation runs continuously and is designed to attract people to a table, which the game is projected onto. Any number of players can join the game for an optional period of time. The goal is to prevent the virtual stylized soot balls, spawning on one side of the table, from reaching the cheese. To stop them, the players can place any kind of object on the table, that then will become part of the game. Depending on the object, it will become either a *wall*, an obstacle for the soot balls, or a *tower*, that eliminates them within a physical range. The number of enemies is dependent on the number of objects in the field, forcing the players to use strategy and collaboration and not the sheer number of objects to win the game.

Our installation is an example of a combination of tangible user interfaces [Shaer and Hornecker 2010] and projection augmented reality [Mine et al. 2012]. Leitner et al. [2008] presented IncreTable, a tabletop game that includes multiple inputs from different devices, including physical objects. Molla and Lepetit [2010] present a similar concept of augmented board game, but in their case the output is shown on a screen and not re-projected on the game. Compared to Leitner et al. [2008], our interaction design can be learned by exploration and thus requires no instructions. This is important in child-computer interaction and in combination with the use of tangibles it empowers the shift from "learning by being told" to "learning by doing" [Hourcade 2008]. We thus believe that our VirtualTable is an excellent concept for development of immersive and engaging learning games for children.

Our approach

VirtualTable uses a computer unit attached to both a Kinect camera and a projector. We first process the input from the Kinect depth camera, then we pass it to the actual game to display the output.

The objects are recognized using the depth camera of the Kinect. We automatically calibrate our software once before the game is actually started, to estimate both a ground depth and the Kinect-projector homography. After the calibration, objects of any significant depth (at least 0.5 cm) can be recognized. The output of the depth camera is used to create a bounding box around the objects. We exclude objects that are connected to the border, to avoid recognizing the players' hands. In Figure 1 (left) we see that the Kinect camera covers an area bigger than the game area, to not accidentally exclude objects lying on the border.

We transmit the identified bounding boxes to the actual game using a custom made protocol. In the virtual game, wall objects are invisible and affect only the behavior of the soot balls. We project a red glow around the towers to distinguish them and give a visual feedback on their range (see Figure 1 (right)). When we update the set of recognized boxes, we compare it with the existing set. Matching boxes have their position updated, interpolating it with their old position to avoid flickering. The remaining boxes are either added or removed to the game accordingly. Objects are distinguished only by shape: elongated objects are walls, square-like objects towers.

The behavior of the soot balls is simulated using Unity Engine's built-in navigation system on navigation meshes. A tower, with a given frequency, shoots bullets to the soot balls within its range, removing them from the game.

The game explores the concept of augmented reality games, combining the tangible sensation of the pieces from board games and the immediate visual feedback from modern computer games.

References

HOURCADE, J. P. 2008. Interaction design and children. *Found. Trends Hum.-Comput. Interact. 1*, 4 (Apr.), 277–392.

LEITNER, J., HALLER, M., YUN, K., WOO, W., SUGIMOTO, M., AND INAMI, M. 2008. IncreTable, a mixed reality tabletop game experience. In *Proceedings of Advances in Computer Entertainment Technology (ACE)*, 9–16.

MINE, M., VAN BAAR, J., GRUNDHOFER, A., ROSE, D., AND YANG, B. 2012. Projection-based augmented reality in Disney theme parks. *Computer 45*, 7 (July), 32–40.

MOLLA, E., AND LEPETIT, V. 2010. Augmented reality for board games. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

