Visual planning of humanoid robot locomotion considering world geometry and friction

February 2017

Martim CRISTINA DE SERPA BRANDAO

Visual planning of humanoid robot locomotion considering world geometry and friction

February 2017

Waseda University

Graduate School of Advanced Science and Engineering

Department of Integrative Bioscience
and Biomedical Engineering

Research on Biorobotics

Martim CRISTINA DE SERPA BRANDAO

Dissertation submitted for the degree of Ph.D. in Engineering.

Supervisor: Atsuo Takanishi

Members of the Committee:
Atsuo Takanishi
Mitsuo Umezu
Jun Ohya
Hiroshi Fujimoto
Hiroshi Ishikawa
José Alberto Rosado dos Santos-Victor

Abstract

Robots have the great potential of replacing or supporting humans at dangerous and straining tasks. For this to be possible, robust real-world locomotion is a crucial robot skill. For example at challenging scenarios such as disaster sites, robots have to be able to perceive, reason about and deal with obstacles, surface irregularities, slipperiness, etc. On top of this, robots might have to temporarily depend on batteries and as such energy expended on locomotion must be minimized. As we will review in this thesis, humans are known to be able to optimize locomotion energy, to prospectively plan their gait in challenging situations (think rubble, ice, rock climbing) and to greatly deal with perception uncertainty. Based on these observations, in this thesis we assume that insights from human gait and perception studies can inform planning and perception methods in robotics.

In particular, in this thesis we tackle the problem of obtaining low-electrical-energy, collision-free and slippage-free locomotion plans for biped humanoid robots. Our objective is to integrate not only collision but also energy and friction into planning algorithms. We tackle the problem at both planning and perception levels, using principles inspired by human gait and perception studies such as energy minimization, approximate gait models, uncertainty models, and hierarchical planning. Concretely, at the planning level we first propose footstep and whole-body-motion planners for biped humanoid robots which are applicable to flat, slanted and slippery terrain. A high-level A* footstep planner uses state-transition-cost and cost-to-go models based on simple principles and representations gathered from human gait literature which lead to energy efficient and human-like motion. Then, a lower-level whole-body planner uses the same cost functions and extra collision constraints to further optimize locomotion plans. At the perception level we propose algorithms for both geometry and friction estimation from vision. At first, we compare the performance of humans to that of different computer vision algorithms at the visual friction estimation task. We reach

important conclusions for robot autonomy, robot teleoperation and human perception of friction. Based on these results we develop an algorithm that estimates friction from material classes using semantic segmentation on Deep Neural Networks. We then estimate the uncertainty of friction predictions and incorporate it into the planning problem itself for robust plan feasibility. Regarding geometry estimation we study uncertainty measures of stereo matching and incorporate them into time-filtering methods for higher 3D reconstruction accuracy. The complete system is integrated and tested both in simulation and on the real robot WABIAN-2 using challenging scenarios with obstacles, slopes, and surfaces of varying friction.

Acknowledgements

I should thank many different people for their contribution to this thesis. First, my advisor, professor Atsuo Takanishi, for openly welcoming me in his lab, for his research advice and for making sure I had all the necessary conditions to conduct my research. Secondly, professor José Santos-Victor, who co-supervised the thesis, most importantly revising the stereo and planning chapters, and who has always encouraged me to go study abroad.

The help of Kenji Hashimoto was also crucial. On the logistics side, he helped with Japanese translations, grant submissions and more. Importantly as well, he helped define the experimental setup of the friction-from-vision datasets, and to revise the planning-related chapters. Ricardo Ferreira had a strong impact in this work, too. He suggested the idea of using stereo costs over the whole disparity to estimate occupancy, and endlessly helped with related paper revisions. We also shared many discussions related to the contents of this thesis. In addition to that, I should thank Hiroyuki Ishii, without whom I would not have had the privilege to participate on a JSPS program for research visits abroad. It allowed me to spend excellent research stays at Stanford, College de France and IST.

The list goes on. From Przemyslaw Kryczka I learned everything I know about the lab's robots, their hardware, electronics, control system, and importantly - how to repair each of them when I screwed up. My work with Lorenzo Jamone, on human-inspired reachable space maps and locomotion with internal models, ultimately lead me to approach the footstep planing problem the way I did (in a way, using human-inspired internal models). Matthieu Destephe shared many of his thoughts about my work, suggested readings, and gave advice on statistics, presentations and others, which also helped shape the thesis' idea. Tatsuhiro Kishi, Takuya Otani and Yukitoshi Minami preciously helped with robot experiments, robot maintenance, mechatronics insights and Japanese translations. Gabriele Trovato contributed to the improvement of the introduction of this thesis and oral

presentation. Finally, so many of Takanishi Lab's masters and bachelor students throughout my Ph.D. assisted with robot experiments and maintenance.

Throughout the course of the Ph.D. I was funded through different projects by the Japan Society for the Promotion of Science (JSPS) and the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT). Needless to say, their support was instrumental for the existence of this thesis (and my survival). Importantly as well, I would like to thank the thesis committee members for their time to review the thesis: Mitsuo Umezu, Jun Ohya, Hiroshi Ishikawa, and Hiroshi Fujimoto.

Last but not least, I should thank Anna Sharko for great many things. In the context of this thesis: for the Japanese translations of the friction-from-vision surveys, for participating in different experiments, and for helping with the collection of the GTF dataset.

Nomenclature

Abbreviations

ABP	-	Average Best Performing (parameter)
AGT	-	Average Ground-Truth (parameter)
AUC	-	Area Under the Curve
BTSAD	-	SAD with Birchfield and Tomasi's pixel dissimilarity
CCOG	-	Cost-Curve Occupancy Grid
CNN	-	Convolutional Neural Network
COM	-	Center of Mass
COP	-	Center of Pressure
DOF	-	Degree(s) of Freedom
FFT	-	Fast Fourier Transform
GTF	-	Ground-Truth coefficient of Friction (dataset)
HSM	-	Histogram Sensor Model
OSA+F	-	OpenSurfaces and Friction (dataset)
RCOF	-	Required Coefficient of Friction
SAD	-	Sum of Absolute Differences
SSD	-	Sum of Squared Differences
SQP	-	Sequential Quadratic Programming
ZMP	-	Zero Moment Point
c.d.f.	-	Cumulative distribution function
p.d.f.	-	Probability distribution function, probability density function

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Overview

Robots have been used in the industry for quite some time now, where they excel at repetitive and controlled tasks such as assembly-line manufacturing, sorting, warehouse transportation and others. Thanks to this maturity, there is now a growing interest in applying robots to service tasks in more general and unstructured environments as well.

Robots that can handle general environments would be useful for a number of applications:

a) **Disaster response**. Hazardous environments such as leaking nuclear power plants, mines or fires pose important challenges to response institutions. In the case of fires and leaking nuclear power plants, reconnaissance missions are needed to assess the damage or reduce risks for human workers, but even simple locomotion on the field is already life-threatening in itself. For example, the Great East Japan earthquake of 2011 lead to nuclear powerplant leaks, which made radiation levels too dangerous for human recon missions [1] thus greatly delaying containment. Fire fighting also involves dangerous recon missions, and is one of the most deadly jobs in the US [2]. *General* robot locomotion capabilities are important here because such environments are usually cluttered with obstacles and surfaces of varying properties and conditions. Throwable teleoperated recon robots have been developed for firesquads, disaster response teams or military to investigate a building's situation and risks [3]. Robots are also used in bomb disposal situations by the military [4].

In response to Japan's 2011 earthquake, several robots were sent to the Fukushima powerplant to measure radiation levels and check the state of the buildings and equipment [1]. Disaster response is currently a hot area of research for robotics in the academia, motivated by recent government funding, problem complexity and an interest in its societal value.

b) **Other professional uses**. Service robotics for professional use is a fast growing market, with worldwide yearly sales in 2015 increasing by 25% from 2014, to an estimated value of US$ 1.6 billion [5]. These include, among others, automated vehicles in manufacturing and other environments, robots for defense, agriculture, professional cleaning, inspection and maintenance.

c) **Personal and domestic use**. Examples of such applications include housekeeping [6], lawn-mowing and entertainment[7]. Personal and domestic robots' sale value was already worth US$1.2 billion in 2015 [5]. For such robots to be successfully deployable worldwide in any home, any backyard, or any event venue, however, they must be capable of dealing with general environments of varied geometry and dynamics.

For most of these applications *locomotion* is a crucial skill. How to efficiently get from one location to another, without damaging the environment or the robot itself, is a necessary skill whether the robot is used for cleaning around an untidy house, navigating an agricultural field among hills and plants and puddles, or rescuing a person from a disaster site. For these robots to actually be adopted by institutions they should work in general environments: any disaster sight, any home, etc. *This thesis is a step towards such general robot locomotion capabilities.* It introduces both perception and locomotion planning algorithms that provide robots with algorithms for safe, efficient and autonomous navigation of general environments. In particular in this thesis we consider environments with obstacles, slopes and varying surface friction.

The planning algorithms introduced here were designed for complex articulated robots, in particular legged humanoid robots. Humanoids are non-specialized robots which have an anthropomorphic body with arms, legs and head, see Figure 1.1 for examples. We opt to study the planning problem for such kind of robot in particular for two reasons. One is their complexity. Humanoids are legged articulated robots with especially complex dynamics

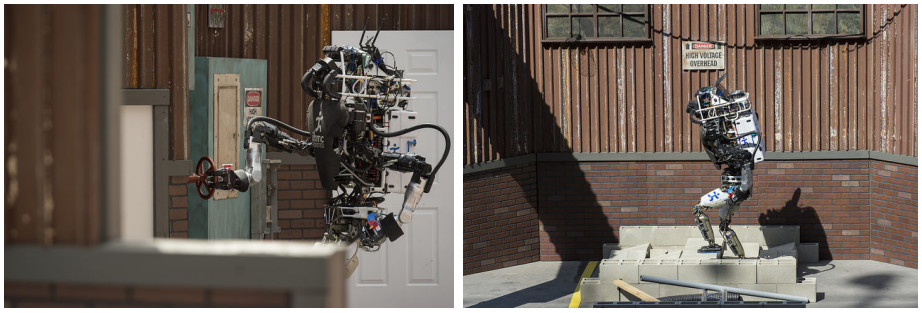


Fig. 1.1 Humanoid robots in urban environments.[1]

and constraints (e.g. contact, stability). This makes them a more general platform to study the planning problem than, for example, mobile robots. Since the algorithms introduced here have been thought with such complex robots in mind, they will be applicable also to other equally complex legged and articulated robots, as well as simpler mobile robots. Another important reason is that anthropomorphic robots have applications in all previously mentioned applications. They may even be preferred to less human-like robots in personal and domestic applications as higher anthropomorphism is often associated with higher familiarity and acceptability from users [8, 9], with the caveat that they should not look "too" human and thus eerie [10]. For both personal and professional applications, some roboticists also argue that anthropomorphism is advantageous for locomotion and manipulation in a human-inhabited world. The argument goes that since urban environments are made by humans and for humans, then human-sized humanoid robots should have special ease at dealing with them: for example with stairs, doorknobs, buttons, or any human-made tools which abound in urban and disaster response scenarios. By focusing on humanoid robotics, we may thus be able to tackle all problems solvable by humans, exactly when we cannot have humans do it.

The general planning problem is actually a hard problem. To exemplify the difficulty of the problem, take a look at Figure 1.2. For the robot to reach the example object for inspection and return in time for battery recharge, it should plan how to get there while minimizing energy consumption but at the same time avoiding slippage and other constraints. However, the ground is not entirely flat and there is a large ice puddle between the robot and

[1] "150605-N-PO203-329", By Office of Naval Research (https://www.flickr.com/photos/usnavyresearch/18529371672), CC BY 2.0.
"150606-N-PO203-565", By Office of Naval Research (https://www.flickr.com/photos/usnavyresearch/18602668501), CC BY 2.0.

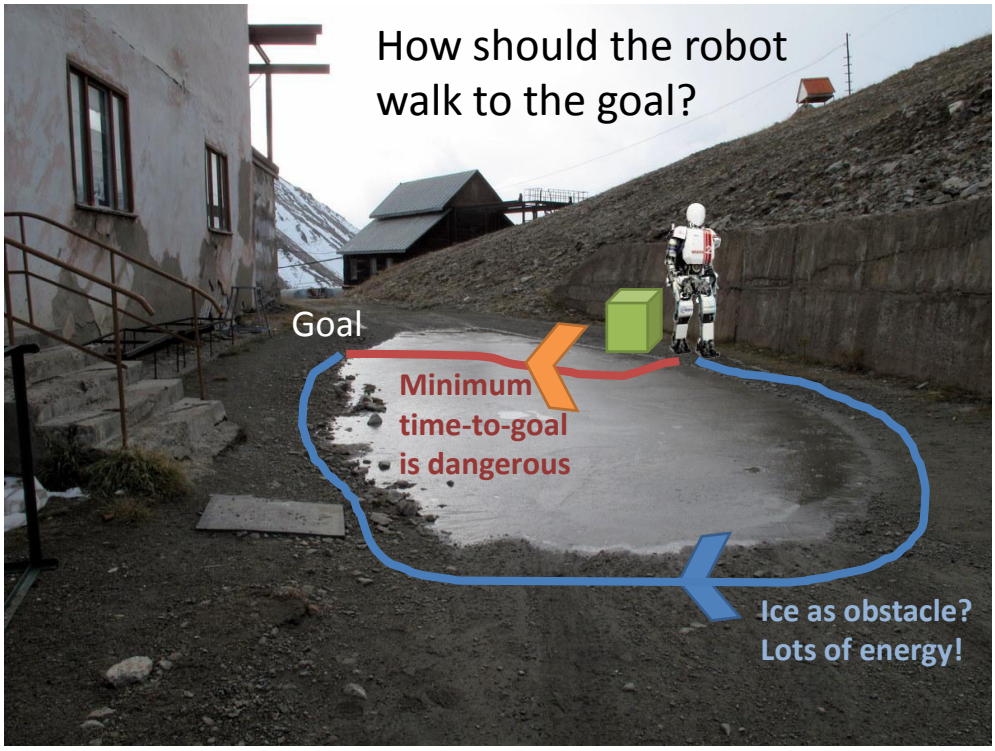


Fig. 1.2 The locomotion problem.[2]

the target. Should the robot walk around the ice? That would take large amount of energy because of the large distance. Should the robot cross the ice? That seems dangerous, but at the same time the distance is shorter, and crossing it very slowly could still be safe and spend less energy in the long run. We will deal with such kind of complex planning problems in Chapter 2. In addition to that, can a teleoperator faithfully estimate the friction of the different surfaces? What about computer vision algorithms? What is the uncertainty in the friction estimates obtained from visual input? Such questions will be tackled in Chapter 3. And finally, what about the estimation of geometry from vision, how should the robot estimate it reliably also taking into account the uncertainty in robot vision measurements? This will be our work in Chapter 4.

Current approaches to solve the legged and humanoid robot locomotion planning problem have strong shortcomings in several aspects:

a) **Robot energy consumption:** A common approach to plan motion of complex articulated robots is to do it hierarchically: to solve lower-

[2]Image adapted from "Kosmostantsiya ice puddle", By Peretz Partensky, available via Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Kosmostantsiya,_ice_puddle_(3992616316).jpg), CC BY-SA 2.0.

dimensional approximate versions of the problem before solving the final problem. In humanoids this usually consists of planning footsteps before the trajectory of the robot's joints. However, existing footstep planners discard energy consumption (and actually friction as well), thus constraining or biasing the final trajectories to highly sub-optimal regions of the space. If applied to our previous example in Figure 1.2, most current planners would opt for the least-distance path - which crosses the ice puddle - without knowing whether the trajectory is energy efficient or even feasible. Footstep planners are usually developed independently of full-body planners, optimizing different cost functions, and thus cannot guarantee that the generated low-cost footstep plans will also lead to low-cost full-body plans.

b) **Physical properties of the environment:** Current high-level planners such as footstep planners [11–14] discard physical properties of the environment. This can lead to unintended slips, falls or overly conservative motion. Friction is a especially important physical property because diverse surfaces with varying friction abound in the real world, from wood to ceramic tiles, grass or ice, which may cause difficulties or huge energy costs for robot locomotion if not considered. In addition to that, slipping in humanoid robots can most easily lead to falling, which can be costly or deadly (even though it is not as problematic in mobile robots).

c) **Friction estimation:** While some planning algorithms do consider environment friction at the lowest-level planning hierarchy [15–17], coefficients of friction are not actually estimated and in practice are set at some constant value by the user or algorithm designer. This could of course easily lead to large prediction errors and consequently sub-optimal stability and/or energy consumption. While friction prediction was outside of the scope of those papers, there is in general a lack of knowledge about how to go about friction prediction from a distance, for example using vision sensors.

d) **Perception uncertainty:** Some of the most popular 3D-reconstruction and mapping algorithms used in robot locomotion, such as occupancy grids [18, 19], do not actually incorporate uncertainty of the underlying sensors, particularly when using stereo vision. The few work that exists related to friction estimation [20] also does not provide uncertainty esti-

mates for its predictions. Such metrics of measurement uncertainty are crucial both for integrating information and reducing error over time, as well as to make locomotion robust to perception errors.

e) **Heavy teleoperation:** Current humanoid robot locomotion and manipulation skills are heavily teleoperated. For example, in the DARPA robotics challenge of 2015, where multiple teams teleoperated robots to do locomotion and manipulation tasks under communication faults, the disadvantages of lack of automation were exactly one of the concerns of most teams. Human teleoperation might not be possible at all times due to communication problems [1], or might be too stressful for the operators because of high responsibility and the overwhelming amount of information to process. Teleoperators can also make judgement mistakes of distances, and humans' optimistic biases in decision making [21] might lead to wrong decisions during locomotion planning. As another example, in the Fukushima powerplant mission of 2011 [1] the teleoperated robot eventually got stuck inside of the radioactive building because of its tough geometry and reliance on power cables (power cables were mandatory since wireless communication was not reliable and the design was focused on teleoperation).

The algorithms and analyses in this thesis address these shortcomings. They provide a way to achieve autonomous humanoid robot locomotion with comprehensive considerations, both in terms of environment properties – complex terrain geometry, obstacles, slippery surfaces - and robot constraints as well, such as stability, collision, battery consumption. As we will see, they do so thanks to a new friction-perception algorithm (Chapter 3) and a tighter integration of planning levels (Chapter 2 and 5). Our planning algorithms extend the amount of factors considered at high-level (footstep) planning levels, leading to safer and more energy efficient motion. We will also analyse the uncertainty in friction and geometry perception, in order to improve both reliability in perception, and locomotion safety.

Before stating our problem and contributions more technically, let us introduce related research in the fields of robot locomotion and robot vision.

1.2 Related research

1.2.1 Robot locomotion

Research on robot locomotion usually distinguishes between two levels of locomotion control: motion planning and control. Motion planning is the problem that we are concerned with in this thesis, and consists of deciding a sequence of states the robot should go through in order for it to arrive at a distant target. It usually involves reasoning about obstacles and high-level waypoints on the way to the target. The reader should refer to [22, 23] for an overview of motion planning algorithms. Control, on the other hand, deals with monitoring and locally adjusting the state of the robot at each point in time to guarantee that the plan (and possible extra constraints such as local stability) is satisfied.

The motion planning problem on humanoid robots is especially challenging since they are underactuated and move by establishing and breaking contacts with the environment. So when a humanoid walks 20cm forward, there are infinitely many trajectories for its joints that satisfy this motion constraint (infinitely many posture sequences of the body), but at the same time the feet should make contact with the environment, which limits the postures to a narrow and complex constraint of feet in manifolds. Because of this complexity of the search-space of humanoid robot locomotion, the problem has been approached with different methodologies that try to reduce the complexity in one way or the other:

a) Contact before motion

In this hierarchical approach, contacts such as footholds or hand positions are planned first while ignoring the complexity of the whole-body. Kinematics constraints such as joint angle limits are usually approximated by distances between contacts [13], and collision by approximate bounding boxes of the whole body [14]. Different methods have been applied to generate contact plans, such as graph search on a discretized state-space [11–13, 24–26] or numerical optimization on environments represented as a set of planes [14]. Such contact-before-motion methods are fast and allow for a simpler subsequent problem of finding full-body motion connecting the footholds. That full-body motion can be planned using for example sampling methods

[26] or numerical optimization [15] once again.

b) Motion before contact

Here the motion of the full-body or of a reference point such as the center-of-mass (COM) is planned first without considering contact, and then the motion is adapted or footholds searched in order to satisfy contact constraints. For example, [27] plans COM motion and then footholds and limb motion that satisfy it. Depending on the implementation, planning with motion primitives [28] can also follow a *motion before contact* approach, since full-body motions saved in a library are searched and composed until a path to the goal is found even if they don't establish contact with the environment at first. Then, contact constraints are considered and the motion is locally adapted until contact and other constraints are fulfilled.

c) Contact and motion

In this approach, popularized in the animation field [29], contact and full-body motion are planned jointly by local adjustment of an initially unfeasible trajectory until contact is made and other constraints satisfied. Used techniques include numerical optimization with continuation [29] or linear complementary problem formulations [30]. Such methods are prone to local minima and thus still require another, simplified, planner to initialize them with close-to-feasible trajectories. They are also computationally heavy and might require minutes to hours until convergence.

1.2.2 World representations and vision for locomotion

Most of the aforementioned motion planning methods have been tested in simulated environments where the geometry is known, usually modeled as 3D polygon meshes [26, 29]. Some of the humanoid planning research conducted on real robots and real environments has also ignored the robot vision problem by using motion capture systems placed across the whole environment that can sense it entirely with high-precision [24], even parts of the environment not under the robot's field-of-view. In real environments outside the laboratory, however, only the robot's sensors are available to the

robot, or at best a group of robots' sensors, and so these measurements should be used to reconstruct the environment geometry.

The study of 3D reconstruction includes the development of sensors and algorithms for distance estimation, such as stereo vision with cameras, active pattern projection with RGB and depth (RGBD) cameras, or rotating laser-rangefinders. It also includes the accumulation of this information over time with filtering methods such as Bayesian filters [18, 31], and the *active vision* problem of selecting where the robot should "look at" in order to improve localization performance [32, 33], map completeness [34–37], or performance of the navigation task [38–41].

Humanoid robots are most usually equipped with two cameras in the head to mimic human's stereo vision. Similarly to animal's stereoscopic vision, stereo in computer vision consists of matching pixels on one camera's image to pixels on the other. This is done by computing the photometric difference between the two pixels, or a neighborhood of the two pixels, using cost functions such as sums of absolute differences (SAD) of pixel color intensities [42]. Distance of each pixel from the camera can then be recovered from the parallax of the matched pixels. Refer to [43] for an overview of different approaches to and methods used for the stereo vision problem. Probabilistic metrics of stereo matching are also used in some of the literature, called stereo confidence measures [44, 45] – and we will discuss them more deeply in Chapter 4. These try to model the probability distribution of distance for each pixel, usually based on statistical models of the measured pixel-matching cost functions.

Recent methods of simultaneous localization and mapping using monocular and stereo cameras have been particularly successful at reconstructing the geometry of large environments such as whole neighborhoods or the indoors of whole buildings [46, 47], as well as extremely fine-resolution reconstruction of whole rooms [48]. Recent progress has been made possible partly by fast and large-scale computational power of GPUs, combined with good methods for loop closure (recognizing previously seen locations in order to detect and adjust errors in reconstructed maps). These technologies have also been used in stereo-equipped humanoid robots such as Atlas [49] for 3D reconstruction of the environment before locomotion planning [15].