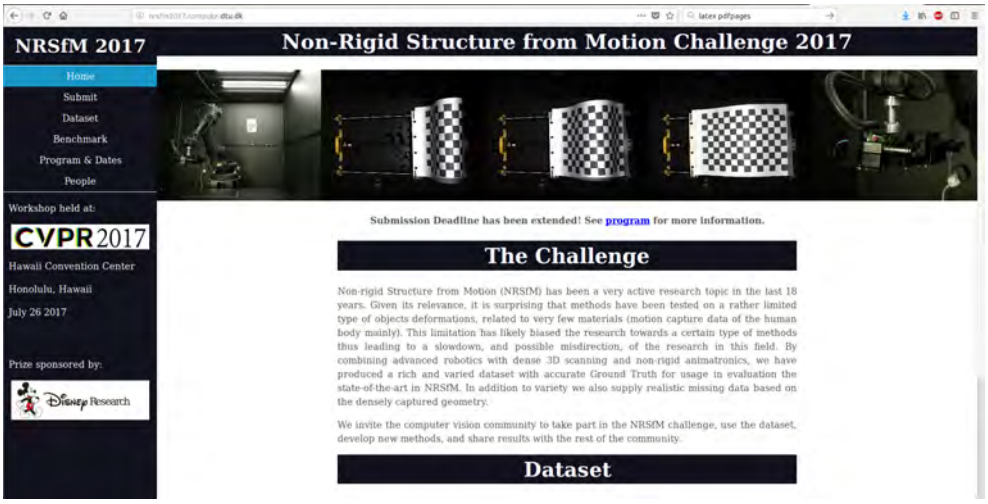
SHAER, O., AND HORNECKER, E. 2010. Tangible user interfaces: Past, present, and future directions. *Found. Trends Hum.-Comput. Interact. 3*, 1;2 (Jan.), 1–137.

SA'15 Posters, November 02-06, 2015, Kobe, Japan
ACM 978-1-4503-3926-1/15/11.
http://dx.doi.org/10.1145/2820926.2820950

APPENDIX H

Non-Rigid Structure from Motion Challenge 2017

This challenge was held at CVPR 2017. Description and results, as well as NRSfM dataset, can be found at `http://nrsfm2017.compute.dtu.dk/`



Bibliography

[Aan+15] Henrik Aanæs et al. “Our 3D Vision Data-Sets in the Making”. In: *The Future of Datasets in Vision 2015* (2015).

[Aan+16] Henrik Aanæs et al. “Large-Scale Data for Multiple-View Stereopsis”. In: *International Journal of Computer Vision* (2016), pages 1–16.

[AG16] Wim Abbeloos and Toon Goedemé. “Point Pair Feature based Object Detection for Random Bin Picking”. In: *Computer and Robot Vision (CRV), 2016 13th Conference on*. IEEE. 2016, pages 432–439.

[Akh+08] I. Akhter et al. “Nonrigid structure from motion in trajectory space”. In: *Neural Information Processing Systems (NIPS 2008)*. 2008.

[Bar+08] Adrien Bartoli et al. “Coarse-to-fine low-rank structure-from-motion”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pages 1–8.

[BHB00] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. “Recovering non-rigid 3D shape from image streams”. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. Volume 2. IEEE. 2000, pages 690–696.

[Bra+09] Sami S Brandt et al. “Uncalibrated non-rigid factorisation with automatic shape basis selection”. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pages 352–359.

[BU17] Min Bai and Raquel Urtasun. “Deep watershed transform for instance segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017, pages 2858–2866.

[Cae+17] S. Caelles et al. “One-Shot Video Object Segmentation”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Chh+17] Ajad Chhatkuli et al. “Inextensible Non-Rigid Structure-from-Motion by Second-Order Cone Programming”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[CMR11] Vincent Couture, Nicolas Martin, and Sebastien Roy. “Unstructured light scanning to overcome interreflections”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pages 1895–1902.

[CSL08] Tongbo Chen, Hans-Peter Seidel, and Hendrik PA Lensch. "Modulated phase-shifting for 3D scanning". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pages 1–8.

[CW14] Wen-Chung Chang and Chia-Hung Wu. "Automated Bin-Picking with Active Vision." In: *Key Engineering Materials* 625 (2014).

[Dal+15] Alessandro Dal Corso et al. "VirtualTable: a projection augmented reality game". In: *SIGGRAPH Asia 2015 Posters*. ACM. 2015, page 40.

[Del+12] Alessio Del Bue et al. "Bilinear modeling via augmented lagrange multipliers (balm)". In: *IEEE transactions on pattern analysis and machine intelligence* 34.8 (2012), pages 1496–1508.

[DGS17] Mohammad Dawud Ansari, Vladislav Golyanik, and Didier Stricker. "Scalable Dense Monocular Surface Reconstruction". In: *International Conference on 3DVision* (2017).

[Dua71] C Brown Duane. "Close-range camera calibration". In: *Photogramm. Eng* 37.8 (1971), pages 855–866.

[Eir+15] Eyþór R Eiríksson et al. "PRECISION AND ACCURACY PARAMETERS IN STRUCTURED LIGHT 3-D SCANNING." In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* 40 (2015).

[Ell+12] L-P Ellekilde et al. "Applying a learning framework for improving success rates in industrial bin picking". In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE. 2012, pages 1637–1643.

[GM11a] P. F. U. Gotardo and A. M. Martinez. "Computing Smooth Time-Trajectories for Camera and Deformable Shape in Structure from Motion with Occlusion". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.10 (2011), pages 2051–2065.

[GM11b] P. F. U. Gotardo and A. M. Martinez. "Kernel Non-Rigid Structure from Motion". In: *IEEE International Conference on Computer Vision*. 2011.

[GM11c] P. F. U. Gotardo and A. M. Martinez. "Non-Rigid Structure from Motion with Complementary Rank-3 Spaces". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011.

[GN12] Mohit Gupta and Shree K Nayar. "Micro phase shifting". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pages 813–820.

[HGM12] Onur C Hamsici, Paulo FU Gotardo, and Aleix M Martinez. "Learning spatially-smooth mappings in non-rigid structure from motion". In: Springer. 2012, pages 260–273.

[HV08] Richard Hartley and René Vidal. "Perspective nonrigid shape and motion recovery". In: *Computer Vision–ECCV 2008* (2008), pages 276–289.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.

[Its15] Itseez. *Open Source Computer Vision Library*. `https://github.com/itseez/opencv`. 2015.

[Jen+17] Sebastian Hoppe Nesgaard Jensen et al. *Non-Rigid Structure from Motion Challenge 2017*. 2017. URL: `http://nrsfm2017.compute.dtu.dk/` (visited on October 26, 2017).

[Jen+BD] Sebastian Hoppe Nesgaard Jensen et al. "A Benchmark and Evaluation of Non-Rigid Structure from Motion". In: *IEEE Transaction on Pattern Analysis and Machine Intelligence (Under Review)* (TBD).

[Jør+17] Troels Bo Jørgensen et al. "A Flexible Suction Based Grasp Tool and Associated Grasp Strategies for Handling Meat". In: *International Conference on Mechatronics and Robotics Engineering* (accepted 2017).

[Jør+BD] Troels Bo Jørgensen et al. "An Adaptive Robotic System for Doing Pick and Place Operations with Deformable Objects". In: *Robotics and Computer-Integrated Manufacturing (Under Review)* (TBD).

[JWA17] Sebastian Nesgaard Jensen, Jakob Wilm, and Henrik Aanæs. "An Error Analysis of Structured Light Scanning of Biological Tissue". In: *Scandinavian Conference on Image Analysis*. Springer. 2017, pages 135–145.

[KDL17a] Suryansh Kumar, Yuchao Dai, and Hongdong Li. "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion". In: *Pattern Recognition* (2017).

[KDL17b] Suryansh Kumar, Yuchao Dai, and Hongdong Li. "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion". In: *Pattern Recognition* (2017).

[Kim+16] Taewoo Kim et al. "A multiple kernel convolution score method for bin picking of plastic packed object". In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE. 2016, pages 4070–4076.

[KJD17] Ömer Can Kücükyildiz, Sebastian Hoppe Nesgaard Jensen, and Leonardo De Chiffre. "Contact area measurements on structured surfaces". In: *Euspen's SIG Meeting*. 2017.

[KL16] Chen Kong and Simon Lucey. "Prior-less compressible structure from motion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pages 4123–4131.

[Kon10] Kurt Konolige. "Projected texture stereo". In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. 2010, pages 148–155.