In general, stereo vision [49, 50] and the (non human-inspired) laser rangefinders [51] are popular sensors used for the reconstruction problem in humanoids. Stereo vision's advantage is its speed (when computed on-

board the sensor), but rotating laser rangefinders have been preferred due to the larger field of view and lighting-independent performance. Their main disadvantage is the long time that is required to acquire a complete scene, since each measurement is only on a plane and so the robot should wait for the laser to rotate 360 degrees before getting a snapshot of the environment.

The output of stereo, as well as laser or RGBD data, is point clouds – which are not efficient representations for robot locomotion planning. Due to the high computational cost of estimating collision between the robot's parts and the environment, several methods have been developed to speed it up using alternative environment representations such as height maps [52], octree-organized grids [53], signed distance functions [54], ellipsoids [55], convex decompositions [55, 56], planar segmentation [49, 52] and meshes [57].

Probably since the motion planning problem is already hard enough when only collision is considered, friction considerations have been mostly absent from the humanoid and legged robot literature except for some simulation-tested methods [26]. Most planning methods consider collision only, and leave stability and friction constraints for the subsequent problem of control through reflexive [58] and optimization-based controllers [15, 16]. Friction has been considered in planning more frequently in space applications using rovers [20]. The friction-from-vision problem in this case consists of predicting wheel slip (i.e. lack of locomotion progression). For example [20] uses visual features to classify terrain classes and machine learning techniques to prediction slip from these classes and rover pose.

1.2.3 Visual scene understanding

The problem of visual estimation of friction and other physical properties is highly related to the problems of visual recognition studied in "scene understanding" – which is concerned with classifying scenes and their different regions and objects, inferring spatial relations, among others. For example, friction is usually mostly associated with material in occupational accidents research [59], as well as in engineering in general [60, 61]. Many computer vision methods in scene understanding have been proposed for estimating materials [62, 63], objects [64], places [65] and others. Most methods consist of developing highly predictive visual features (e.g. color, texture [66], neural representations [62–64]) and using machine learning techniques (e.g.

SVMs, neural network back propagation) to learn a classifier from features to material/object/place class. Recently, deep neural network architectures have surged as the highest performing methods in most datasets, due to their large number of parameters and fast parallelization schemes made possible with GPUs.

1.2.4 Human-inspired algorithms

Another field of research with which this thesis is concerned is that of human-inspired algorithms. One of the possible motivations for their use is the following: when humans are better than machines at a certain task to be automated by algorithms, then it makes sense to try to use similar principles or processes to that shown by human behavior (or physiology) in order to improve those algorithms. In the example of the games of Chess [67] and Go [68], before computers surpassed human player performance much of the efforts went into encoding human heuristics or automatically learning from human games. Of course, the use of human-inspired algorithms does not necessarily mean human-level performance to be the final goal. Using the same example, the AlphaGo algorithm surpassed human performance by using a clever combination of learning from human games using neural networks, together with self-practice using reinforcement learning techniques [68].

In computer vision, the use of deep neural networks for visual tasks has been partly motivated by findings of the organization of the human brain's visual cortex [69]. Several well-performing visual features for mapping and object recognition applications are also inspired by principles in human perception [70, 71]. In numerical optimization, a large area of research is also focused on improving optimization algorithms by using principles observed in animals such as bees, flocks of birds and ants [72], some of which have been shown to work similarly to decision making processes in the human brain [73, 74]. These algorithms are among the best performing global optimization algorithms, together with other (evolution and natural selection-inspired) genetic algorithms.

In robotics, human-inspired visual servo control has been used to simplify the locomotion problem, for example in [75, 76]. In particular, [75] uses the fact that the task of reaching for a visual target in the absence of obstacles can be simplified to a simple control of direction using eye-hand and head-

walking-direction couplings which are studied in human reaching [77, 78] and locomotion tasks [79, 80].

Particularly in humanoid robotics, as we focus on in this thesis, it makes sense to consider human-inspired algorithms for vision and locomotion since both the visual system and body morphology are similar to humans'. Humans excel at visual and locomotion tasks: they are known to optimize locomotion energy [81–84], they plan their gait in challenging and varied situations from simple flat locomotion to rock climbing.

Different principles of human gait have been applied to humanoid robot planning. For example the regularities in simpler gait spaces of step lengths and step widths [11], the use of heel-contact and toe-off phases of gait [85], and the optimization of jerk [86] or energy [11].

1.3 Objectives and contributions

In a nutshell, the goal of this thesis is to try to answer the following question:

> *How can robots, in particular humanoids, autonomously navigate general environments with slopes, obstacles and slippery surfaces – in a way that considers energy consumption, friction, collision, stability and uncertainty in measurements?*

We are concerned with both planning and perception aspects of this broad goal, and we organize it into several smaller objectives:

a) To develop algorithms for planning full-body locomotion trajectories that consider world geometry and friction, as well as robot energy consumption, slippage and stability;

b) To investigate the usefulness of applying principles and representations of human gait into humanoid locomotion planners;

c) To develop algorithms for estimating friction and geometry from vision;

d) To understand human's visual perception of friction and whether it is reliable for teleoperation;

e) To integrate the algorithms into a single system and evaluate it in real-world scenarios.

The thesis has two important contributions to the robot motion planning field of research. One of them is a new algorithm for footstep planning that achieves low electrical energy consumption and low slippage risk by using human-inspired gait models. The other is a robust and objective-consistent hierarchical planning method which considers trajectory costs, collision, stability, friction and friction measurement uncertainty.

To the research field of robot vision, the thesis also contributes with a visual friction estimation algorithm which provides uncertainty estimates for robust planning. It further introduces an innovative approach to friction estimation of previously unseen materials using text mining; and a geometry estimation algorithm which accumulates stereo and its uncertainty over time for high precision and visual robustness. Finally, it provides some interesting and important conclusions regarding the dangers of assigning the friction perception task to a human teleoperator.

The innovative aspects of this thesis, from the point of view of considered environmental and robot factors in contact planning, can be seen in the comparison of Table 1.1. The comparison ignores the context and subtleties of each system, but it shows how this thesis 1) simultaneously includes friction, speed and dynamic stability considerations at the contact planning level whereas state-of-the-art planners do not; 2) actually predicts surface friction from sensors whereas state-of-the-art systems do not. The latter is due mainly to a strong research focus of the robotics community on the planning and control sides rather than perception, as well as a reliance on experimental validations in simulation where coefficients of friction are known with certainty (i.e. robot perception is unnecessary).

1.4 Outline

The structure of the thesis is shown in Figure 1.3. It is organized as follows.

In Chapter 1, “Introduction”, we introduce the background and motivation for this research, as well as nomenclature and related work.

In Chapter 2, “Humanoid locomotion planning considering world geometry and friction”, we start off by reviewing the human gait literature which we find relevant to our planning problem. Concretely we show that human gait is planned, and we identify planning variables and objectives. Based on these insights we propose a new extended footstep planning algorithm and

Table 1.1 Comparison between state-of-the-art contact planners and this thesis

		[13] [14]	[12]	[26]	[15]	This thesis
Environment considerations in planner	Geometry	○	○	○	○	○
	Friction	×	×	△	△	○
	Other physical properties	×	×	×	×	×
Robot considerations in planner	Energy	×	○	×	△	○
	Speed	×	○	×	△	○
	Collision	○	×	○	○	○
	Slippage	×	×	△	△	○
	Joint limits	×	○	△	△	○
	Dynamic stability	×	×	×	△	○
Robot perception	Geometry	×	×	×	○	○
	Friction	×	×	×	×	○

Note: △ indicates the factors are considered at the full-body motion planning level (not footstep/contact planning).

evaluate it in different simulated scenarios. We also discuss certain aspects in which the obtained robot motion is human like. We conclude with an extension to full-body motion planning on the same objectives which makes it tightly integrated with the footstep planner.

While the environments considered in Chapter 2 are assumed to be completely known by the planner, clearly the assumption does not apply on a real robot. The following two chapters then focus on estimating environment properties from sensors and characterizing their uncertainty. Both start from images as input but they independently estimate friction (Chapter 3) and geometry (Chapter 4). The two chapters are independent from each other and can be read in any order. This is represented in the thesis flowchart of Figure 1.3 as a parallel organization. In particular, in Chapter 3, "Visual perception of friction", we analyze both human and computer vision performance at the friction from vision task. Using especially designed datasets, we start by asking the question of which visual features humans use to estimate friction and how useful their predictions are for robot teleoperation. Based on the results we then propose an algorithm to estimate friction and its uncertainty pixel-wise from images. In Chapter 4, "Visual perception of

geometry", we deal mostly with uncertainty modeling in stereo vision. We benchmark different models of stereo matching confidence and propose a new "histogram sensor model" which leads to high reconstruction performance. Finally we propose a method to integrate these models into a time-filtering algorithm: occupancy grids.

Then, in Chapter 5, "Vision-based hierarchical planning in the real world", we show results of the fully integrated system in a real biped humanoid robot. We describe the scenario we built on the laboratory with varying degrees of friction; we discuss software architecture and implementation details used for computational speed; and we show results of the robot navigating such scenarios with visually estimated friction and geometry. After proving the capabilities of the real system on challenging scenarios in the laboratory, we further show simulation results in a collection of real outdoor scenarios acquired with a 3D camera.

Finally, in Chapter 6, "Conclusion and discussion", we summarize the achievements of this thesis, and discuss limitations, open questions and future directions of research.

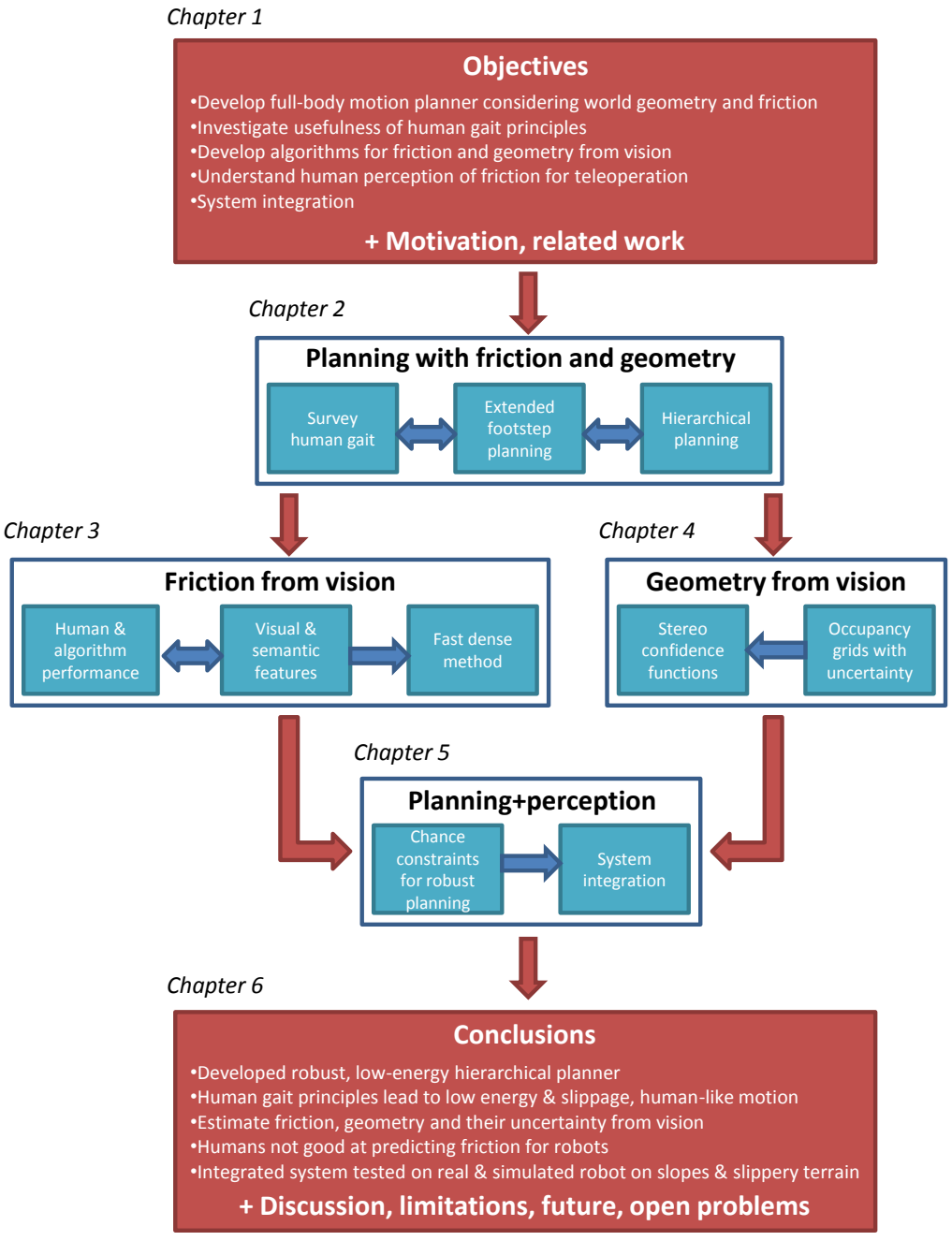


Fig. 1.3 Thesis outline

Chapter 2

Humanoid locomotion planning considering world geometry and friction

2.1 Introduction

In Chapter 1 we discussed how humanoid robot locomotion planning is an important problem with applications in disaster response and service. Current planners excel at obstacle avoidance, but do not consider important factors such as ground friction and energy consumption. These are especially important in outdoor environments where the robot will depend on batteries and surface conditions might be challenging: slippery, inclined, etc. While it is still not clear how planners should be formulated in order to consider many of such factors, one of our claims in this chapter is that using principles and representations in human gait literature can lead to natural improvements of humanoid locomotion planners. We will also look at how to alleviate the complexity of the full-body locomotion planning problem when considering world geometry and friction.

Concretely, the objectives of this chapter are the following:

a) To show that principles and representations in human gait are applicable to footstep planning of humanoids

b) To check whether human-inspired optimization principles in footstep planning lead to practical energetic efficiency improvements in humanoids

c) To check whether a human-inspired footstep planner leads to human-like walking behavior

d) To propose a method to integrate footstep planning and full-body planning in order to consider world geometry and friction, as well as full-body kinematics and stability.

To this end, in Section 2.2 we will give an overview of anticipatory human gait literature and identify principles and representations useful to humanoid locomotion in a variety of scenarios. Then in Section 2.4 we will propose a footstep planning algorithm based on those principles and representations which plans both footstep positions, orientations, timing and parameterized COM motion. Finally in Section 2.5 we will propose a hierarchical planning algorithm for full-body planning. Whenever appropriate, we will show how the results obtained in our humanoid robot match observations in human walking behavior.

2.2 Background

2.2.1 Humanoid footstep planning

Footstep planning algorithms are a computationally attractive solution to the humanoid locomotion planning problem since they reduce the search space from whole-body motion to footstep positions and orientations. Current footstep planners excel at obstacle avoidance, but do not consider important factors such as ground friction and energy consumption. These are especially important in outdoor environments where the robot will depend on batteries and surface conditions might be challenging: slippery, inclined, etc.

The footstep planning problem is closely related to the study of anticipatory human gait adaptations. For example, representations of walking used in human gait literature to describe anticipatory gait control are closely related to those used in high-level motion planning algorithms in robotics, such as footstep planning, contact planning and other task-space planning approaches. Both typically deal with observations in terms of a high-level representation of walking such as modality, foot position, orientation, timing, limb stiffness or center-of-mass (COM) height.

For humanoids, the footstep and contact planning problems have been tackled with search [11–13, 24, 25], sampling [26] and optimization [14, 87] algorithms. Search-based planners such as A* [11, 12, 24] and its variants [13, 25] have been used successfully to plan obstacle free paths in both static and dynamic [24] scenarios. Recently, purely optimization-based planners have also been proposed [14], which eliminate the sub-optimal discretization problem inherent to search-based planners. Sampling-based [26] planners allowing for multiple contacts (e.g. hands, knees) are useful for very complex environments, although at a high computational cost, which can be slightly ameliorated with a good selection and adaptation of motion primitives [28]. While the aforementioned planners focus on finding collision-free paths, in this thesis we go one step further: considering energy, collision and friction.

One important step in footstep planners is to estimate whether a given stance or step is feasible or not. Some authors opt to approximate feasibility by rough reachability of the feet [13], full inverse kinematics feasibility [26], or smart collision checking [88]. In this chapter we use both rough reachability intervals to discard obvious unfeasible poses, as in [13], but also learn a model of feasibility from physics simulation: where feasibility is both static and dynamic.

Research closely related to the method we introduce in this chapter includes [11], in which terrain and energy-related cost functions are used in A* search to compute optimal cost plans. They sum a set of empirical human biomechanics-inspired models of energy cost that are polynomial functions of step length, width and rotation. Also [12] uses a similar approach, with quadratic cost functions on sequences of footstep positions. On the other hand, in this thesis we consider also timing variables and surface friction. We do not assume polynomial relationships and instead use an off-the-shelf machine learning algorithm to learn the relationship between variables from data. And finally we make claims concerning energy consumption and human-like motion.

In this thesis we prevent slippage of the robot by planning, which complements other feedback control approaches to friction-constrained biped walking [58, 89, 90]. While feedback control can help reduce tangential-to-normal force ratios locally, it may not be sufficient in very low friction surfaces. For example a robot with rubber soles would be subjected to less than 0.15 kinetic friction when walking on ice. Slipping can be reduced in such low friction floors without changing gait, but not eliminated [89].

Feedback control approaches usually consist of friction cone constraints in inverse dynamics [17] or operational space control framework [90]. Design parameters in the preview controller [91] can also be slightly tuned to reduce the Required Coefficient of Friction (RCOF) for a fixed gait, and feedback ZMP controllers manually adapted to account for friction [89]. Efforts have also been put into reactive reflex controllers that, without changing gait parameters, try to reduce slipping after it is detected (e.g. by waist or foot acceleration reflexes [58]). In this thesis we take the complementary high-level approach, by optimizing energy and eliminating slippage as much as possible by changes in gait. Such approach solves the known problem of reactive controllers to not be able to avoid slipping on fast gait [58], and at the same time leverages on human gait literature findings supporting energy and stability optimization at the footstep level, which is not just reactively but also anticipatorily controlled.

Such a planning approach to the friction problem is closely related to algorithms that try to decrease the risk of slipping even in low friction conditions. For instance, [92] proposes a method for grasp synthesis prioritizing low "friction sensitivity", such as to prefer grasp configurations that are stable even for low COF. Similarly in the biped locomotion literature, [89] changes parameters regulating center-of-mass motion such that the minimum COF where the robot can walk without slipping is decreased.

2.2.2 Humanoid full-body motion planning

Several approaches exist to the friction-constrained motion planning problem for legged robots. One approach is the non-hierarchical, full-scale trajectory optimization formulation with implicit contact constraints of [29, 30]. While technically elegant and showing promising results, these can still be computationally expensive for online planning. In order to make the problem tractable, full-body motion can instead be planned after contact (or footstep) planning [14, 15, 26, 93], in what is called the *contact before motion* approach which we already introduced in Section 1.2.

Whether contact constraints are given by a footstep planner or implicitly defined in the full-body motion planning problem, numerical optimization has recently proven to be an effective approach to the problem [15–17, 29, 30]. Such an approach defines an optimization problem where variables are the body's joint angles or torques at several waypoints or collocation points.

Then, constraint functions are designed such as to respect joint angle limits, actuation limits, contact constraints, and possibly even stability, full-body or centroidal dynamics, and collision. Collision is usually represented as penetration (signed-distance) constraints [57] and computed using libraries for rigid body dynamics such as the Open Dynamics Engine [94] or Bullet [95]. Friction constraints can also be added to these planning formulations, as linear constraints on the contact forces.

2.2.3 Anticipatory gait control in humans

a) Human gait is planned

The claim that humans also plan gait, and footsteps in particular, is supported by several evidence in both children and adults. For example, children walkers (average 14 months) switch walking modality from bipedal to quadrupedal on a waterbed after visual inspection of its waviness or haptic exploration [96]. Children also use haptic exploration on slopes to decide whether to walk, crawl, slide down in sitting or backing positions or not traverse them at all [97].

Across numerous studies of adult human walking there is also the observation of a "cautious gait" style used in uncertain environments [98–100] or after sensory loss [101, 102]. For example [98–100] observed a specific cautious gait mode when there is awareness of a slippery surface, which is then adapted to the specific slipperiness condition found. Typically on slippery surfaces, walking speed is decreased, the COM is centered over the supporting limb and limb stiffness is increased [98–100]. Even when there is no knowledge of the degree of slipperiness, stride length [98, 99, 103], foot contact angle [100, 103–105] and vertical heel contact velocity [104] decrease, while knee flexion increases [100, 105]. According to [100], these surface-approach changes are learned over prior slip experience and are applied to different conditions when surface properties are unknown. Further knowledge of the coefficient of friction changes muscle activation and how the foot interacts with the floor. On slippery terrain, both these gait and muscle activation patterns become characteristically different since the first step on the surface, which indicates an anticipation strategy and not reactive adaptation of normal gait. A cautious gait is also used in other uncertain circumstances such as when vision is blurred by light scattering lenses

[101]. Another interesting observation is that human walking trajectories on steep slopes such as mountains or hills are not straight least-distance paths but more energy-efficient curved paths uphill [106, 107]. Interestingly, [108] showed that visual perception of slant changes from viewpoint (downhill looks steeper and is also more difficult), which suggests that climbing gradients could be a result of perception of slant.

All these examples show how humans adapt high-level gait parameters such as modality, footstep position and timing or COM trajectories by some sort of motion planning based on visual or haptic perception of the environment.

Part of these observations have been obtained in robotics and animation literature by optimization algorithms, for example lower step lengths and lower COM [109] on slippery terrain. In [109] this was achieved by a low-level joint controller.

b) Optimization variables and objectives

The optimal gait of humans, according to [106], is related to fitness of the species and is a function of several factors such as speed, acceleration, endurance, energy and stability. Human gait studies have shown that these can be modeled by simple principles and using equally simple high-level representations of gait. For example, step length and cadence have been shown to have a linear relationship [110]. Also, simple empirical equations of step length and step rate proposed by [81] lead to contours of energy consumption per meter which match subject data from different studies. In particular, the metabolic "cost of transport" (energy per unit distance) is a frequent optimization objective studied in human gait literature. Humans have been shown to choose an average step length and frequency that minimizes average energy cost per distance [81–84]. Minimization of vertical cost of transport, mainly by regulation of COM height, also explains locomotion patterns on steep slopes as shown by [107]. Studies usually model energy as oxygen consumption [81], joint or muscular work [111] and body or COM work [112]. Energy recovery [113] of the COM is also another considered objective related to COM work.

The previously stated measurements have been shown to vary systematically with high-level gait parameterizations such as step length [97–99, 103, 114], step width [103], speed [98, 105, 114], COM height [98], knee flexion

at heel strike [100], foot angle and velocity at heel strike [99, 103, 104, 115], double support and swing times [105, 110, 115], and limb stiffness [98]. The same variables have also been shown to be used, whether directly or indirectly, to regulate the Required Coefficient of Friction (RCOF): the ratio of shear to normal ground reaction force (i.e. tangential to normal force) [98–100]. The RCOF constraint should be kept below the ground's coefficient of friction to avoid slips and consequent falls, but it is planned and not just controlled reactively [98].

Travel time, acceleration and orientation error are also other functions which can be optimized to predict COM trajectories in flat goal-directed paths indoors [86].

2.3 Human-inspired models of energy and slippage

2.3.1 Model definition

From the anticipatory gait control studies mentioned previously we selected simple gait variables, as well as energy and slippage related functions, such that:

i. They are easily applicable to current humanoid locomotion planning algorithms, namely footstep planning;

ii. They predict walking behavior observations in different human gait literature for a variety of scenarios. In particular we focus on observations on slippery environments, flat and slanted terrain.

a) Gait variables

Step length, width and height. As discussed in Section 2.2, both energy and RCOF have been shown to vary systematically with these variables. Also, their application to robot footstep planning is straight-forward since these are simple distances between feet.

Double support time and leg swing time. As discussed, these vary systematically in adaptations to slippery terrain. Inclusion of these variables into (extended) footstep planning should add flexibility to the planner in order to lower gait accelerations. It may thus allow the robot to navigate

more slippery terrain.

Knee flexion angles. These also vary systematically in adaptations to slippery terrain [100]. Furthermore, they are related to COM height which explains adaptations in steep slopes and slippery terrain. For robot locomotion, planning COM trajectories is also crucial for stability and feasibility considerations. In this thesis we parameterize the COM height trajectory through inflexion points of a knee angle trajectory spline.

b) Gait objectives

COM work as optimization objective. As discussed in Section 2.2, energy optimization and in particular COM work explains walking patterns in both flat and sloped terrain [83, 107]. The advantage of this model for robotics when compared to, for example, electrical energy or torque minimization is basically its simplicity. Since only COM velocity and force profiles are required to estimate COM work, it applies to both complex robot models and simple single-mass robot models. There is also the motivation of passive dynamic walkers [116] which optimize COM work by construction.

For humanoid robots, we can learn a COM work model in simulation as a function of the previously stated gait variables. We compute total COM mechanical work as:

$$E_{\mathrm{COM}} = \int_{t_0}^{t_1} |\mathbf{v}.\mathbf{F}| dt, \tag{2.1}$$

where $\mathbf{v}$ and $\mathbf{F}$ are the velocity and total force vectors at the COM, respectively, and t_0, t_1 the beginning and ending time of a step (i.e. $t_1 - t_0 = \Delta t_{ds} + \Delta t_{sw}$).

RCOF as a constraint. As discussed, RCOF has been shown to vary on slippery terrain with the chosen variables.

RCOF [99] is defined as the maximum ratio of tangential-to-normal force applied at the feet during a given step:

$$\mathrm{RCOF} = \max_{t \in [t_0; t_1]} \left| \frac{F_{\mathrm{T}}(t)}{F_{\mathrm{N}}(t)} \right| \tag{2.2}$$

where F_{T} is the tangential force and F_{N} normal force at the feet. In this thesis we assume a Coulomb friction model. Therefore, note that if RCOF is lower than the actual coefficient of friction between feet and floor, slippage is theoretically prevented during that step. As with the energy model, we can learn a RCOF model in simulation as a function of the previously stated

gait variables.

We use function approximation to obtain each model as a function $\hat{f} : \mathbb{R}^{3+P} \rightarrow \mathbb{R}$ where inputs are the variables mentioned previously (i.e. step length, width, height and p) and outputs are the measurements E_{COM} and RCOF. Obtaining a model implies generating many walking patterns with different variable inputs, observing the energy and slippage results in physics simulation, and using these as training points for function approximation.

2.3.2 Our experimental platform: the WABIAN-2 humanoid

In our experiments we use the humanoid robot WABIAN-2 [117], which we show in Figure 2.1. WABIAN-2 is a human-size humanoid robot, 1.5 meters tall, weighting 64kg and having 41 DOFs. Joints are driven by DC-motors with high gear reduction ratios of around 200. Each motor is associated with one relative encoder and one motor driver for position control. We simulate the robot using the Open Dynamics Engine (ODE) for physics simulation on the V-REP robot simulator [118].

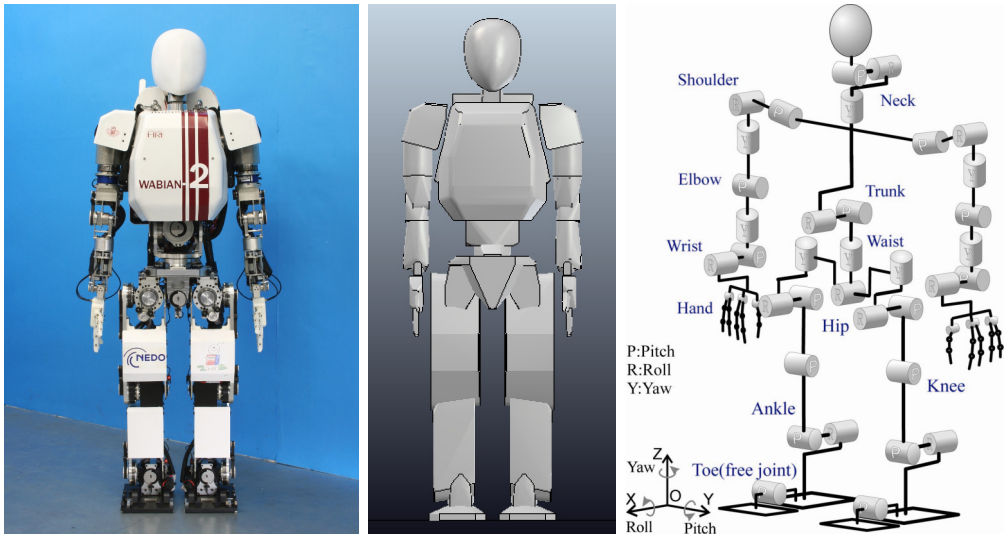


Fig. 2.1 The humanoid robot WABIAN-2, used in our simulation experiments. From left to right: real robot, simulated, DOF.

2.3.3 Resulting energy and slippage models on WABIAN-2

We trained the energy and slippage models by running physics simulations which explore the space of steps $(f_{j-1}, f_j, f_{j+1}, p_j)$ and collecting measurements of E_{COM} and RCOF. Each simulation consists of a symmetric and periodic gait of steps with constant step length, width, height and p. The patterns also start and finish with zero COM velocity and are stabilized with the Pattern Generator described in [119]. Since the simulations consist of symmetric periodic gait, step lengths (usually defined as the distance between two consecutive feet at heel strike [100]) are the same as stance lengths, and likewise for width and height.

We chose to use an Infinite Mixture of Linear Experts (IMLE) [120] for function approximation due to its high query speed and low number of experts, while still allowing for online learning if necessary. Error performance is comparable to that of Gaussian Processes [120]. Models were trained by uniform sampling of the input space and using the necessary number of experts to obtain a standardized mean squared error (SMSE) lower than 0.1.

We used Open Dynamics Engine (ODE) for physics simulation on the V-REP robot simulator [118], at a 4ms control cycle (ODE computation time step 1ms, global ERP 0.8, all other parameters set to their default values). The robot's joints are position controlled using the same gains as the real robot (proportional gain between 0.7 and 0.8). We used the Walking Pattern Generator described in [119] which stabilizes the walking motion based on the robot's full dynamical model and works for varying COM height motion. ZMP reference trajectories were placed at the center of the stance foot during the swing phase and cubic-spline-interpolated to the other foot during the double support phase. Full trajectories of the knees were obtained by cubic spline interpolation between a minimum flexion angle at impact ϕ_0 and maximum flexion angle at stance ϕ_{st} and swing ϕ_{sw}, as shown in Figure 2.2.

The limits of stance reachability were set according to the kinematic chain of WABIAN-2 by manual inspection:

- $\Delta x \in [0; 0.38]$ meters, where x points forward,
- $\Delta y \in [0.17; 0.30]$ meters, where y points to the left (symmetric interval if f_{j+1} is a right foot),

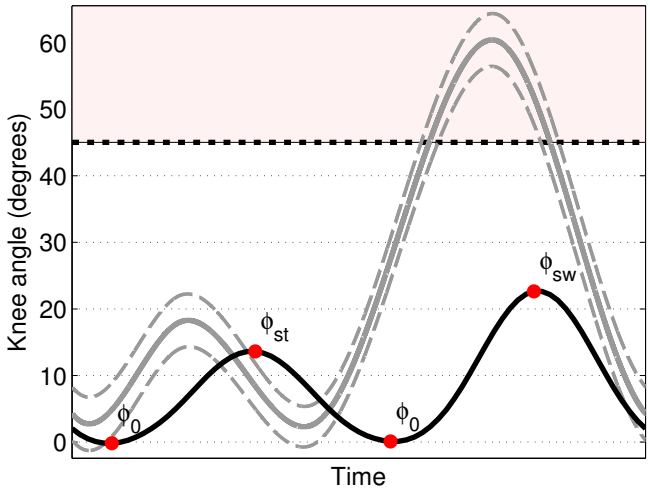


Fig. 2.2 Knee trajectories used for the robot are interpolated with a cubic spline between a minimum flexion angle ϕ_0 at impact and maximum angles at stance ϕ_{st} and swing ϕ_{sw}. Average and standard deviation of human data is plotted in gray based on [121]. The robot's curve can be made close to that of humans by adjusting double support time (moving ϕ_0 to the left in this example) and stance angle (ϕ_{st} up), however ϕ_{sw} cannot exactly match human data ($\phi \leq 45°$, pink region is unfeasible).

- $\Delta z \in [-0.15; 0.15]$ meters, where z points upward,
- $\Delta\theta \in [0; 30]$ degrees, where θ runs counter-clockwise (symmetric interval if f_{j+1} is a right foot).

The state transition (i.e. step) parameter vector was defined as $p = (\Delta t_{ds}, \Delta t_{sw}, \phi_0, \phi_{st}, \phi_{sw}) \in \mathbb{R}^5$, and sampled within the intervals:

- $\Delta t_{ds} \in [0.09; 1.8]$; $\Delta t_{sw} \in [0.9; 1.8]$ seconds,
- $\phi_0 \in [1; 21]$ degrees,
- $\phi_{st} \in [5; 45]$; $\phi_{sw} \in [5; 45]$ degrees.

Due to the high dimensionality of the models, we had to obtain thousands of training points from simulations. To reduce training time we trained two separate versions of each model: one for level, one for inclined terrain. We used all dimensions except Δz on the level terrain version, and an approximate model on inclined terrain. In the latter, knee trajectories have a narrow feasibility space (collisions, complex motion) and so we constrained them such as to obtain a fixed foot-COM height trajectory. With this approximation, models were learned in around 2 days of simulation. In total

we generated around 12,600 different walking patterns. Each pattern is a sequence of 6 symmetric steps of constant step length, width, height and p. From these simulations we gathered measurements of E_{COM} and RCOF.

Figure 2.3 shows the E_{COM} model as a function of step length and height, for two different slope friction values ($\mu = 0.2$ and 0.4). The energy at each steplength-stepheight combination also depends on the other parameters p, and so the minimum E_{COM} across p is shown at each point. The gradient of the energy is mainly dominated by the step height value, indicating high energetic cost for slanted terrain. The maximum feasible slope angle for each friction value can be seen by the absence of colored energy values, and is approximately 18° for RCOF$<$ 0.2 and 45° for RCOF$<$ 0.4. The high energetic cost of slanted terrain actually leads to a preference of shallow walking slopes as we will show in Section 2.4.2.

Figure 2.4 shows the contours of E_{COM} for level walking. The figure shows that most of the energy is spent in double support: the shorter Δt_{ds} the lower the energy. Leg swing time mostly does not influence COM energy, which reflects the fact that the alignment of velocity and force are low when compared to double support (motion on the sagittal plane is close to an inverted pendulum).

We show the RCOF model in Figure 2.5. RCOF is mainly dependent on the time spent in double support (contours are vertical in the right-most Δt_{ds}, Δt_{sw} plot). The higher Δt_{ds} is, the lower the RCOF. Also, the lower the step length, the lower the RCOF. Our interpretation is that both increasing Δt_{ds} and decreasing step length lead to lower COM accelerations during double support and thus a more static gait, because of that tangential forces are lower and so is RCOF. These observations match human data as we will discuss in the next section.

2.3.4 Comparison with human observations

The optimization objectives and variables proposed in this chapter were inspired by human gait literature. We now compare the results of our models and planner with the observations in human gait mentioned in that section.

a) Horizontal cost of transport [81, 83]

The plots in Figure 2.4 showed energy consumption per step. A known result from human biomechanics is, however, on the energetic cost per dis-

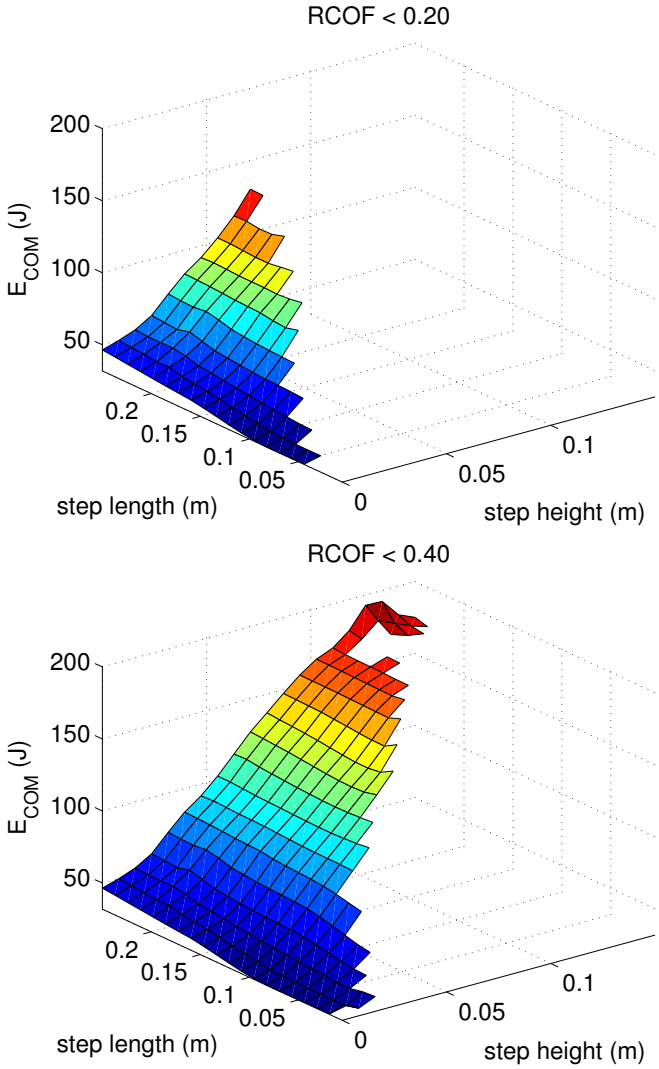


Fig. 2.3 Minimum E_{COM} on slopes, as a function of step length and step height. Measured in physics simulation.

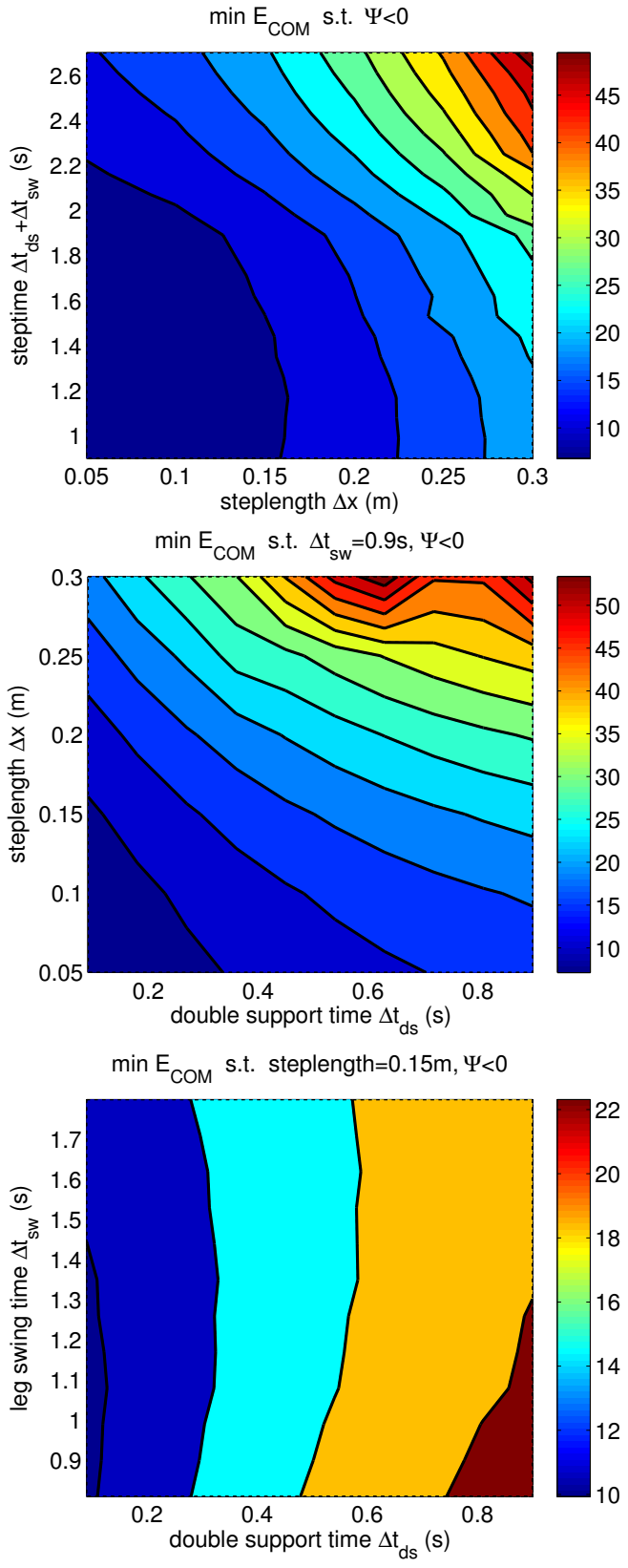


Fig. 2.4 Minimum E_{COM} on flat terrain, measured in physics simulation.

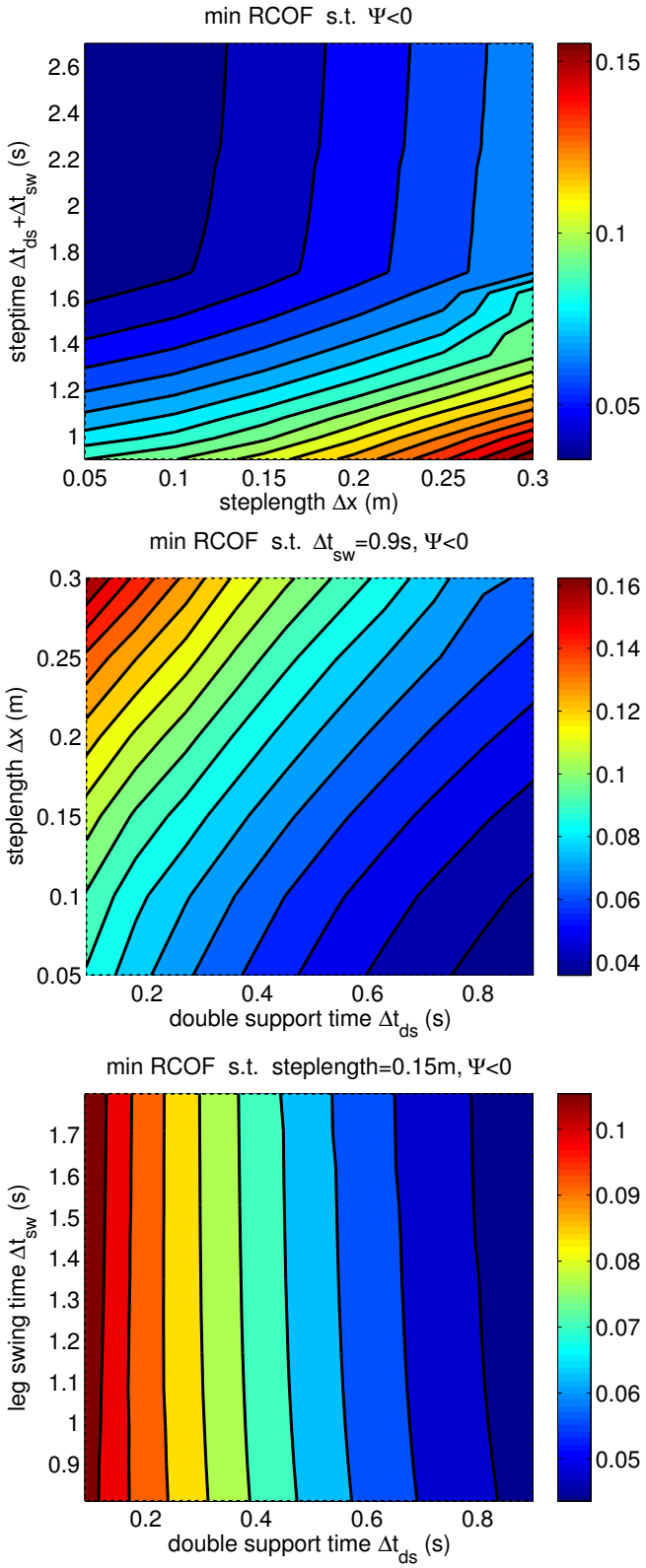


Fig. 2.5 Minimum RCOF on flat terrain, measured in physics simulation. RCOF is the maximum ratio of tangential-to-normal force over a step. It indicates the minimum ground coefficient of friction μ where the robot can walk without slipping.

tance (i.e. cost of transport). The contours of human oxygen consumption per meter in steplength-steprate space actually resemble an hyperbola [81]. An empirical formula explaining this data was estimated by Zarrugh et al. [81], using which we computed the energy consumption of a human with WABIAN-2's physical limits (maximum step length 0.35m, maximum step rate 1.20). Figure 2.6 shows the humans' cost of transport prediction, as well as WABIAN-2's actual cost of transport (i.e. minimum E_{COM} per distance). The hyperbolic shape of the energy contours is similar to both humans and robot. The energy minimum seems to be slightly shifted towards a higher step rate in the robot's case, which we assume to be due to motor efficiency once again, although it could also be related to a lower range of motion of the knees in our robot (up to 45 instead of 60 degrees). The similar shape is not surprising since it has also been reproduced by computer simulations of a simple bipedal walking model [83] using COM work optimization during toe-off.

b) Required Coefficient of Friction [98, 99, 103, 105]

Figure 2.5, which shows the robot's RCOF model, also matches observations in human gait. The figure shows that the higher Δt_{ds} is, the lower the RCOF. And also the lower the step length, the lower the RCOF.

2.4 Footstep planning with human-inspired models

2.4.1 The extended footstep planning algorithm

We now formulate the footstep planning problem using the human-inspired models of the previous section. We consider the problem of finding a sequence of N footsteps $f_j = (x_j, y_j, z_j, \theta_j) \in \mathbb{R}^4$, $j = 1, ..., N$, such that energy is minimized and with feasibility and no-slippage as constraints. The plan starts at a fixed initial stance $s_1 = (f_1, f_2)$ and finishes at a fixed goal stance $s_{N-1} = (f_{N-1}, f_N)$. N is unknown; (x_j, y_j, z_j) and θ_j are position and yaw orientation of a foot in a global coordinate frame; for convenience f_j is a left foot if j is odd, right if j is even.