[KTZ05] M Pawan Kumar, PHS Ton, and Andrew Zisserman. "Obj cut". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Volume 1. IEEE. 2005, pages 18–25.

[LDA10] Xavier Lladó, Alessio Del Bue, and Lourdes Agapito. "Non-rigid metric reconstruction from perspective cameras". In: *Image and Vision Computing* 28.9 (2010), pages 1339–1353.

[Lia+15] Xiaodan Liang et al. "Proposal-free network for instance-level object segmentation". In: *arXiv preprint arXiv:1509.02636* (2015).

[MG17] Jeffrey Mahler and Ken Goldberg. "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences". In: *Conference on Robot Learning*. 2017, pages 515–524.

[OB08] Søren I Olsen and Adrien Bartoli. "Implicit non-rigid structure-from-motion with priors". In: *Journal of Mathematical Imaging and Vision* 31.2 (2008), pages 233–244.

[PA82] Jeffrey L Posdamer and MD Altschuler. "Surface measurement by space-encoded projected beam systems". In: *Computer graphics and image processing* 18.1 (1982), pages 1–17.

[Pal+09] Marco Paladini et al. "Factorization for non-rigid and articulated structure using metric projections". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pages 2898–2905.

[Pal+12] M. Paladini et al. "Optimal Metric Projections for Deformable and Articulated Structure-From-Motion". In: *International Journal of Computer Vision (IJCV)* 96 (2 2012), pages 252–276. ISSN: 0920-5691. DOI: 10.1007/s11263-011-0468-5. URL: http://dx.doi.org/10.1007/s11263-011-0468-5.

[PMC10] Brian L Price, Bryan Morse, and Scott Cohen. "Geodesic graph cut for interactive image segmentation". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pages 3161–3168.

[Qui+09] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software*. 2009.

[RC11] Radu Bogdan Rusu and Steve Cousins. "3D is here: Point Cloud Library (PCL)". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.

[RC17] Radu Bogdan Rusu and Steve Cousins. *Region growing segmentation*. 2017. URL: http://pointclouds.org/documentation/tutorials/region_growing_segmentation.php (visited on September 13, 2017).

[Rie+12] Hayko Riemenschneider et al. "Hough regions for joining instance localization and segmentation". In: *European Conference on Computer Vision*. Springer. 2012, pages 258–271.

[Ste+17] Jonathan Dyssel Stets et al. “Scene reassembly after multimodal digitization and pipeline evaluation using photorealistic rendering”. In: *Applied Optics* 56.27 (2017), pages 7679–7690.

[THB08] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. “Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors”. In: *IEEE transactions on pattern analysis and machine intelligence* 30.5 (2008), pages 878–892.

[TK92] C. Tomasi and T. Kanade. “Shape and Motion from Image Streams under Orthography: A Factorization Approach”. In: 9.2 (1992), pages 137–154.

[VL17] Paul Voigtlaender and Bastian Leibe. “Online adaptation of convolutional neural networks for video object segmentation”. In: *arXiv preprint arXiv:1706.09364* (2017).

[Wan+17] Zhe Wang et al. “Pose Estimation with Mismatching Region Detection in Robot Bin Picking”. In: *International Conference on Intelligent Robotics and Applications*. Springer. 2017, pages 36–47.

[WOL14] Jakob Wilm, Oline V Olesen, and Rasmus Larsen. “SLStudio: Open-source framework for real-time structured light”. In: *Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference on*. IEEE. 2014, pages 1–4.

[WTH07] Guanghui Wang, Hung-Tat Tsui, and Zhanyi Hu. “Structure and motion of nonrigid object under perspective projection”. In: *Pattern Recognition Letters* 28.4 (2007), pages 507–515.

[YFU12] Jian Yao, Sanja Fidler, and Raquel Urtasun. “Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pages 702–709.

[Zen+17] Andy Zeng et al. “Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge”. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pages 1386–1383.

[ZFU16] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. “Instance-level segmentation for autonomous driving with deep densely connected mrfs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pages 669–677.

[Zha00] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pages 1330–1334.

{2.7182818284}ϑφερτυθιοπσδφγηξκλ