The energetic cost E_{COM} of transitioning from a stance s_{j-1} to s_j depends on both the stances and some extra parameters $p_j \in \mathbb{R}^P$. p represents state

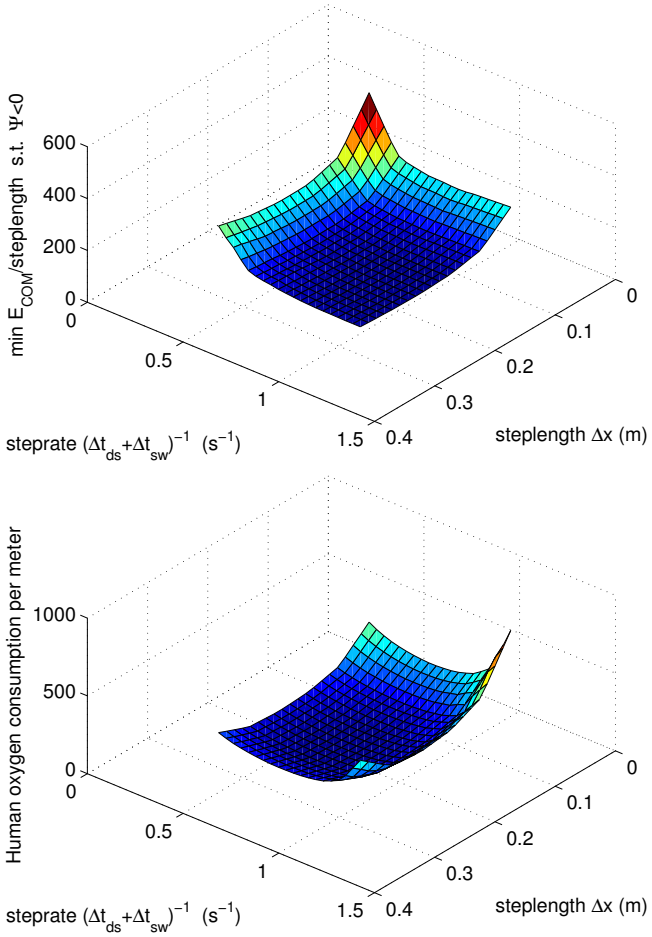


Fig. 2.6 Comparison of our robot's and humans' cost of transport. Top: Our robot's minimum E_{COM} per distance travelled. Bottom: Oxygen consumption of a human with WABIAN-2's physical limits, given by the empirical formulas of human walking of [81]. Units are in percentage of the minimum.

transition parameters that might provide different ways for s_j to be reached from s_{j-1}, such as step timing and COM motion. In this section we use $p_j = (\Delta t_{ds}, \Delta t_{sw}, \phi_0, \phi_{st}, \phi_{sw}) \in \mathbb{R}^5$ which are double-support time (i.e. time spent on s_{j-1}), swing time (i.e. time spent with the swing leg in the air), and minimum knee flexion, maximum stance knee flexion and maximum swing knee flexion angles. Throughout the thesis we will also refer to a state (i.e. stance) transition by a "step".

The general problem we are trying to solve in this section is

$$\begin{aligned} \underset{N, f_3..f_{N-2}, p_2..p_{N-1}}{\text{minimize}} \quad & \sum_{j=2...N-1} E_{\text{COM}}(f_{j-1}, f_j, f_{j+1}, p_j) \\ \text{subject to} & \\ \text{RCOF}(f_{j-1}, f_j, f_{j+1}, p_j) &< \min(\mu_{j-1}, \mu_j, \mu_{j+1}) \\ \Psi(f_{j-1}, f_j, f_{j+1}, p_j) &< 0 \\ a < p_j &< b \end{aligned} \tag{2.3}$$

where the function Ψ implements feasibility constraints on the stances and steps due to kinematic, dynamic or controller limitations. In this section we assume coefficient of friction μ_j is known for each f_j, and a Coulomb friction model so that RCOF is a tangential-to-normal force ratio. Bound constraints on the step parameters are implemented with vectors a and b.

Similarly to the human-inspired RCOF model, the feasibility model is learned in simulation. We define it as $\Psi \in \{-1, 1\}$ and use value 1 for unfeasible points and -1 for feasible. To discard obvious unfeasible stances we first use a footstep parameterization as in [13] to obtain a heuristic approximation of footstep reachability: in a stance s_j, reachability is approximated by a set of intervals for the variables $(\Delta x_{j+1}, \Delta y_{j+1}, \Delta z_{j+1}, \Delta\theta_{j+1})$, which are distances from the first footstep to the second, i.e., $\Delta x_{j+1} = x_{j+1} - x_j$, etc. Stances outside these intervals are considered unfeasible with $\Psi = 1$. Steps are also considered unfeasible if COM motion respecting the reference ZMP trajectory cannot be found using our Walking Pattern Generator [119], joint limits are reached or the robot falls during physics simulation. Similarly to the energy and slippage models, we still fit a continuous mixture model even though training points are discrete $\Psi \in \{-1, 1\}$, leading to interpolation regions between -1 and 1. While planning, we enforce a slightly conservative feasibility constraint of $\Psi < 0$ to avoid uncertain regions far from feasibility ($\Psi = -1$).

We solve (2.3) by a hybrid discrete search and continuous optimization-based planner. We first constrain the footstep (position) space to a point cloud of traversable points $(x, y, z) \in \mathbb{R}^3$ and a discrete set of orientations in the global coordinate frame: $\theta \in \{0^\circ, \frac{360}{D}^\circ, ..., \frac{360(D-1)}{D}^\circ\}$, where D is the number of uniform footstep directions. Then we compute the optimal-cost path from the initial to goal stance on this space using Anytime Repairing A* (ARA*) [122]. ARA* requires a state transition cost function $c(s_{j-1}, s_j)$, and a heuristic cost-to-go function $h(s_j)$. It will find the optimal path to the goal given enough computation time and an admissible h. If interrupted anytime, then the algorithm still returns a sub-optimal path with provable bounds. Please refer to [122] for further details.

In our case the state transition cost $c(s_{j-1}, s_j)$ is the minimum-energy transition between the two consecutive stances $s_{j-1} = (f_{j-1}, f_j)$ and $s_j = (f_j, f_{j+1})$, given by:

$$\begin{aligned} c(s_{j-1}, s_j) = \min_{p_j} \; & E_{\text{COM}}(f_{j-1}, f_j, f_{j+1}, p_j) \\ & \text{subject to:} \\ & \text{RCOF}(f_{j-1}, f_j, f_{j+1}, p_j) < \min(\mu_{j-1}, \mu_j, \mu_{j+1}) \\ & \Psi(f_{j-1}, f_j, f_{j+1}, p_j) < 0 \\ & a < p_j < b \end{aligned} \tag{2.4}$$

Hence, even though states in A* search are discretized stances, step parameters are computed from continuous optimization on the state transitions.

Regarding the heuristic $h(s_j)$, we set it equal to a lower bound on the cost from s_j to the goal which assumes no obstacles, optimal cost of transport and infinite friction. This way $h(s_j)$ never overestimates the true cost to the goal (i.e. is admissible), as required for A* optimality. We compute the bound as the minimum horizontal cost of transport times distance:

$$\begin{aligned} h(s_j) = d_{xy}(s_j, s_{N-1}). \min_{f_k, f_{k+1}, p_k} \; & \frac{E_{\text{COM}}(f_{k-1}, f_k, f_{k+1}, p_k)}{d_{xy}(s_{k-1}, s_k)} \\ & \text{subject to:} \\ & \Psi(f_{k-1}, f_k, f_{k+1}, p_k) < 0 \\ & a < p_k < b \end{aligned} \tag{2.5}$$

where $d_{xy}(s_j, s_{N-1})$ is the Euclidean distance on the horizontal plane from stance s_j to stance s_{N-1} (i.e. the distance between left feet and the right feet summed). True costs to goal will actually be higher than (2.5) since optimal step parameters might not be feasible for the whole distance and more costly paths might be necessary due to kinematics constraints, obstacles, friction or slope.

In practice, we pre-compute and store on a hash table the results of equation (2.4) for a large number of footstep displacements and coefficients of friction. Similarly, we only need to solve the optimization problem in (2.5) once. Planning a path from an initial stance s_1 to a goal stance s_{N-1} then consists of a straightforward ARA* (or A*) search where each time a state transition is considered we:

1. access a hash table to obtain the state transition cost (2.4)
2. compute the heuristic cost-to-go from the distance to goal and the pre-computed cost-of-transport using (2.5).

2.4.2 Resulting paths and energy consumption

We will now use the described footstep planning algorithm together with the human-inspired models of the previous section to plan footsteps in a variety of scenarios with geometry and friction constraints. We will analyze the walking paths generated by the described E_{COM}-optimal planner in practice, as well as the paths' expected electrical energy consumption. Our motivation for estimating electrical energy consumption was not only due to its practical value in robotics, but also because mechanical work in humans is related to metabolic energy (i.e. oxygen consumption) [84, 107]. Since the real WABIAN-2's joints are driven by DC-motors [123], we compute electrical energy as

$$E_{\text{ele}} = \sum_i \left(\int_{t_0}^{t_1} |\tau_i \omega_i| dt + \int_{t_0}^{t_1} R_i I_i^2 dt \right) \tag{2.6}$$

where i is an index of the motor, τ is motor torque and ω angular velocity. I refers to current, which in simulation is computed as $\tau/(r.K_\tau)$, where r is the motor's gear reduction ratio and K_τ the torque constant, taken from the motors' data sheets. RI^2 are the power losses due to motor armature resistance and we ignore mechanical losses such as joint friction. We will compare the resulting electrical energy consumption obtained by our planner with a set of baselines:

i. minimum-travel-time planner,

ii. minimum-sum-of-torques planner,

iii. directly optimizing electrical energy consumption E_{ele} as defined in (2.6). The results for the baselines were obtained using exactly the same planner equations (2.4) (2.5) and implementation, the only difference being that we replaced E_{COM} by $(\Delta t_{ds}+\Delta t_{sw})$, $\int \sum_i \tau_i^2 dt$, and E_{ele} respectively.

a) Implementation details

As mentioned previously, we solved (2.4) for a large number of footstep displacement and μ values; and stored the results on a hash table. In our experiments this hash table had $18,491$ entries. To solve (2.4) this many times took approximately 2 hours. When planning, we simply query the table to obtain state transition costs and step parameters p from the transitions' footstep displacement and μ values. Query time is at the microsecond level.

We implement point cloud discretization with PCL [124] using 5cm grid-filtered point clouds. The search for successors of a stance is done by a range search of points around the fixed foot. Also, the directions of footsteps were discretized uniformly with $D = 24$.

We use the official implementation of ARA* [122] in the Search-Based Planning Library (SBPL) [125]. The optimization problems (2.4) (2.5) are first solved with the global optimization algorithm DIRECT [126], which is then refined using the sequential quadratic programming algorithm SLSQP [127]. Both optimization algorithms are implemented in the NLOpt library [128]. The functions E_{COM}, RCOF and Ψ are each implemented as an infinite mixture of linear experts (IMLE). During ARA* search we use pre-computed versions of (2.4) for speed. However, after the final solution is obtained we further refine the step parameters p by solving (2.4) using SLSQP, warm-started by the values stored on the hash table.

b) Results

We conducted the experiments in three different scenarios which we will now describe and analyze. Energy consumption results are reported in Table 2.1.

The first scenario (Figure 2.7) was as follows: the robot stands in a ground with friction $\mu_{\text{ground}} = 1.0$ and has to walk to a target which is

straight ahead, 3m away. Between the start and finish points there is an "ice patch" of very low friction μ_{ice}. We conducted several planning experiments with different $\mu_{\text{ice}} \in \{0.12, 0.06\}$ and different widths of the ice patch ($\{0.5, 1\}$m). Figure 2.7 shows that using our planner the robot walked through the ice for $\mu_{\text{ice}} = 0.12$ (specifically it walked 5% slower than the optimal speed with increased double support), but walked around the ice if $\mu_{\text{ice}} = 0.06$. When we doubled the ice patch width but kept the low friction $\mu_{\text{ice}} = 0.06$, the planner found it more optimal to go through the ice approximately twice as slow (with increased double support) than around a great distance. In terms of expected electrical energy (Table 2.1), the paths generated by our planner spent 2110 J, 2427 J and 3031 J respectively. We also conducted experiments constraining the planner to take the alternative, sub-optimal choice of avoiding the ice patch when it is optimal to cross it and vice-versa. Such sub-optimal choices would lead to 14%, 19% and 10% more electrical energy respectively. Thus, an increase in COM work (sub-optimal plan) lead to an increase in electrical energy consumption. The electrical energy obtained by our optimal planner was relatively close to the real minimum of E_{ele}. Optimizing electrical energy directly lead to 25%, 12% and 18% less consumption than optimizing COM work. On the other hand, optimizing travel time (common objective function of footstep planners) would lead to drastic energy spending, increasing by 26%, 51% and 92%. Optimizing joint torques decreased energy spending slightly by 11%, 3% and 13%.

The second scenario (Figure 2.8) was as follows: there are two stairs at equal distance to the robot ($x = 1$ meter away, $y = \pm 0.50$m), both ending at the same final height ($z = 0.50$m). One of the stairs has 3 high steps while the other has 6 lower steps. The goal of the robot is to reach a distant centered position $(x, y, z) = (3, 0, 0.5)$m. The energy cost should be the same if the stairs were identical. We show the obtained footstep plan in Figure 2.8. The figure shows that the planner opts for the lower-but-many-step stairs. The reason for this result is that on steep stairs, steps become too costly for the distance traveled. Notice that the slope of the energetic cost E_{COM} in Figure 2.3 is high in the direction of step height. We will further analyze the cost of slanted locomotion in Section 2.4.3. In terms of expected electrical energy (Table 2.1), our planner's path was 13% away from the true minimum of E_{ele}. The sub-optimal choice of taking the few-but-high stairs would increase consumption by 9%, and optimizing travel time would also

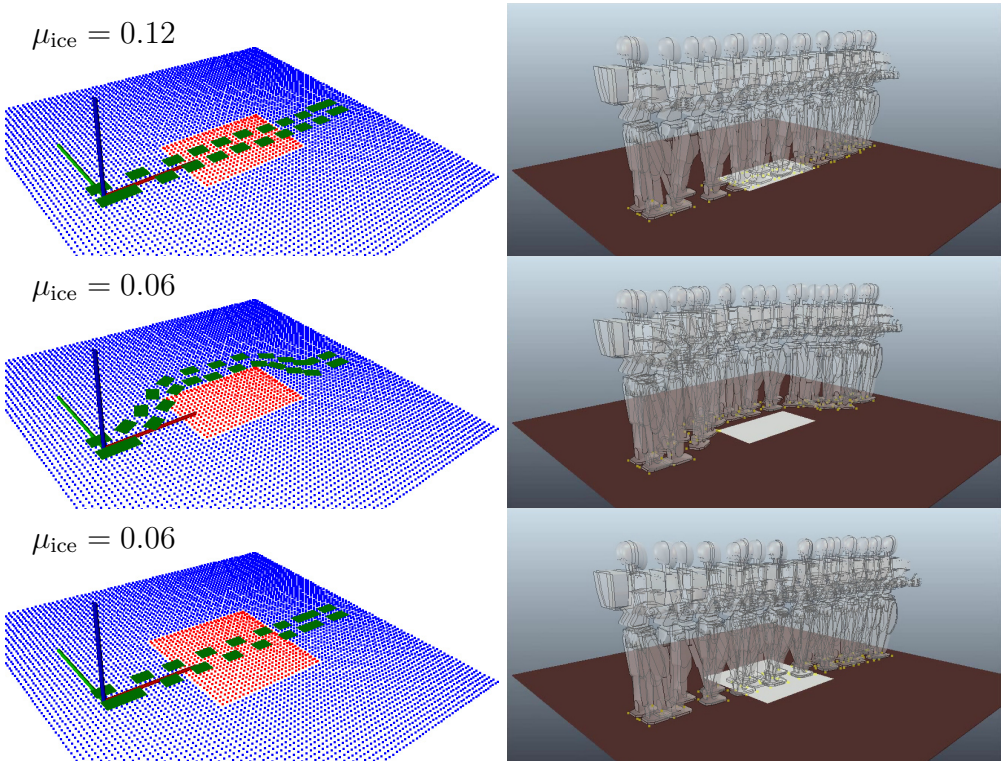


Fig. 2.7 Optimal plans obtained by our planner in the "ground and ice-patch" scenario. The top row shows the footstep plan and point cloud (red has friction μ_{ice}, blue $\mu_{\text{ground}} = 1$). Left: robot crosses a narrow ice patch ($\mu_{\text{ice}} = 0.12$). Middle: robot walks around the patch if its slipperiness is increased ($\mu_{\text{ice}} = 0.06$). Right: robot walks slowly through the same ice patch in case the ice is wider (energy spent avoiding it would be too high).

increase consumption by 40%. Optimizing joint torques lead to basically the same performance as E_{COM} (0.5% more energy).

The final scenario was as follows: the robot has to climb a slope to a target which is straight ahead, 2.5m away measured on a straight line connecting the start and target points. The slope has an angle of $\alpha \in \{10, 20, 25\}$ degrees. We show the planner and simulation results in Figure 2.8. The optimal path for the two shallowest slopes was in a straight line to the target, but for $\alpha = 25°$ the optimal path was curved and at a slightly lower inclination. These results match observations in human mountain paths as we will discuss in Section 2.4.3. In terms of expected electrical energy (Table 2.1), our planner's path for the 25 degree slope is only 5% away from the true minimum of E_{ele}. The sub-optimal choice of taking a straight path to the target, instead of curved, would increase consumption by 1%. Optimizing travel time would increase consumption drastically by 97%. Obtaining a path by optimizing joint torques revealed to be unfeasible for our planner's time limit (which was 10 minutes), while an optimal plan was returned for E_{COM} in 10 seconds. By analyzing our model and planner data our conclusion is that the sum-of-torques function has high variance due to noise in simulated joint torque measurements, and its optimization is prone to get stuck in local optima. The electrical energy minimizing planner also includes a joint torques term and correspondingly also took longer to solve the path to optimality (177 seconds) than when using COM work.

For all scenarios our E_{COM}-optimal planner found a first sub-optimal path within 1 second and the optimal path within 1 minute. The computational speed improvement obtained by using pre-computed energy costs for different step-friction combinations was of around one order of magnitude for both the initial and optimal paths. The ODE-simulated robot successfully walked without falling in all situations, even at high slipperiness and slope levels.

From the optimal-vs-suboptimal experiments our results indicate that E_{COM} correlates well with E_{ele}. Still it was less susceptible to local minima and long planning times than E_{ele} or torque-minimization. These three quantities (E_{COM}, E_{ele}, sum-of-torques) are all actually related with each other: Pearson correlation on data used for energy model training was $r = 0.78$ between joint torques and E_{ele}, $r = 0.54$ between joint torques and E_{COM}, $r = 0.58$ between E_{COM} and E_{ele}, and $r = 0.64$ between E_{COM} and joint mechanical work. Practically for our setup the human-inspired E_{COM}

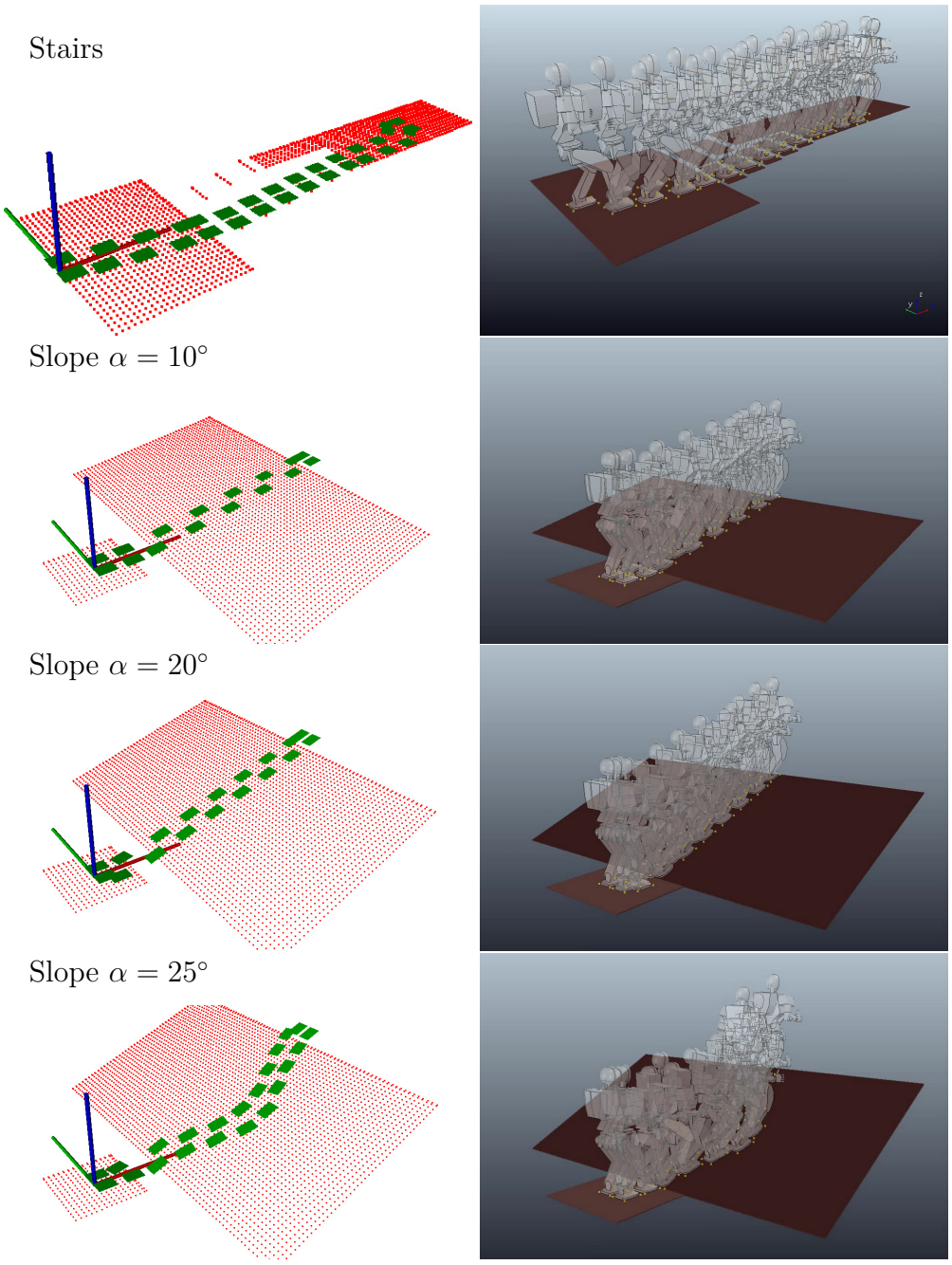


Fig. 2.8 Optimal plans obtained by our planner in the "Stairs" and "Slope" scenarios. On steep stairs and slopes, it is more energy optimal to walk a longer inclined distance but at a lower angle.

seems to be the best objective function choice as a compromise between energy consumption and computation time. Better optimization techniques could probably make direct optimization of E_{ele} more interesting, but in any case our proposed planner can be applied to both functions.

2.4.3 Comparison with human observations

The optimization objectives and variables proposed in this section were inspired by human gait literature, as described in Section 2.3. We now compare the results of our models and planner with the observations in human gait.

a) Gradient of mountain paths [106, 107]

As we showed in model and planning results in Figure 2.3 and 2.8, high E_{COM} of slanted terrain leads to a preference of our planner towards shallower slopes. In our example scenarios, the robot preferred low-step stairs, and planned a curved 20 degree path on a steep 25 degree slope. Likewise in humans, mountain paths are predicted by oxygen consumption experiments on slopes [106, 107]. According to [107], humans prefer to climb steep mountains at a maximum inclination of approximately 14 degrees, and in order to do that they climb not straight to the mountain peak but in a curved pattern. Mountain path observations are also partly reproduced by assuming minimization of COM mechanical work [107] which is our objective function in this section. In Figure 2.9 we plot the chosen climbing angle versus the straight-line slope angle both for humans and our robot. The curve corresponding to humans was obtained by the data in [107]. The curve's shape is the same for humans and our robot: straight-line path until a certain angle, constant lower climbing angle after that. The angle at which this transition occurs is however different (approximately 14° for humans, 20 for the robot). We believe this to be due to differences in motor efficiency since WABIAN-2's weight, dimensions and joint positions are inspired by humans. We calculated the extra (constant) energy consumption of humans that would lead to the same plot as our robot's, and found it to be 0.5cal/kg/m. This curve is also shown in Figure 2.9.

Table 2.1 Estimated electrical energy consumption of our planner using different objective functions

Scenario	E_{COM} (ours)	Suboptimal E_{COM}	Travel time	Sum-of-torques	E_{ele} (ideal energy consumption)
Narrow ice $\mu = 0.12$	2110 J	+14%	+26%	-11%	-25%
Narrow ice $\mu = 0.06$	2427 J	+19%	+51%	-3%	-12%
Wide ice $\mu = 0.06$	3031 J	+10%	+92%	-13%	-18%
Stairs $\mu = 1$	4116 J	+9%	+40%	+0.5%	-13%
Slope $\mu = 1$, $\alpha = 25°$	4908 J	+1%	+97%	(failed)	-5%

*Note: Reported energy is the estimated electrical energy consumption (2.6). Percentage values represent additional energy as a percentage of E_{COM} (i.e. $(E' - E_{\text{COM}})/E_{\text{COM}}$). "Suboptimal E_{COM}": refers to a plan that takes a sub-optimal navigation option (i.e. around the ice instead of through; through instead of around; using the few-but-high-step stairs; walking straight on a 25° slope instead of in a curve) although still optimizing E_{COM} given that constraint.

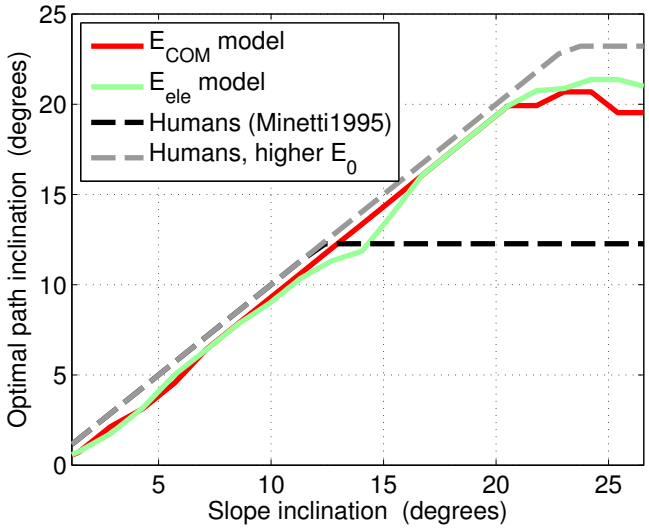


Fig. 2.9 Optimal path inclination angle α_{path} as a function of the slope angle α. If $\alpha_{\text{path}} < \alpha$ then the path is curved at shallower inclination and longer total distance.

b) Required Coefficient of Friction [98, 99, 103, 105]

According to [99] humans reduce RCOF (the shear-to-normal force ratio) when walking on slippery terrain, which in our planner we assume to be a walking constraint such that RCOF$< \mu$. Figure 2.5 shows an increase of RCOF for an increase in double support, which means that to be able to walk on more slippery terrain (lower RCOF) the robot should opt for a conservative gait that is more static, with lower tangential speeds and accelerations. Also in humans a "cautious", more static, gait has been observed in humans walking on slippery terrain [98–100], as referred in Section 2.2.3. On the other hand [105] specifically observed an increase in double support time when walking on slippery terrain. Regarding step length, [98, 99, 103] also observed that this variable is lower when humans walk on slippery terrain. Reduction of step length is actually an anticipation strategy used just before stepping on slippery terrain [98, 99, 103], just as in our robot's case it is planned by assuming a constraint on RCOF [99]. While our planner uses a hard RCOF constraint, the decision was mainly motivated for practical and conservative reasons: a hard constraint lowers the risk of falling by theoretically avoiding slippage completely and thus not having to rely heavily on reactive slippage control. Humans, on the other hand, could possibly use RCOF or a related metric as a soft constraint, although we are not aware of any investigation on these lines.

2.4.4 Simulated versus real robot

We ran a small subset of experiments on the real robot to compare real electrical energy E^*_{ele} to the simulated E_{ele} model. Figure 2.12 shows the simulated optimal E_{COM} per step-length and E_{ele} per step-length, for several step-length values while varying all other step parameters. We also show the real measured E^*_{ele} per step-length in the same figure for comparison. To obtain E^*_{ele} we made the robot walk in the laboratory for a total of 18 steps for each step-length value (using the energy-optimal step parameters obtained from simulated models). We used motor current measurements given by the motor drivers, and computed torques from current. Each point in the graph is the average energy over the 18 steps. The standard deviation of the measurements is also shown in the same figure. The minimum energy per distance is obtained at the same step-length of 0.15m for all models (i.e. 0.30m stride length). The standard deviation of the energy measurements on the real robot is low, especially at the optimum, which we believe to be due to higher stability as well. The figure also shows that COM mechanical energy E_{COM} overestimates energy consumption after the minimum, and that this overestimation is lower in case a more complex model is used (i.e. joint work plus a τ^2 term). Figure 2.10 shows one of the real robot experiments taken at optimal step-length.

To observe the impact of mechanical energy and heat losses in E^*_{ele}, we also show in Figure 2.11 the total electrical power across a 6-step experiment. We decompose power into joint mechanical power (first term of equation 2.6) and power losses (second term of equation 2.6). Mechanical power dominates power consumption over heat losses of the DC motors, and closely follows the total energy of the system. This agrees with the other results, in that optimizing mechanical work might be sufficient for good energy consumption of the robot.

2.5 Hierarchical full-body planning

2.5.1 Hierarchical planning architecture

We will now build on the "extended footstep planning" algorithm of the previous section to plan full-body motion using a *contact before motion* approach.

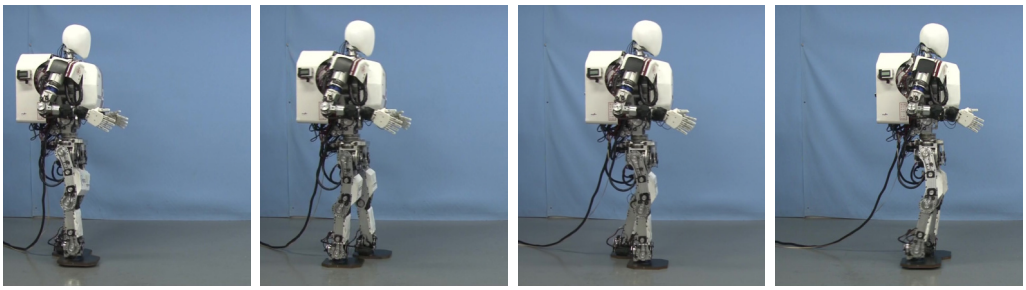


Fig. 2.10 Real robot walking with E_{COM}-optimal parameters (red dot in Figure 2.12).

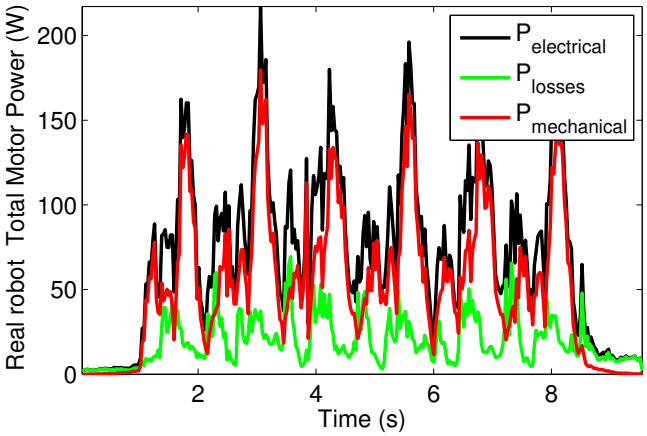


Fig. 2.11 Real total electrical power measured over a 6-step trial. $P_{\text{electrical}} = P_{\text{mechanical}} + P_{\text{losses}}$.

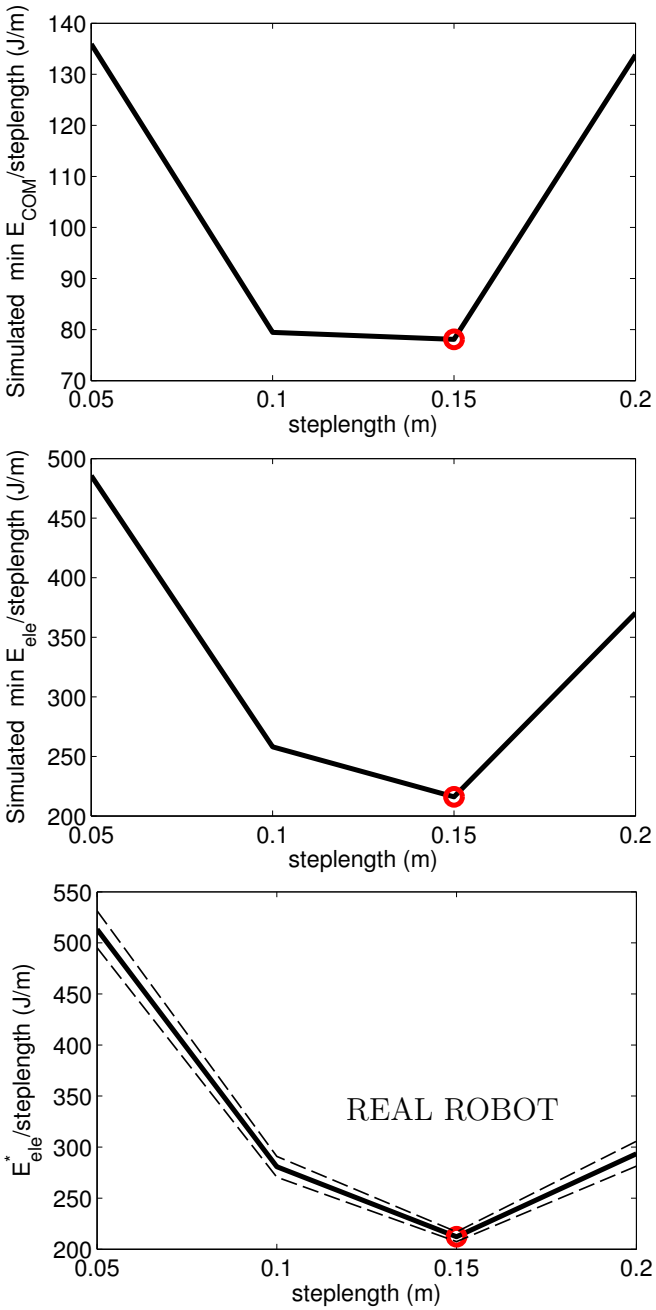


Fig. 2.12 Simulated versus real energy consumption. Mechanical energy E_{COM} (left), simulated electrical energy E_{ele} (middle) and measured electrical energy E^*_{ele} (right). The real energy curve was obtained by averaging over 18 steps for each step-length and standard deviation is also shown.

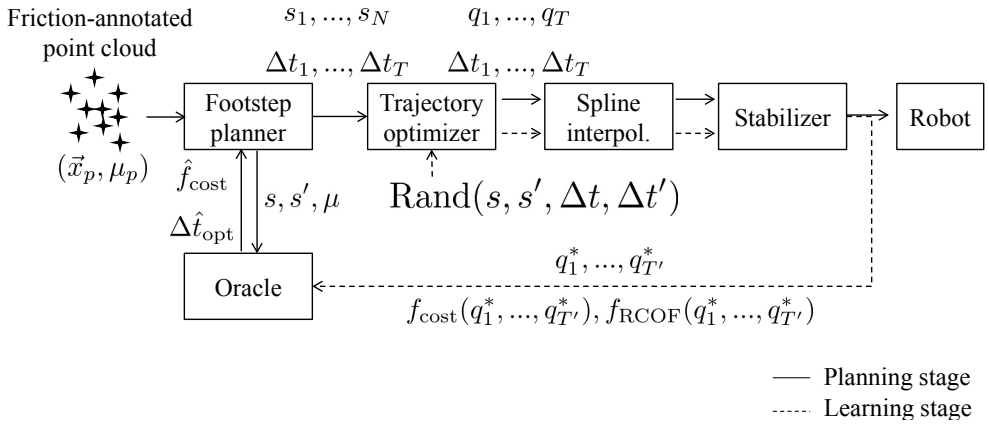


Fig. 2.13 Our hierarchical planning architecture, which uses trajectory optimization to minimize a cost function f_{cost}, as well as oracle costs to plan footstep placement and timing that will have low predicted f_{cost}.

When a planning problem is organized in such a hierarchy, it is important to enforce consistency between the functions optimized at each level, in order to avoid very suboptimal results. For example, if E_{COM} is optimized at the footstep level, then the subsequent full-body trajectory optimizer should arguably optimize E_{COM} with body motion as well. If the energy function optimized at the footstep planning level correctly predicts the energy obtained by the full-body planner, then footsteps will be full-body-optimal. In the context of hierarchical planning, it then makes sense to call the energy and slippage models used in the footstep planner by "oracles" - black boxes which predict the output of a full-body trajectory optimizer.

See Figure 2.13 for a visual representation of the architecture we propose. A footstep planner first searches a stance graph using transition costs provided by an oracle. The stances are then used as constraints in a full-body trajectory optimizer that considers full-body trajectory costs, collisions, joint limits and static stability. The obtained trajectory is finally interpolated and locally adapted for dynamic stability using a ZMP-based method. The oracle basically takes each stance transition and predicts the costs obtained at the end of the whole planning pipeline. This leads to footstep plans which optimize the same criteria as the full-body planner. The trajectory costs could be E_{COM}, as used in the previous section, or any other reasonable function. In this section we will use squared static torques, despite their slightly lower energetic performance, mainly since they are easily integrated into trajectory optimization.

a) Extended footstep planning with an oracle

When considering full-body trajectory optimization it makes sense to slightly generalize the definition of a stance s as a set of contacts with the environment. A contact is a tuple (link, position, rotation), and a neighbor stance s' either adds or removes a contact with respect to s. Since we are still dealing only with biped walking in this thesis, stances will transition from double-support to left-foot-contact, to double-support, to right-foot-contact, back to double-support, etc. The advantage of this representation instead of using double-support stances only is that the swept-volume between consecutive stances can be used by the optimizer to guide a swing leg out of collision. Such an approach is also used by other works focusing on collision detection [88]. Furthermore, on top of the stance feasibility constraints Ψ, for collision safety we will now also compute foot-foot and COM-environment collision checking using bounding boxes for the feet and trunk.

The state transition and heuristic equations (2.4) (2.5) are similar, except we now switch the notation to stances and their neighbors. We define them as

$$\begin{aligned} c(s, s') = & \min_{p} \ \hat{f}_{\text{cost}}(s, s', p) \\ & \text{subject to} \\ & \hat{f}_{\text{RCOF}}(s, s', p) < \mu \\ & \Psi(s, s', p) < 0 \\ & a < p < b, \end{aligned} \tag{2.7}$$

$$h(s) = d_{xy}(s, s^{\text{goal}}) . \min_{s, s', p} \frac{\hat{f}_{\text{cost}}(s, s', p)}{d_{xy}(s, s')}. \tag{2.8}$$

The functions $\hat{f}_{\text{cost}}$ and $\hat{f}_{\text{RCOF}}$ serve the same purpose as the learned models of the previous section, but are now given not by physics simulations but by an oracle which predicts the value of f_{cost} and f_{RCOF} obtained at the end of the whole planning pipeline. We implement $\hat{f}_{\text{cost}}$ and $\hat{f}_{\text{RCOF}}$ as hash tables. The tables are filled offline, by feeding the whole planning pipeline (i.e. trajectory optimization, interpolation, dynamic stabilization) with uniformly distributed samples of (s, s', p) as shown in Figure 2.13. The discrete optimization problems in (2.7), (2.8) are then solved for a large number of

discretized stances and coefficient of friction values, and finally stored in new hash tables for fast access to costs and heuristics during search.

b) Full-body trajectory optimization

The full-body trajectory optimizer takes a footstep plan with N stances and produces a full-body trajectory, parameterized by T discrete-time waypoints. Waypoints are full-body robot configurations $q_t \in \mathbb{R}^D$, $t = 1, ..., T$, where D is the number of degrees-of-freedom consisting of the joints' angle values and the pose of the robot base. Each stance is associated with 2 full-body postures (at start and midstance) and so $T = 2N$. For convenience we use s_t to refer to the stance associated to q_t.

Our optimizer solves the problem

$$\underset{q_1,...,q_T}{\text{minimize}} \quad f_{\text{cost}}(q_1, ..., q_T) + \alpha f_{\text{collision}}(q_1, ..., q_T) \tag{2.9a}$$

subject to

$$f_{\text{stance}}(q_t, s_t) = 0 \quad \forall_{t\in 1,...,T} \tag{2.9b}$$

$$f_{xy}(q_t) \in \mathcal{P}_t \quad \forall_{t\in 1,...,T} \tag{2.9c}$$

$$f_{\text{roll}}(q_t) = 0 \quad \forall_{t\in 1,...,T} \tag{2.9d}$$

$$A_t q_t \leq b_t \quad \forall_{t\in 1,...,T}, \tag{2.9e}$$

where $q_1, ..., q_T$ are the optimization variables, α is a penalty constant and:

- The function f_{cost} computes the sum of the squared static torques of all joints at all waypoints. In the static condition joint torques only compensate for gravity, and so f_{cost} is given by

$$f_{\text{cost}}(q_1, ..., q_T) = \sum_{t=1}^{T} \tau(q_t)^\top \tau(q_t), \tag{2.10}$$

$$\tau(q_t) = \sum_{i=1}^{L} J_i(q_t)^\top F_{g_i}, \tag{2.11}$$

  where $\tau(q_t)$ is the vector of joint torques at configuration q_t, L is the number of links of the robot, $J_i(q) = \frac{\partial x_i}{\partial q}$ is the COM position Jacobian of link i, and $F_{g_i} = m_i \begin{pmatrix} 0 & 0 & 9.8 \end{pmatrix}^\top$ is the force of gravity applied at link i. The function is implemented in the *trajopt* library [57], which we use in our setup.

- The function $f_{\text{collision}}$ is a collision cost as proposed by [57] and implemented in *trajopt*. It is the sum of a discrete collision cost computed

by the signed distance between each link and all other geometries, and a continuous collision cost computed by the signed distance between the swept volume of each link with the environment.

- The function $f_{\text{stance}}(q_t, s_t)$ computes the pose error of all links in contact as a $6C$-dimensional vector where C is the number of active contacts in s_t. This is computed as the translation and axis-angle error between the target link pose (given by s_t) and the current link pose (given by q_t).

- The function $f_{xy}(q_t)$ computes the (x,y) coordinates of the COM, and $\mathcal{P}_t$ is the support polygon of s_t. The constraint thus enforces approximate static stability. The support polygon of s_t is computed by the convex hull of the horizontal projection of links in contact and does not include contacts removed in s_{t+1}.

- The function $f_{\text{roll}}(q_t)$ computes the rotation around the X axis for the waist link, with respect to the global reference frame. This constraint is necessary as "zero roll" is an assumption of the subsequent dynamic stabilization method.

- A_t, b_t enforce joint angle and velocity limits.

We solve problem (2.9) using the Sequential Quadratic Programming method of [57] as implemented in the *trajopt* library[1].

c) Interpolation and stabilization

To obtain a densely-sampled trajectory for execution on the robot, we interpolate trajectory waypoints using hermite cubic splines with derivatives set to zero for smooth contact transitions. The time between two consecutive waypoints q_t is given by the oracle (equation Equation (2.7)).

Since the obtained trajectory is not dynamically stable, we then apply an FFT-based ZMP trajectory compensation scheme [119]. The method considers the rigid-body dynamics of the full body and locally adapts COM motion on the horizontal plane using analytic inverse kinematics to iteratively reduce the error between the real and reference ZMP trajectory. We set the reference ZMP trajectory to the interpolated $f_{\text{xy}}(q_t)$, which were

[1]URL: http://rll.berkeley.edu/trajopt

used in the optimization problem (2.9) and are inside the support polygon at each waypoint. Furthermore, our implementation of the analytic inverse kinematics of the robot WABIAN-2 assumes zero roll angle of the waist link with respect to the world reference frame. We include this constraint in the optimization problem (2.9) for consistency.

2.5.2 Results

In the following experiment we gave the planner the task of computing a full-body trajectory from an initial stance in double-support to a goal stance 1.5 meters ahead. In the middle of the trajectory we placed a high obstacle which would lead to collision if a full-body planner was not used. See 2.14 for the results. Trajectory optimization parameters are the collision penalty weight α of equation (2.9), which we set to 50, and the distance at which the collision penalty starts being applied (for all links except those in contact), which we set to 2.5cm. The collision penalty distance for the head link was set to a higher value of 10cm for clearly visible safety in the figures.

Collision checking during footstep planning is made with a slightly shrunk bounding box: 20cm lower than if the robot were fully stretched. This is so that plans are found according to the maximum capabilities of the robot (i.e. assuming it can bend down to 20cm). Thanks to this, in this experiment a footstep plan was found on a straight line to the target and the full-body motion automatically bent the trunk and knees in order to avoid collision between the head and the obstacle.

We can further complicate the environment with obstacles that force a footstep detour. See Figure 2.15 for an example. In this case, we placed an extra small obstacle in the middle of the course but close to the ground to force the footstep planner around it. In this environment, the footstep planner obtains a sequence of footsteps free of bounding-box-collision which goes around the low obstacle through the left. Then, the full-body planner obtains a trajectory that respects those footsteps. Once again, the high obstacle forces the robot to bend down to avoid head collision. At all stages, energy is minimized and therefore in both experiments (Figures 2.14 and 2.15) paths are short - and knee-stretched when that does not lead to collision.

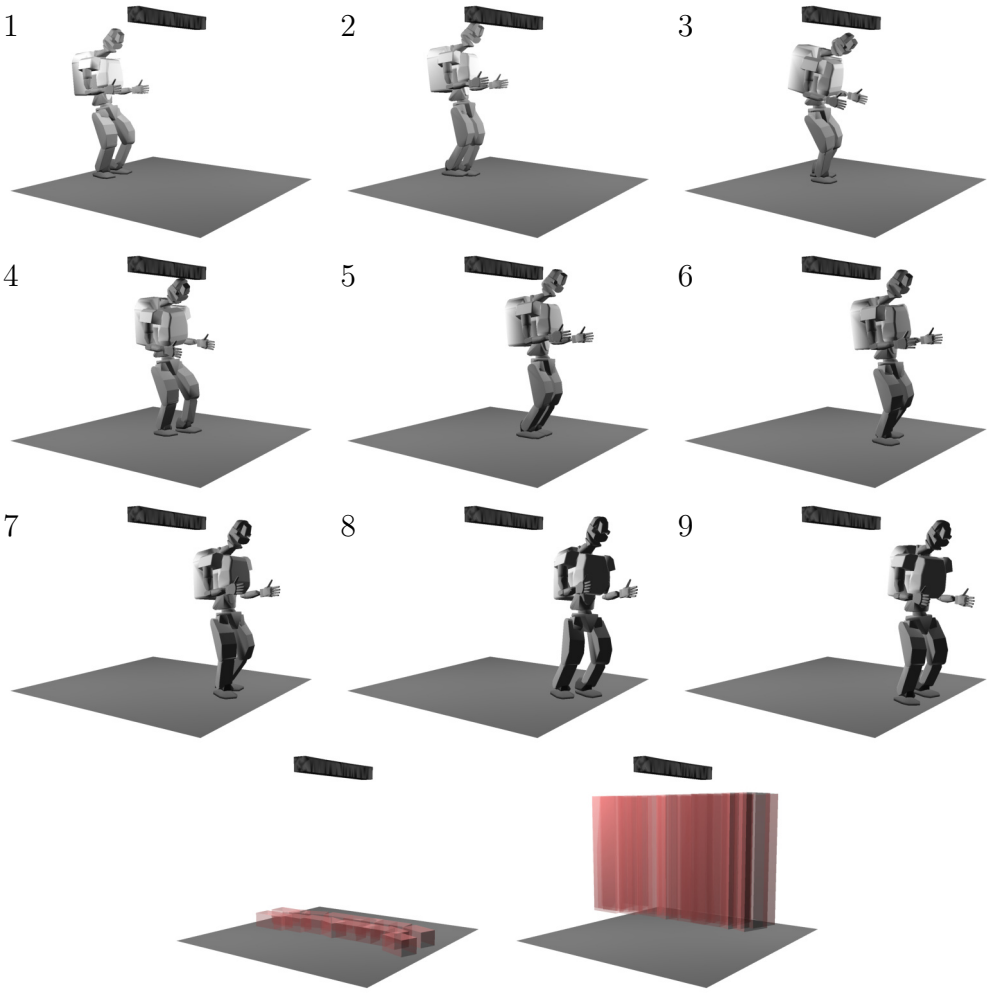


Fig. 2.14 Hierarchical locomotion planning with a high obstacle: walking sequence (1-9), footstep plan, and collision bounding boxes.

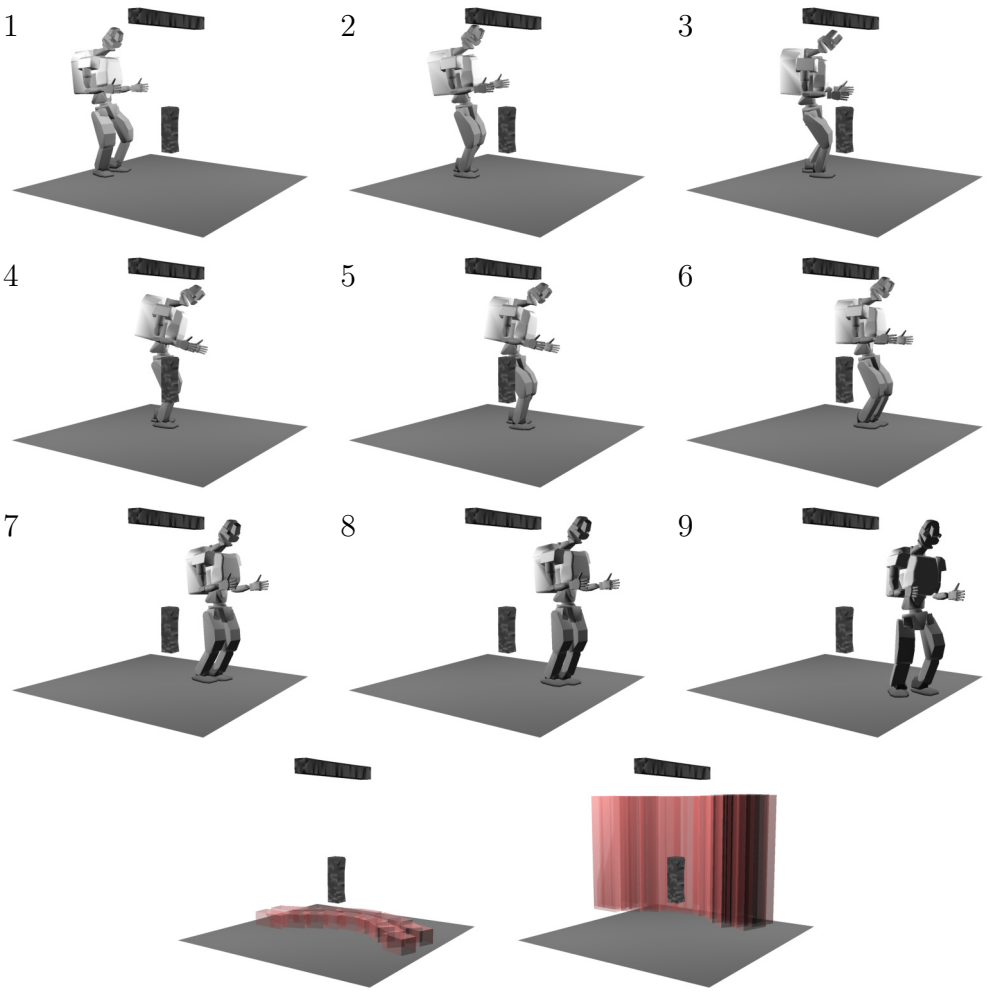


Fig. 2.15 Hierarchical locomotion planning with high and low obstacles: walking sequence (1-9), footstep plan, and collision bounding boxes.

2.6 Discussion

We will organize the discussion of this chapter according to the objectives we set in the beginning.

2.6.1 Applicability of human gait principles

We showed that representations of walking used in human gait literature are tightly related to the footstep planning problem. Furthermore, we showed that energy and RCOF of both human and humanoid walking vary systematically with these representations.

Importantly this chapter shows that planning time variables along with footstep placement, as humans do, is crucial when including ground friction in the problem. The required coefficient of friction (RCOF) for a slip to occur decreases with the decrease of step length and with the increase of double support time, thus allowing the robot to walk on very slippery surfaces by adjusting these variables (as happens with humans [98, 99, 103, 105]). This contrasts to the common practice in humanoid robotics to use constant step times, so investigating human gait here clearly brought some innovations to footstep planning.

It is worthy of note that the models we obtained might differ from the ones obtained with different robots or using different controllers. The extended footstep planning approach is still general, and all that is required to apply it is to learn the E_{COM}, RCOF, Ψ models in simulation with the desired robot and controller.

2.6.2 Energetic advantages of human-inspired models

We showed that COM work is related to electrical energy consumption. According to DC-motor-based electrical power consumption estimates from simulation data, planned paths had close to optimal electrical consumption, and higher COM work lead to higher electrical energy. Our experiments also showed that, at least for our robot and stretched-knees walking controller, minimizing COM work at the footstep planning level leads to low energy consumption on the real robot as well. These observations and the simplicity of the model suggest COM work to be an effective objective function for planning of robot locomotion.

On the other hand, torque minimization also leads to low predicted electrical energy consumption and might be more easily integrated in full-body motion planners, depending on their implementation. While physics simulation of joint torques is noisy and can lead to instabilities on extended footstep planning, that was not the case for our hierarchical motion planner since it does not rely on physics simulation.

2.6.3 Human-like walking behavior

As we showed in this chapter, footstep planning with human-inspired variables and models of energy and slippage leads to human-gait-predicted behavior. In particular, in our experiments we replicate the following observations of human gait:

a) Energy contours are hyperbolic in step length-rate [81, 83],

b) RCOF is reduced on slippery terrain [99],

c) Step length is reduced on slippery terrain [98, 99, 103],

d) Double support time is increased on slippery terrain [105],

e) There is an optimal climbing angle for steep slopes [106, 107]. In other words long steep trajectories will be curved.

2.6.4 Hierarchical motion planning

Our hierarchical planning architecture combines footstep with full-body motion planning, in a way that can deal not only with slippery and slanted terrain, but also complex obstacle placement. Collision-checking is done with a slightly shrunk bounding-box at the footstep planning level, followed by full-body-mesh collision-checking at the joint level. One important detail is that the bounding-box approximation should be chosen such as to approximate the minimum feasible volume occupied by the robot on that stance, in order not to avoid narrow passages. At the same time, too low of a volume might be a bad approximation and so lead to impossible constraints for the full-body optimizer to solve. Care should be taken to assure that bounding-box dimensions are appropriate.

Our footstep planner's computation times were comparable to other state-of-the-art planners even though we plan extra step parameters and

consider energy and friction. Importantly, we compute parameters other than footstep placement from state transitions, which reduces the A* search space and increases search speed. Pre-computing energetic cost for many combinations of footstep placement and μ also allowed for faster search than if (2.4) were to be solved explicitly for each state expansion. Instead we solve it only for the final obtained path, reducing computation speed by one order of magnitude.

Full-body trajectory representation is an important aspect of the algorithm that can be improved. Full-body motion in this paper was interpolated after trajectory optimization at waypoints. While we did this for implementation simplicity, one possible direction of improvement could be to use the spline representation directly in the optimization problem, using constraints at collocation points. In addition to that, dynamics could also be added to the trajectory optimization problem for more versatile motion.

2.7 Summary

In this chapter we showed that footstep planning for humanoid robots, by using simple principles (i.e. COM work, RCOF) and gait representations (i.e. step length, width, height, double support time, swing time and knee flexion) from human gait literature, leads to both human-like walking behavior and low electrical power consumption. Importantly, we showed through several simulation experiments that the footstep planner we proposed here is well suited for challenging outdoor scenarios since it accounts for ground friction and energy consumption. We also proposed an architecture for hierarchical planning which is objective-consistent and considers trajectory costs, collision, stability and friction. Using this hierarchical planner the (simulated) robot could navigate also in environments with complex obstacle geometry, where a combination of footstep planning and full-body motion is necessary to avoid collision (e.g. going around a low obstacle while bending down not to collide with another high obstacle).

The algorithms introduced in this chapter thus work for varied terrain: flat, inclined, with obstacles and slippery. As we stressed in Chapter 1, they are relevant since not only obstacles but also different terrain types abound in the real world, and locomotion choices should take them into account - whether for safety or energetic considerations.

The planners proposed here rely on the assumption that world geometry and coefficient of friction are known. This is of course a strong assumption, especially because it is not obvious whether friction estimation is even possible before contact - and how high its uncertainty is. Answering these questions is going to be the purpose of our next chapter.

Chapter 3

Visual perception of friction

3.1 Introduction

We now turn into the problem of predicting friction properties of surfaces from visual input. This is an important problem in model-based motion planning methods such as the ones we described in the previous chapter: without good estimates of friction the robot may slip, which in turn may cause challenges to controllers and lead to a fall. So the crucial question is how well can algorithms predict friction of a surface from visual sensors before contact. In addition to that, it is important that algorithms provide estimates of the uncertainty of predictions, so that this information can be used by the planning algorithms for robustness. And finally, human performance at the task can provide useful information for our purpose, such as suggesting possible visual features to encode into algorithms, or to understand whether humans can accurately predict friction at all - whether they should teleoperate robots in slippery terrain or not.

The objectives of this chapter are the following:

a) To understand what kind of visual and semantic features best predict human judgements of friction (Section 3.3)

b) To quantify the performance of humans at predicting friction for a robot foot (Section 3.4)

c) To quantify the performance of algorithms at predicting friction for a robot foot (Section 3.4)

d) To propose methods that can quickly and densely compute friction from images, as well as integrate and provide estimates of uncertainty (Section 3.5).

3.2 Background

3.2.1 Friction perception in robots

The friction estimation literature in robotics has been mostly focused on its measurement during contact. Examples include COF estimation using specially-designed sensors [129], and material classification through dynamic friction model fitting while stroking surfaces with a robotic finger [130]. On the legged robot locomotion literature, there is an interest in identifying slips when they occur [131–133], in order to trigger changes in controllers [131] or activate reflexes [58]. For example, [131] estimated slipping force by comparing predicted ground reaction forces and those measured with a force sensor on the foot. On the other hand, [132] uses Kalman filtering of IMU measurements to detect slippage, and [133] applies a similar approach to a quadruped robot which considers active contact information as well.

Planning algorithms can prioritize low "friction sensitivity" and prefer robot configurations that are stable even for low COF [89, 92]. But even then it is important to have an estimate of the actual friction and its uncertainty. Otherwise, planned motions may be too conservative and suboptimal, or too aggressive considering reflex controllers' robustness. One option to tackle this problem is through learning from experience. Notably, [20] uses visual terrain classification and slope to estimate friction on a rover. The authors train their models on image sequence datasets of rover navigation.

Terrain classification approaches such as [20] are limited by the size of the datasets: for example if the dataset does not contain "ice" then the algorithm will not be able to accurately predict friction for this class. However, as we will show in this chapter, it is possible to use other sources of knowledge such as text on the internet to help predict friction for classes not present on the dataset. Such an approach to the small-dataset problem is similar to the method used for affordance estimation in [134]. Affordance estimation is the task of automatically classifying which actions can be applied to different objects, which the authors compute using distances between vector representations of words.

Fig. 3.1 The OSA+F dataset (8 materials, 96 images), one example per material category. carpet/rug, concrete, fabric/cloth, granite/marble, metal, stone, tile and wood.

3.2.2 Friction perception in humans

Human performance at the *friction from vision* task has the potential to inform the robotics and computer vision communities of which features to use for prediction. Humans are known to use visual cues to estimate friction, related to surface texture [135], shine [136] and detection of materials or contaminants (e.g. water) [137]. Furthermore, in the human gait literature there is evidence that humans use accumulated previous experience to predict friction and adapt walking style before touching slippery ground, as we discussed in Section 2.2.3.

Still, humans make friction judgement mistakes that lead to slipping for example due to over-reliance on gloss or other lighting-related visual features [136], and they are known to have difficulties in estimating coefficient of friction values [135]. Therefore it is still not clear how well humans estimate friction from vision.

3.3 Friction from vision in humans

3.3.1 Dataset

For the purpose of better understanding (and predicting) human perception of surface friction during locomotion, we created the OSA+F dataset. We

started from the open and crowd-sourced OpenSurfaces dataset [138], along with the texture attribute annotations of [63], referred to as OSA (OpenSurfaces plus texture Attributes). The reason that took us to start from the OSA dataset is the large amount of data available: real-world scenes annotated with object, material, texture, scene and illumination judgements. Each image is annotated with segments drawn by the subjects and each segment is attributed an object name, material class (1 out of 22) and the applicability of texture classes (boolean vector of size 11, e.g. whether the segment's texture is chequered or not, marbled or not, etc). Albedo and reflectance judgements also exist for most segments. We considered the data available with the OSA dataset most suitable for the friction estimation task since human judgements of friction are usually associated with gloss [136], material and texture [135].

We selected a high-quality, class-balanced subset of the OSA dataset appropriate for our task. First, for high-quality annotations, we discarded segments with negative judgement scores. Since our goal is to obtain a dataset for friction estimation of locomotion surfaces, we only considered segments corresponding to traversable planar surfaces. Traversability was manually annotated by the authors. From the high-quality traversable segments we selected 96 segments for the OSA+F dataset. These were obtained by solving a mixed-integer linear program maximizing total segment area, subject to the constraints:

i. each material has exactly 12 occurrences in the dataset,

ii. each texture has at least 10 occurrences in the dataset,

iii. each image has only one segment in the dataset (to prevent similar segments from the same image).

The resulting OSA+F dataset consists of 96 segments, from 96 images, and 8 material classes with 12 occurrences each. We show one example image for each material class in Figure 3.1.

We collected human judgements of friction for each image segment through an online survey with random image order, one image per page, prepared using the Limesurvey software [139]. Subjects were 14 graduate students from the mechanical engineering department with normal or corrected-to-normal visual acuity. Each image segment was judged by the subjects using a slipperiness Likert scale of 1 to 6 (i.e. 1 least slippery, 6 most slippery).

Fig. 3.2 Example image from the OSA+F dataset overlaid with a red square indicating the area of interest.

We opted for this scale after preliminary experiments showing larger scales to be difficult to judge, "slipperiness" to be easier to rate than "friction", and because the same scale is used on different material judgement experiments in the human vision literature [140]. The explanation of the scale was present in all pages. The questions were framed as how slippery the subjects expected the surfaces to be in case they were walking on them with their normal shoes. As a post-processing stage we normalized judgements to a friction scale instead of slipperiness (i.e. $y = 1 - \frac{y_{\text{likert}}}{6}$, thus 0 is lowest friction, $\frac{5}{6}$ highest). On the survey, segments were indicated by a red square overlaid on the image, computed as the largest-area square inside the OSA segment. See Figure 3.2 for an example.

3.3.2 Considered features

To understand which features human judgements of friction best correlate with, we use the annotations provided by the dataset as well as other features based on image statistics.

Fig. 3.3 Example of an intrinsic image decomposition. From left to right: original image, reflectance image, and shading image.

a) Semantic classes

Since the OSA+F dataset provides high-level semantic classes associated with each surface (and therefore each slipperiness estimate), we can use these semantic classes for friction prediction.

Given an input image of material m, we predict friction to be the mean over the training set y on images of the same material:

$$\mathrm{f}_{\mathrm{MatMean}}(m) = \frac{1}{|M|} \sum_{k \in M} y_k, \tag{3.1}$$

where M is the set of images labeled with material m. When the input image material m is not present on the training set, we use an average friction prior $\mathrm{f}_{\mathrm{MatMean}}(m) = \bar{y}$. We apply the same logic for texture and scene label features $\mathrm{f}_{\mathrm{TexMean}}$, $\mathrm{f}_{\mathrm{SceMean}}$.

b) Gloss

Higher gloss surfaces are usually judged by humans (sometimes mistakenly [136]) as more slippery. Inspired by this observation, we use shading as a feature for friction prediction. Intuitively, we can make an algorithm that analyzes the shading of the scene and classifies a surface as more slippery if it has glossy specular reflections, and less slippery if it is more matte.

In computer vision terms, shading can be estimated by an intrinsic image decompostion algorithm. These algorithms decompose an original image I into two layers: a shading layer S (irradiance, illumination) and reflectance layer R (albedo, the surface's color). The layers are estimated such that $I = R \cdot S$. See Figure 3.3 for an example decomposition. Several algorithms exist to estimate this decomposition, such as Retinex [141] or other more complex examples [142]. In this paper we use the Retinex algorithm (implementation

in [143]) due to its order of magnitude faster computation time while still achieving high performance [142]. Given an input image, we run Retinex to obtain its shading image and compute the histogram of shading values over the region of interest to estimate friction in that region.

We use the the maximum and standard deviation of shading as features:

$$\mathrm{f}_{\mathrm{ShadMax}} = \max_{(i,j)\in C}(S_{i,j}), \tag{3.2}$$

$$\mathrm{f}_{\mathrm{ShadStd}} = \sqrt{\frac{1}{N}\sum_{(i,j)\in C}(S_{i,j} - \bar{S})^2}, \tag{3.3}$$

where i, j are indices of the the shading image inside the region of interest C, N is the number of pixels in that region and $\bar{S}$ the region's mean shading. During training, we fit the features to training data using ordinary least squares (OLS) linear regression.

c) Roughness

Humans also use visual estimations of surface roughness to predict friction [135]. Intuitively, frequent variations in image intensity can be used to predict high surface roughness, which is generally associated with high friction. We compute the magnitude of the image gradient with a Sobel filter and use the average magnitude of the response as a feature:

$$\mathrm{f}_{\mathrm{GradMu}} = \frac{1}{N}\sum_{(i,j)\in C}\|\nabla I_{i,j}\|, \tag{3.4}$$

where i, j are indices of the image inside the region of interest C and N the number of pixels in that region. During training, we fit the features to training data using OLS linear regression.

3.3.3 Results: predicting human judgements

We now analyze the data collected and friction prediction results. We use two metrics for algorithm evaluation:

i. Root Mean Squared Error (RMSE) between real and predicted friction values on the test set. Results reported are 2-, 5- and 10-fold cross validation values of the RMSE (i.e. average RMSE over the 2, 5 and 10

test sets respectively). All dataset splits are provided together with the datasets.

ii. Pearson correlation significance ($p < 0.05$ or $p < 0.01$) between real and predicted friction values on the whole dataset. We use this metric to estimate how chance could be responsible for the correlation between algorithms' predictions and real friction. Due to the relatively small size of the datasets, we choose to report p values on the whole dataset instead of the test sets.

We computed the average and standard deviation of friction judgements for each material, texture and scene. Figure 3.4 shows the results. Qualitatively, friction variability within materials is smaller than within texture label or scene context. According to a 2-way ANOVA, several relationships between materials are statistically significant: carpet's friction estimates are higher than all other classes; and concrete, fabric, metal and stone are all higher than granite, tile or wood. In the case of textures, the only significant difference is between the labels grid and paisley. Scenes are also poorly informative in this dataset: the only significant difference is between bedroom and foyer. These results indicate material to be a better candidate for prediction of human judgements of friction.

One recurrent observation in human perception literature is the reliance of humans on gloss to estimate friction [136]. We test this hypothesis on the dataset by computing the Spearman correlation between friction judgements and gloss/shine estimates as given by the original OpenSurfaces dataset. The Spearman correlation coefficient is $r = -0.344$ ($p < 0.01$), which indicates a significant relationship between the two. However, when computing the correlation independently for each material class, we found that gloss only correlates significantly with friction judgements for the material granite/marble $r = -0.781$ ($p < 0.01$).

Figure 3.5 shows 8 images of the dataset sorted from highest to lowest mean human friction judgement. For each image we also show data used for image-based features: the gradient image, the shading image and the histogram of values in the shading image. Interestingly, we note that floors with strongly specular reflexions (i.e. higher gloss) are considered the most slippery of the whole dataset, which can be observed in the shading image by larger mean and maximum shading values. The figure also shows that simple single features such as gradient or gloss are insufficient to predict

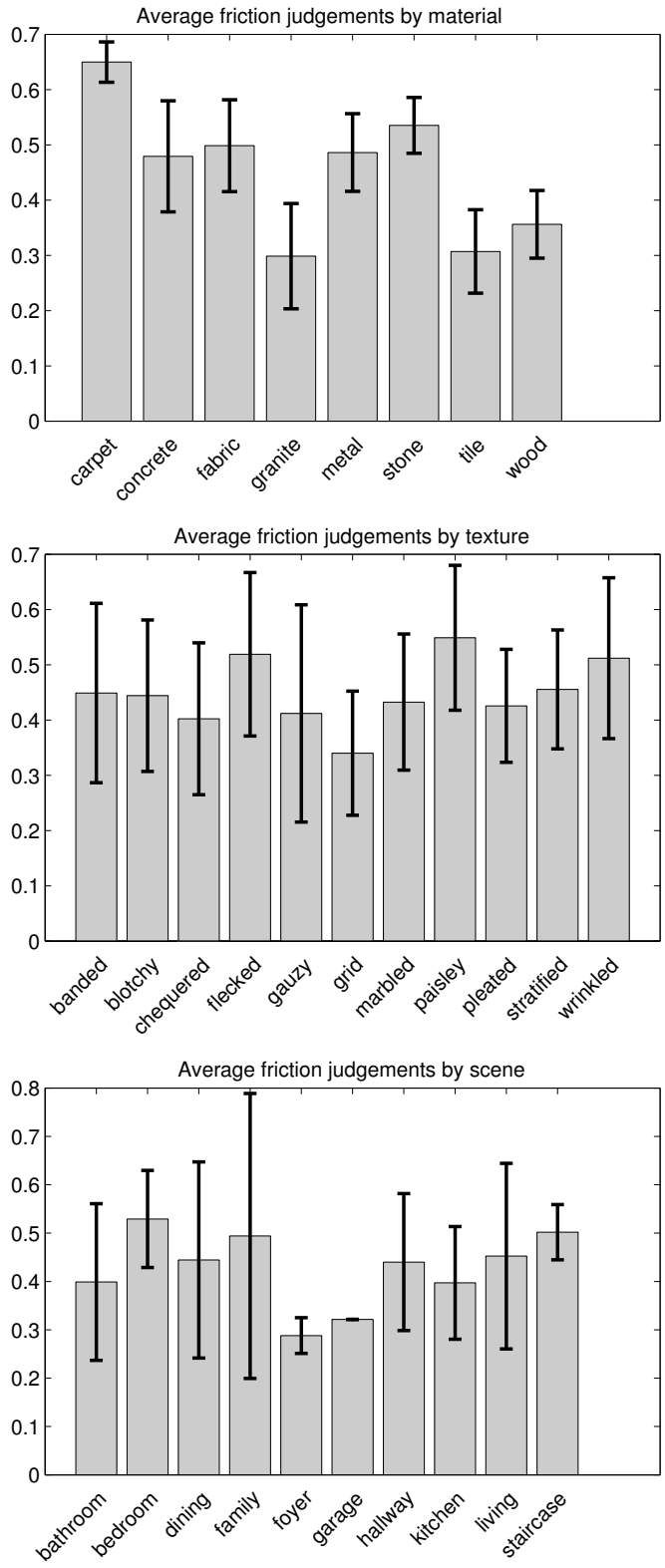


Fig. 3.4 Average and standard deviation of friction judgements for each material, texture and scene label on the OSA+F dataset.

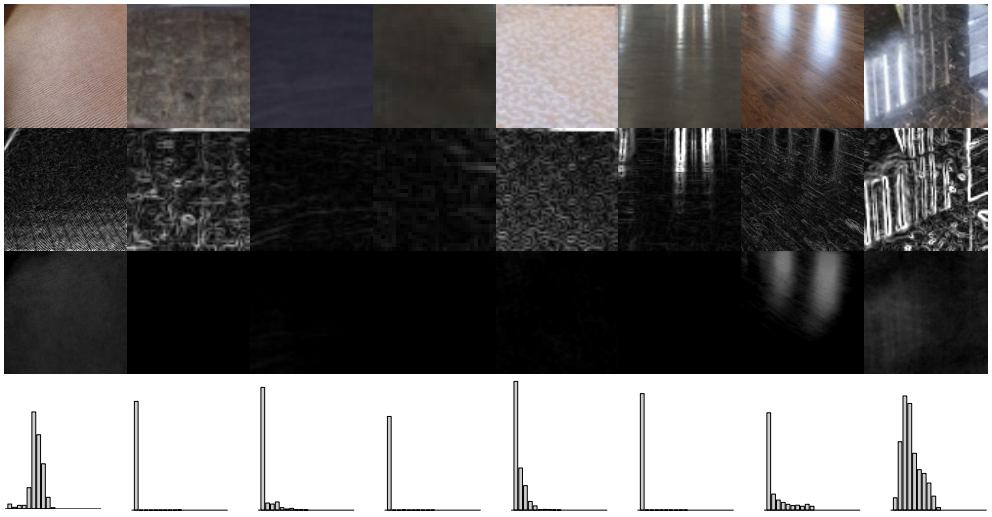


Fig. 3.5 Eight images from OSA+F sorted from highest to lowest average friction judgements. From top to bottom: original image, gradient image, shading image, histogram of shading image.

human judgements of friction. For example, the surface with most perceived friction (a carpet) is according to our simple "maximum shading" feature very slippery due to what looks like a glow in the surface.

Next, we used the mentioned features to predict y: the average human friction judgement for each image. In Table 3.1 we show the prediction errors. We show both 10-, 5- and 2-fold cross validated RMSE values, the algorithms' rank according to the average of the previous three values, and significance of Pearson correlation. We use the following two baselines for better comparison and interpretation of the results:

i. "Constant friction" baseline: the mean of y over the training set is used as the prediction;

ii. "Single subject" baseline: we use a single subject's friction estimates as the prediction. We do this for each subject as a predictor and then average the results over all subjects. The objective is to measure performance of a single human in accomplishing the same task as the algorithms (i.e. estimate the average person's friction judgement).

The single subject baseline achieved 0.104 RMSE on 5-fold cross validated results, which was slightly lower than the constant friction baseline (0.137) but indicates high variability among subjects. In fact, inter-subject variability is high (σ =0.166, or 41% of the mean). The best performing algorithms were MatMean and WordMM, which scored 0.083 RMSE. This

Table 3.1 OSA+F: predicting mean human data

Features	$\text{RMSE}_{\text{CV10}}$	RMSE_{CV5}	RMSE_{CV2}	p	AvgRank
Const	0.137	0.137	0.140		2
SingleSubj	**0.103**	**0.104**	**0.104**	*	1
HumanGloss	0.131	0.129	0.132	*	5
GradMu	0.137	0.138	0.142		8
ShadStd	0.125	0.125	0.129	*	3
ShadMax	0.128	0.128	0.131	*	4
TexMean	0.136	0.137	0.140	*	6
SceMean	0.142	0.134	0.158	*	7
MatMean	**0.081**	**0.083**	**0.086**	*	1

Note: $p < 0.05$ is marked with *, $p < 0.01$ with **. TexMean, SceMean and MatMean use ground-truth semantic labels (i.e. of texture, scene, material).

result matches the previously stated observation that in this dataset material is highly discriminatory.

Interestingly, human judgements of gloss as provided by the original OpenSurfaces dataset scored 0.129 RMSE. The simple statistics of shading images we developed, ShadStd and ShadMax, had a similar but slightly lower error (0.125 and 0.128). All other features either performed at constant baseline level or did not have significant correlation with the mean human judgements. In general, features had similar performance at the different cross validation ratios.

3.4 Friction from vision for robot locomotion

3.4.1 Dataset

We now focus on quantifying the performance of different features, as well as human teleoperator judgements, at the task of predicting coefficient of friction from images for robot locomotion. Importantly, the coefficient of friction depends on properties of both surfaces in contact, and thus the main objective of building this dataset is not to train predictors applicable to all robots, but to quantify humans' and algorithms' performance at the task. Our assumption is that the conclusions taken from our robot foot's

Fig. 3.6 The GTF dataset (14 materials, 43 images), one example per material category. asphalt, brick, carpet/rug, cobble, concrete, dirt, granite/marble, leaves, linoleum, metal, mud, stone, tile and wood.

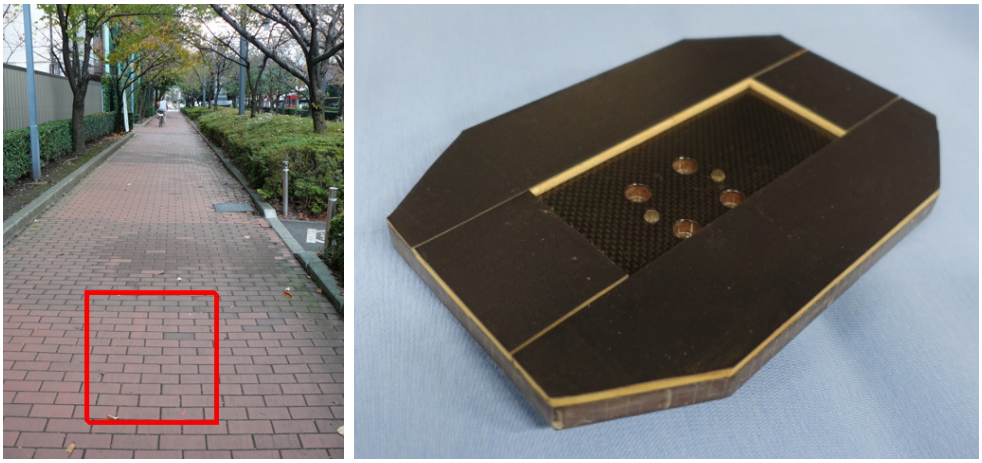


Fig. 3.7 Left: example image from the GTF dataset overlaid with a red square indicating where the coefficient of friction was measured. Right: sole of the humanoid robot foot used for the coefficient of friction experiments, GTF dataset.

data may generalize to different robot feet as well.

The dataset consists of 43 mostly outdoors images. These are annotated with material class, ground-truth coefficient of friction measured on a humanoid robot foot, and human judgements of friction similar to those in OSA+F. We show the human-sized humanoid robot foot we used in Figure 3.7. The foot is rigid and its sole is covered with a high-stiffness soft material for shock absorption and an anti-slippage sheet. Locations of the dataset images were chosen such as to cover the same material classes as in OSA+F, as well as extra "dirt", "mud" and "leaves" classes which are common outdoors. At each location, we first measured the maximum friction force by pulling the foot with a spring-scale until it started moving for around 10 trials. We recorded the static coefficient of friction value as the average of the trials divided by the foot's weight. The standard deviation of COF measurements over trials was on average $\sigma = 0.047$. The foot was loaded with a 1.5Kg mass and surfaces were checked to be horizontal with a spirit level device. After measuring the coefficient of friction, we removed the foot from the locomotion surface and took pictures of the surface and surroundings using a consumer level camera, along with an annotation of the image location where friction was measured. See Figure 3.7 for an example picture.

Human judgements of friction were collected as well, using the same procedure as in OSA+F. However, all subjects were given the actual robot foot

to look at, feel and experiment on their tables before taking the survey (all subjects' tables were of the same material). The questions were framed as how slippery the subjects expected the surfaces to be in case they were walking on them while wearing the robot's feet as shoes. The subjects responding to this survey were 12 of those who also participated in the OSA+F survey. The dataset contains images of asphalt (3), brick (3), carpet/rug (5), cobble (1), concrete (3), dirt (4), granite/marble (3), leaves (1), linoleum (2), metal (8), mud (1), stone (1), tile (6) and wood (2). We show one example image for each class in Figure 3.6. Unlike the OSA+F dataset, GTF is not class-balanced. Some material classes are under-sampled, which creates difficulties in training-based algorithms using material class as a feature. We will now propose a solution to deal with such difficulties: friction prediction without training examples using material class prediction and text mining.

3.4.2 Semantic features and text mining

When predicting friction from a surface of a semantic class which has not been observed before, one can assume a rough prior such as the one we proposed in the previous section: average friction over the training set. However such a method will unnecessarily make very wrong predictions. For example, if an image-based classifier predicts a surface to be of the material "asphalt" but the friction training set consists only of COF measurements for "concrete" and "ice", the average of the two COF is probably much lower than that of asphalt even though it is intuitively more similar to concrete. We argue that to solve this problem we can use text mining.

Text mining methods such as LSA [144] or word embeddings [145, 146] have been used to obtain affordance relations [134] and various other semantic relations [145]. In the case of this paper we are interested in material-material relations such as "asphalt is similar to concrete", and material-slipperiness relations such as "asphalt co-occurs with the word slippery often". We explore both these kinds of relations in this paper through the use of word embeddings.

Word embedding algorithms, such as Word2vec [145] or GloVe [146], embed words into semantic vectors. Each word is represented by a vector of usually 50 to 1000 dimensions, and the cosine similarity between words

$$c_{i,j} = \frac{w_i \cdot w_j}{\|w_i\|\|w_j\|}, \tag{3.5}$$

is proportional to their co-occurrence in the training set. Here w_i is the vector representing word i. Using the previous example, we can thus estimate the co-occurrence of "asphalt" with "concrete", or even "asphalt" with "slippery" by simple internal products to estimate how similar the two materials are, or how slippery asphalt is. For the results in this thesis we trained Word2vec and GloVe models on the complete Wikipedia article dump of 20080103. We chose algorithm parameters by varying them within the ranges recommended in the respective publications, such as to optimize model performance on the semantic tasks described in [146]. Final parameters common to both algorithms were: vector dimension 400 and window size 10. Word2vec-only parameters were: CBOW architecture, negative sampling 10, frequent word sub-sampling 10^{-5}.

After word embeddings are trained, we use semantic similarity queries to estimate friction of an input material. Since the word embeddings exist for all words on the text corpus, we can theoretically estimate friction for thousands of classes. We propose two algorithms for estimating friction using word embeddings.

a) Material-Material similarity

For materials present in the training set this method is the same as the semantic-class method described in Section 3.3.2. However, when the input image material m is not present on the training set, we use the friction of the "most similar material" $\hat{m}$ in the training set:

$$\mathrm{f}_{\mathrm{WordMM}}(m) = \frac{\bar{y} + \mathrm{f}_{\mathrm{MatMean}}(\hat{m})}{2}. \tag{3.6}$$

$$\hat{m} = \arg\max_{j} c_{m,j}, \tag{3.7}$$

We average $\mathrm{f}_{\mathrm{MatMean}}(\hat{m})$ with the friction prior $\bar{y}$ in order to attenuate errors due to possible wrong material associations.

b) Material-Slipperiness similarity

In this method we estimate friction by word-similarity between the queried material name and a list L of slipperiness-related words[1]. The intuition

[1]The full list of slipperiness-related words we use is: slipped, slipping, skid, slue, slew, slide, skidded, slued, slided, skidding, slueing, sliding, lubricious, nonstick, slick, slimed,

behind this approach is that the more often a material co-occurs with words such as “slip”, “slipped”, “slippery” in text then the more likely it is to be slippery for the average contact material. The advantage of the method is that no friction measurements have to be made in order to rank materials by predicted friction, which might be sufficient for some robotic applications (e.g. always plan paths through least slippery options). In this paper we still linearly fit the function to training data, just like with the rest of the features. The feature we propose is the maximum similarity between an input material m and the slipperiness words in list L:

$$\mathrm{f}_{\mathrm{WordMS}}(m) = \max_{j \in L} c_{m,j}. \tag{3.8}$$

3.4.3 Results: predicting COF

We did the same analysis as in the OSA+F dataset with GTF data, now targeted at robot locomotion. Figure 3.8 shows the average and standard deviation of friction judgements and real COF for each material. According to a 2-way ANOVA, linoleum had significantly higher friction than all other materials except mud and stone. Wood, asphalt and mud were significantly larger than dirt and leaves. Interestingly, the high COF of mud was not predicted by most human judgements, because contrary to some subjects' intuition mud was sticky rather than slippery. Finally, carpet COF is only significantly higher than dirt.

We also show 8 images of the dataset sorted from lowest to highest COF in Figure 3.9. We can see how this dataset is more challenging than OSA+F. Surfaces with least friction now include both leaf-covered and wet concrete. The intrinsic image decomposition does not detect specular reflections on the wet case, leaves lead to what could naively look like a surface of high roughness (when in fact leaves can slide easily), glossy wood is actually not slippery for the robot foot because of its anti-slippage sheet, etc. Such examples indicate once again that material classification might be the safest option to friction estimation, although detection of surface “contamination” is crucial as well (e.g. of water, oil, leaves, grain, dust). In fact, one main observation we made during the collection of this dataset was that since the foot is flat, smooth and rigid, its COF is the lowest on contaminated

slimy, slithering, slithery. They were obtained by searching and conjugating words related to the word slip on WordNet [147].

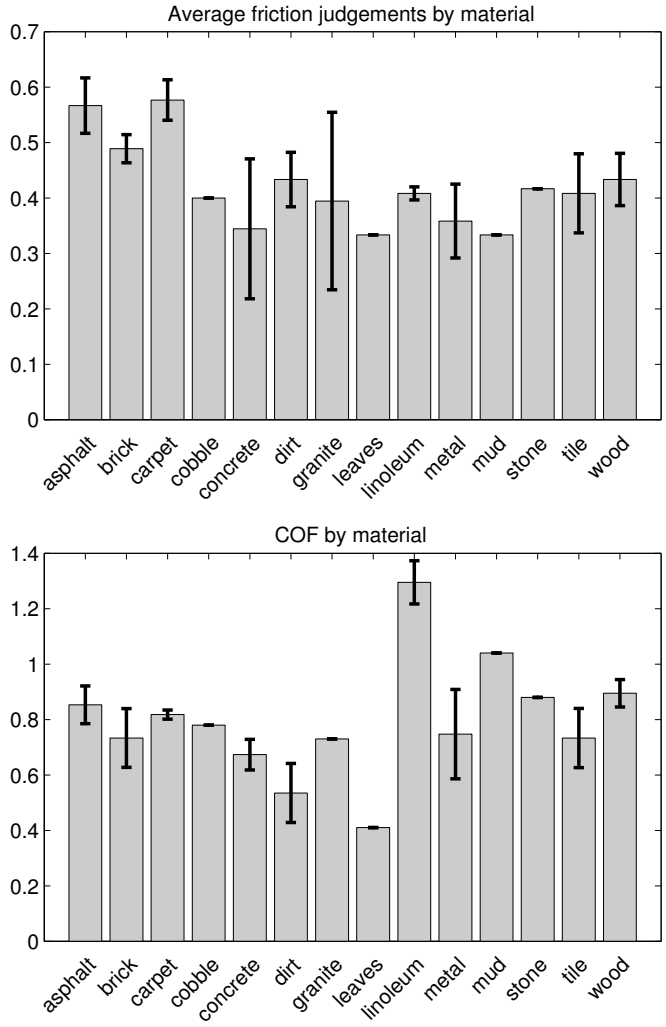


Fig. 3.8 Average and standard deviation of COF for each material on the GTF dataset.

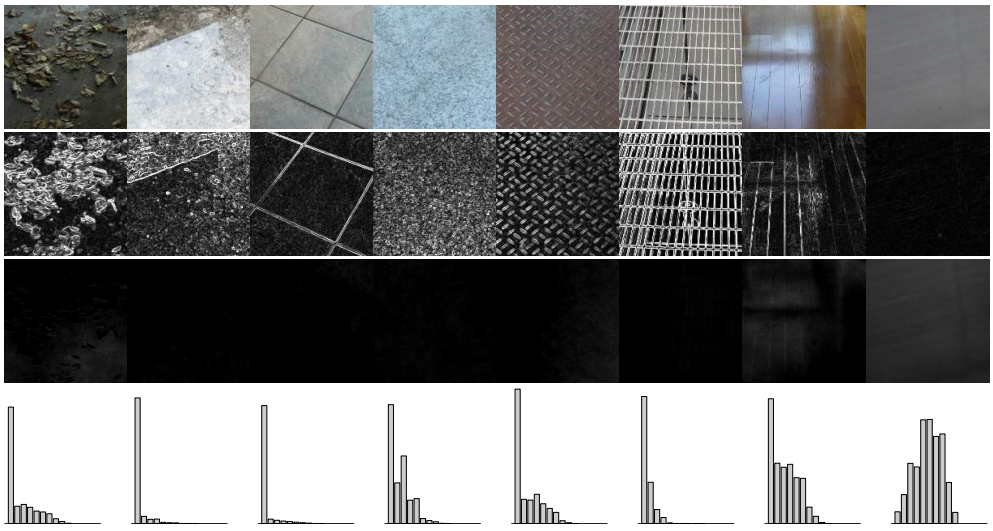


Fig. 3.9 Eight images from GTF sorted from lowest to highest COF. From top to bottom: original image, gradient image, shading image, histogram of shading image.

surfaces: clean marble had high friction, dusty was low; small 1mm^2 stones, leaves, or water drastically reduced the COF.

Next, we used the visual, rough semantic, and text-mined semantic features to predict y: the real COF. In Table 3.2 we show the algorithm evaluation results. The results shown in this table were obtained assuming ground-truth material is known. We use the following three baselines for better comparison and interpretation of the results:

i. "Constant friction", as in the previous section,

ii. "Single subject", as in the previous section. Note that Features are human judgements of friction, while the target function y is the real COF. The motivation for evaluating this metric is to find out whether an average inexperienced robot operator, even if familiarized with the robot's foot, can predict or not the friction coefficient between the robot and ground. This has of course strong implications for the design of control architectures and interfaces for remotely controlled robots.

iii. "Mean Subject", the average of the subjects' friction judgements. Therefore, we measure how much a group of inexperienced robot operators, instead of a single operator, can help predict COF. The motivation is to compare this metric with the single subject metric, thus helping to understand whether an increase in robot operators (e.g. crowd-sourced

Table 3.2 GTF: predicting COF

Features	$RMSE_{CV10}$	$RMSE_{CV5}$	$RMSE_{CV2}$	p	AvgRank
Const	0.188	0.194	**0.182**		3
SingleSubj	0.176	0.187	0.189		2
MeanSubj	**0.174**	**0.186**	0.187		1
GradMu	0.172	0.180	0.176		5
ShadStd	0.177	0.191	0.196		6
ShadMax	0.171	0.187	0.182	**	4
MatMean	0.130	0.137	**0.134**	*	1
WordMM	**0.127**	**0.135**	0.141	*	2
WordMS	0.155	0.170	0.180	**	3

Note: $p < 0.05$ is marked with *, $p < 0.01$ with **.

operators) may increase prediction performance.

The constant friction baseline in this dataset achieves 0.194 RMSE on 5-fold cross validate results. Perhaps surprisingly, single subject judgements of friction achieve performance roughly equal to constant baseline, meaning they are poorly predictive of real COF in this dataset. Also, using multiple subjects (MeanSubj) did not improve performance considerably when compared to the average result obtained with a single subject. Image features (GradMu, ShadStd, ShadMax) were roughly as predictive as human judgements, actually up to 6 % better. However, the image features' correlation with real COF was only significant for ShadMax, which was also a good predictor in the human data of the OSA+F dataset.

Once again material classification, MatMean, was the highest scoring method, achieving 0.137 RMSE on 5-fold cross validation. WordMS further improves performance by around 2% since it deals with classes unseen on the training set. Interestingly, our material-slipperiness word similarity method WordMS achieved higher (and statistically significant) performance when compared to both human judgements and image features. Results shown in Table 3.2 for WordMM and WordMS were obtained using the Word2vec algorithm for word vector training. We also evaluated performance on a different word embedding algorithm, GloVe [146], which is together with Word2vec currently one of the best performing on semantic tasks [148]. On average, the RMSE on GloVe-trained vectors was 3% higher.

3.4.4 Text mining to predict human judgements?

When considering a large number of materials, the method based on word embeddings that we propose can also predict human judgements of friction. To prove this we conducted one further experiment where we asked 19 new subjects to rank a list of 19 different materials[2] from most to least slippery. The question included only the names of the materials and no supporting images. We computed the average ranking of materials over the subjects and compared this average with the word similarity score given by WordMS (3.8). The Spearman correlation between the human rankings and $\mathrm{f}_{\mathrm{WordMS}}$ was a low but significant $r = 0.4607$ ($p < 0.05$). Word embeddings trained on Wikipedia thus seem to encode some knowledge of human judgements of friction, even though at a low correlation level. The same procedure applied to the class-averaged friction values of the OSA+F dataset, perhaps due to the low number of materials which was 8, does not lead to a significant correlation between word embeddings and human judgements of friction.

3.5 Fast, dense, large-scale friction from vision

In the previous sections we quantified the error of different visual features and the (best-case) error of semantic classes at predicting friction for robot locomotion. Best-case because semantic classes were provided as ground-truth and presumably uncorrupted by noise. While material class was one of the most predictive features, its estimation from vision is a difficult problem in itself. Also, we did not yet formally provide a way to predict friction densely (i.e. for each image pixel), to estimate the uncertainty of predictions and to do all of this quickly for a robot application in practice. This will be the objective of this section.

The basic idea will be to do pixel-wise material classification from images with a fast algorithm based on convolutional neural networks and then estimate friction from materials. Assuming the distribution of friction for each

[2]The complete list was: asphalt, brick, cardboard, carpet/rug, ceramic tile, concrete, fabric/cloth, glass, grass, ice, leather, linoleum, marble/granite, metal, mud, plastic, puddle on asphalt, stone, wood. The subjects were told that with the exception of ice, mud and puddle all materials were dry. Like the original OSA+F task, we also told the subjects to make their judgements assuming they are walking with their normal shoes.

material is known (or learned through robot locomotion experience), then the uncertainty in material friction can be used together with material classification uncertainty to give a final estimate of friction uncertainty. Such probability distributions of material friction can be obtained by measuring COF directly on the robot foot in many different surfaces and materials on an initial stage (before making the robot walk). Although that is our approach in this thesis, they could alternatively be estimated by automated learning during robot locomotion, or a combination of both.

3.5.1 Material CNNs with friction distributions

We propose to estimate friction densely from visual input by classifying surface material at each image pixel and assuming known (or learned) probability distributions of friction for each material. For convenience we will use the term "friction of a material" to refer to the coefficient of friction between the robot foot sole and a second surface of a given material.

We consider a pixel-wise labelling algorithm that, given an input image I with n pixels, provides a probability distribution $P(X|\theta, I)$, where $X = \{x_1, ..., x_n\}$ are the pixel labels and θ are internal parameters of the algorithm. Each pixel can take one of m possible labels, such that $x_k \in \mathcal{L} = \{l_1, ..., l_m\}$. Furthermore, let each label be a material associated with a probability distribution function (p.d.f.) of a coefficient of friction $p(\mu|l_i)$. Then the conditional p.d.f. of $\mu^{(k)}$ (the coefficient of friction at pixel k) is

$$p(\mu^{(k)}|\theta, I) = \sum_{i=1}^{m} p(\mu|l_i) P(x_k = l_i|\theta, I). \tag{3.9}$$

For the results we will show in here, we estimated the friction distributions $p(\mu|l_i)$ experimentally, by measuring maximum friction force of the robot foot on several surfaces for each material. We will describe the procedure in more detail in the next section.

We use a deep convolutional neural network (CNN) to obtain pixel-wise material predictions $P(x_k = l_i|\theta, I)$. In particular we use the encoder-decoder architecture of [149], which achieves good results in image segmentation applications and is characterized by a low number of parameters. Its low number of parameters leads to fast inference, which is crucial for robotics. The architecture consists of an encoder network of 13 convolutional layers as in VGG16 [150], followed by a decoder network of 13 layers and a final

softmax layer. The output of the last layer of the network (a softmax classifier) is at each pixel a vector of probabilities for each class, that is, the probabilities $P(x_k = l_i|\theta, I)$ used in equation (3.9).

3.5.2 Results

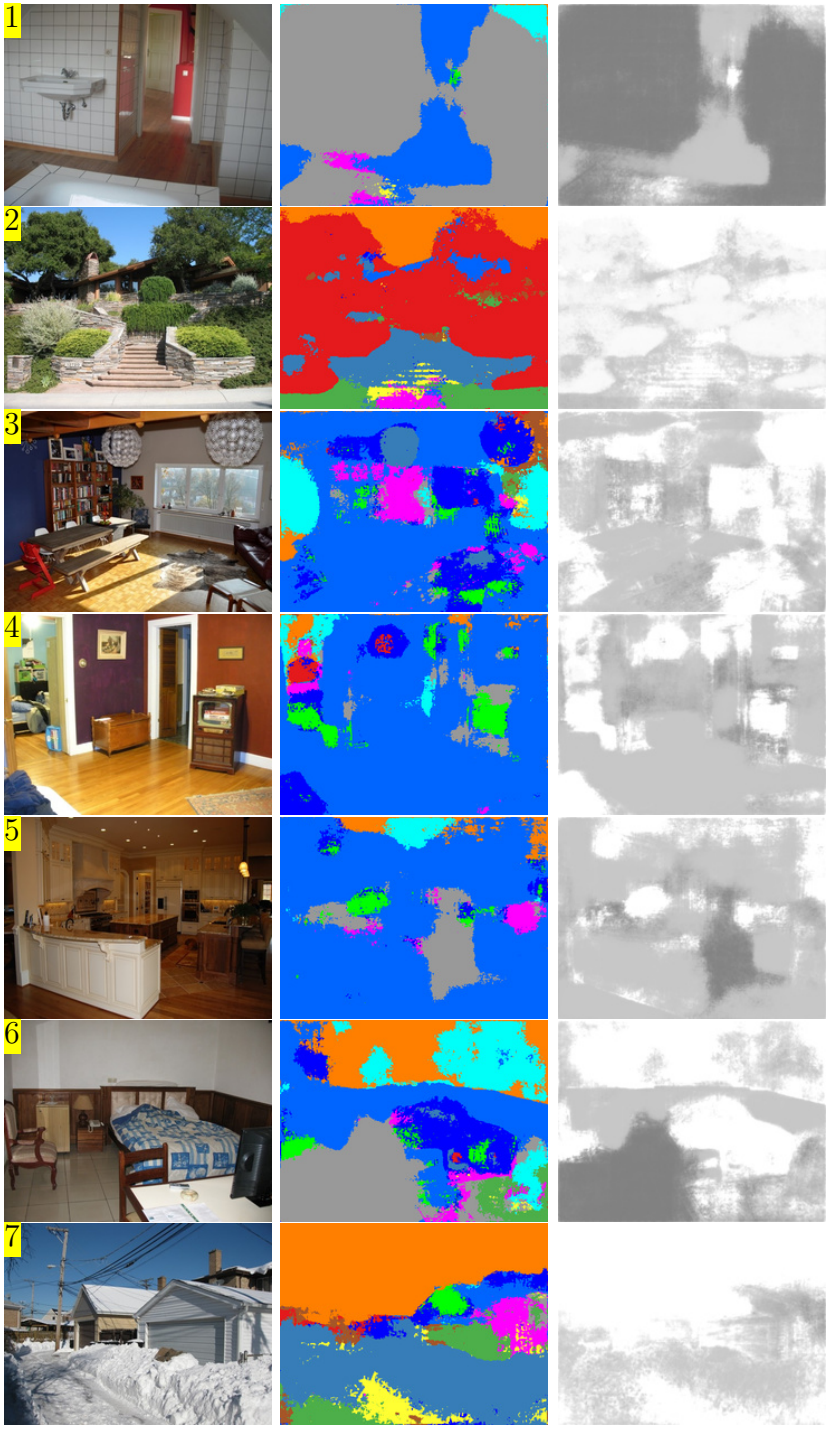
a) Material recognition

To train the CNN we first collected 7,791 annotated images from publicly available semantic-segmentation datasets: 5,216 from the VOC2010 Context dataset [151] and 2,575 from the OpenSurfaces dataset [138]. We selected all images in the datasets with at least one of the following labels: *asphalt, concrete, road, grass, rock, sand, sky, snow, water, carpet, rug, mat, ceramic, tile, cloth, fabric, marble, metal, paper, tissue, cardboard, wood*. Due to similarity between some classes at the image and semantic level we joined the labels (*asphalt, concrete, road*), (*carpet, rug, mat*), (*ceramic, tile*), (*cloth, fabric*) and (*paper, tissue, cardboard*). The total number of considered classes in the output CNN layer was 14. *Sky* was only included to avoid classifying it as any of the other materials on outdoor pictures.

We used stochastic gradient descent with 0.1 learning rate and 0.9 momentum as in the original SegNet publication [149], and trained the network on an Amazon Elastic Cloud node with a 4GB NVIDIA GPU. We ran a total of 90,000 iterations with a mini-batch size of 5 (maximum allowed by the GPU). Training was done on 60% of the images, while the other 40% were used as the test set.

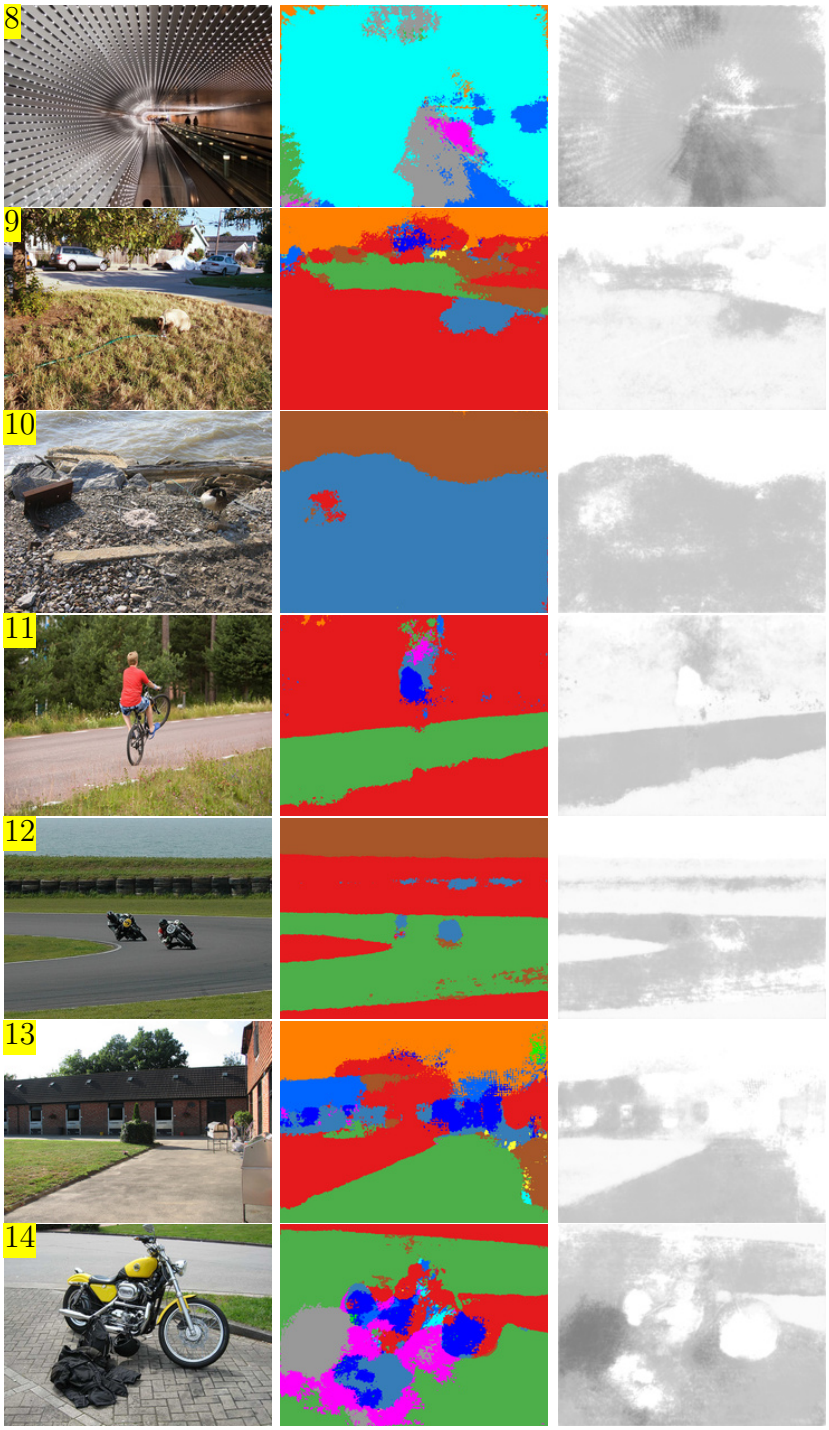
We obtained a global classification accuracy of 0.7929 and class-average accuracy of 0.4776 on the test set. See Figure 3.10 for examples of the (highest probability) material predictions given by the CNN on the test set. The global accuracy is comparable to state-of-the-art performance in semantic segmentation (e.g. [62, 149]), and the class-average accuracy is slightly below state-of-the-art (which is around 0.60 [149]). We believe one important way to improve classification accuracy is to improve the dataset itself since, for instance, there is moderate visual similarity between some of the materials such as marble and ceramic, and some materials are lowly sampled (e.g. the lowest sampled materials are snow and sand, present in 173 and 46 images respectively).

The material segmentation results in Figure 3.10 show an overall good



Friction images are quantiles $Q_{1-0.95}$ of equation (3.9). Darker shades of gray correspond to higher friction, such that white is $\mu = 0$ and black $\mu = 1$.

Fig. 3.10 Example material and friction predictions from the test set



Friction images are quantiles $Q_{1-0.95}$ of equation (3.9). Darker shades of gray correspond to higher friction, such that white is $\mu = 0$ and black $\mu = 1$.

Fig. 3.10 Example material and friction predictions from the test set (continued)

accuracy of the CNN, particularly on wood, grass and sky labels. The figure also shows typical misclassifications such as white walls recognized as sky or metal (picture 6), hard snow as rock (picture 7), and some overlap between asphalt/road, ceramic and marble. These are arguably understandable since material labels themselves semantically overlap. However, our approach to the visual friction estimation problem is such that if there is uncertainty in the material label, then this uncertainty can be used to weight the friction of the surface through material and friction probability distributions.

b) Friction estimation

We empirically measured the coefficient of friction associated with each material label using a force gauge and the robot foot loaded with a 1.5kg mass. The foot is rigid and its sole is covered with a high stiffness soft material for shock absorption and an anti-slippage sheet. We checked whether surfaces were horizontal with a level, then placed the foot and measured maximum friction force values with the force gauge. See Figure 3.11 for an illustration of the procedure. We took 5 friction measurements on each surface, and used at least 3 surfaces of each material. We fitted a normal distribution to the measurements, obtaining separate parameters μ_i and σ_i^2 for each material, where μ_i is the mean friction of material i, and σ_i^2 the variance. The materials *sand, snow, water, cloth, paper* were an exception, and since our robot is currently not capable of walking on them (i.e. fall or damage risk is too high) we directly set them to $\mu_i = 0$, $\sigma_i^2 = 0$. We similarly set *sky*'s friction to zero as well. See Table 3.3 for the parameters of the friction p.d.f. of each material.

In Figure 3.10 we show the test-set's highest probability material predictions along with the $(1-\eta)$-quantile of the coefficient of friction. This quantile is the value of the variable μ at which $(1-\eta)$th of the cumulative distribution function is contained, or in other words, a lower bound of friction. We set a typical value of η =0.95, meaning friction will be higher than the displayed values with probability 0.95. The friction images are darker where friction is higher ($\mu = 1$ would be black). Note that ceramic-like surfaces have high predicted friction (pictures 1, 5, 6); beds and jackets have very low friction (pictures 3, 4, 14); grass patches have lower friction than roads (pictures 2, 9, 11, 12, 13); and that water is mostly white - zero friction - (pictures 10, 12).

Fig. 3.11 Coefficient of friction measurement. We estimate the COF of a material by several measurements of the maximum friction force on the robot foot, loaded with a 1.5kg mass.

Table 3.3 Normal distribution parameters of each material's coefficient of friction, measured manually on the robot foot

Material	μ_i	σ_i
Asphalt	0.74	0.12
Grass	0.53	0.10
Rock	0.80	0.08
Carpet	0.82	0.02
Ceramic	0.97	0.05
Marble	0.83	0.15
Metal	0.80	0.15
Wood	0.88	0.12
Sand, Sky, Snow, Water, Cloth, Paper*	0	0

Note: materials marked with a * are assumed to be untraversable by our robot and so set to zero without measurements.

The figure also shows the advantage of using the whole probability distribution of materials (instead of using the highest probability material) to estimate friction. For example in picture 14, the jacket on the ground is classified as cloth and rock depending on the region, but friction is low on most of the object's area since the *cloth* label still has high probability.

3.6 Discussion

We will organize the discussion of this chapter according to the objectives we set in the beginning.

3.6.1 Features used by humans

We replicated recent results in the human perception literature, correlating human judgements of friction and surface gloss/shine [136]. However, we found that this correlation was only significant for the marble material (but not, for example, for tiles). We hypothesize that humans rely on illumination-based features only for certain materials for which it might be predictive. The dataset we designed for this purpose, OSA+F, will hopefully prove useful to the human perception community.

3.6.2 Human performance for teleoperation

Human judgements have low predictive power of COF in the GTF dataset, meaning it might be a wrong choice to trust slipperiness judgement to inexperienced robot operators even if they are familiarized with the robot's foot. We can also imagine a robot operation setup where several perception decisions are crowd-sourced over a group of operators. However, even using the mean of 12 subjects as a predictor leads to lower performance than image-based statistics. Constant-friction baselines might actually be safer than human guesses according to 2-fold cross validated results. The observation matches recent findings in the human literature [135] where COF was difficult to estimate for humans. Our proposed image-based feature related to gloss, the maximum image shading, obtained better, significantly correlated, performance than humans. While friction prediction based on material class was the best performing method, the material classification task is still challenging for state-of-the-art computer vision algorithms (70%

accuracy [62, 63]). Thus, one way a robot teleoperator could assist the procedure could actually be by material labeling.

3.6.3 Algorithmic performance

Material was the most predictive feature for both COF (0.130 RMSE) and human judgements of friction. Image features based on intrinsic shading images performed worse (0.171 RMSE) but slightly better than baseline. Both in this paper and others relying on material classification for predicting friction (e.g. [20]), problems may arise when new materials are traversed. Thus, we proposed methods based on text mining for friction estimation of previously unseen material classes. Matching new materials to trained ones by material-material similarity improved performance by 2%. Estimating friction of a material by the co-occurrence of the material with slipperiness-related words in text was better (0.155 RMSE) than image-based statistics and human subjects at COF-prediction.

We showed that algorithms based on text mining may compensate for lack of robot experience in novel scenarios. These are also likely to improve their performance as Natural Language Processing algorithms improve. An interesting open problem is to find ways to adapt the methods based on text mining we proposed here. One important improvement would be to estimate friction between two specific materials. As proposed here, WordMS estimates friction from co-occurrences between material and slipperiness-related words. Therefore, it obtains not an estimate of friction between two specific materials, but an average estimate of friction of the reference material with all materials which co-occur with it in text.

3.6.4 Fast, dense friction and its uncertainty

We empirically showed that friction estimates in our CNN-based algorithm are more consistent with object/material borders than the highest-probability material label segmentation, which shows a good integration of segmentation uncertainty into friction estimation. We also showed that the algorithms work for varied terrain.

In this thesis we opted to decouple the problem into material segmentation and per-material friction distributions. Even though we obtained the material friction distributions manually, these could also be learned over time

with locomotion experience. Alternatively, friction could also be learned from images directly by end-to-end training, for example by initialization of a CNN with the parameters obtained with our architecture.

Problems might occur when a material the robot cannot walk on, such as water in our case, is mistakenly given very high confidence - very low friction could lead to the subsequent planning algorithms not finding a feasible path to the goal. Our view is that the solution could be semi-supervision where a teleoperator can correct a segmented region's material label. A related problem is that of using the normal distribution to model coefficients of friction. The normal's long tail extends to negative values, which contradicts the definition of coefficient of friction. That fact also leads to lower (i.e. more conservative) friction quantiles than if a bound-respecting distribution was used. This further reinforces the problem of path feasibility when untraversable materials are mistakenly given high confidence. To alleviate this problem, different distributions could be experimented with - in particular the generalized extreme value distribution has been shown to more faithfully model measurements of coefficient of friction [152].

For the context of this paper all surfaces were dry. Wet surfaces could also be included, although from our experience they should be treated as separate material labels (e.g. "dry metal" and "wet metal") so that the distribution $\mu|l_i$ does not become bimodal. Thus, one important detail in this work is the notion of material, which should be taken in a broad sense, as a visually distinguishable terrain class.

Compared to [20], we estimate the coefficients of friction instead of slip, thus decoupling the problem from the physical robot. Importantly, our approach allows for sharing material friction data among different robots as long as they have similar foot soles.

While building new, larger, completely robot-acquired datasets would be advantageous for the field and allow the application of methods based on deep neural networks [62, 63], several challenges still lie ahead since autonomous locomotion in varied terrain by complex robots is still an open problem.

3.7 Summary

In this chapter we showed that human judgements of friction correlate with surface material and (for some materials) with gloss. We showed that they are not good (i.e. rough baseline level) at the task of predicting COF of a robot foot and thus such judgements should probably not be relied upon during teleoperation. On the other hand, we measured the error of friction predictions given by algorithms, which was relatively low for material-class features. We then proposed a fast algorithm to estimate friction and its uncertainty densely for each pixel of an image using material CNNs and material friction distributions. We empirically showed that the algorithm works for varied terrain.

Chapter 4

Visual perception of geometry

4.1 Introduction

We have seen how to plan humanoid robot locomotion in a perfect world with known surface friction and geometry (Chapter 2), and then how friction and its uncertainty can be estimated from vision in the real world (Chapter 3). We now need to turn to the problem of estimating world geometry and its uncertainty from vision in the real world too.

As we discussed in Chapter 1, there are several ways to represent geometry - from grids to meshes and others - and several sensors available to measure 3D geometry. Of particular interest to us here are stereo vision sensors, i.e. camera pairs, since we are concerned with humanoid robots. However, uncertainty of stereo matching, which is the method used to recover 3D geometry from image pairs, is still not well understood. Functions used to model stereo matching uncertainty are usually called "stereo confidence measures" and we will introduce and compare them. In addition to that, common world map representations used for fast collision checking such as occupancy grids, which we will focus on here, do not integrate this uncertainty into their algorithms even though there could be advantages in terms of reconstruction performance or robustness.

With these issues in mind, the objectives of this chapter are the following:

a) Compare the performance of different stereo confidence measures at the task of estimating stereo matching uncertainty (Section 4.3)

b) Improve their performance through parameter estimation techniques and a new non-parametric function (Section 4.3)

c) Integrate stereo confidence measure into occupancy grids to improve 3D reconstruction performance over time (Section 4.4).

4.2 Background

4.2.1 Stereo vision

In stereo vision, 3D information (i.e. object positions in the world) is extracted from 2D measurements (i.e. images from two cameras) by a process called stereo matching. For each pixel on one image, a line on the other image is scanned for the matching pixel. Each pixel along this line then corresponds to a 3D point in the world. Cost functions are used to assign a confidence to each match hypothesis and this vector of costs along one line is usually called the cost-curve. When the highest confidence match is chosen for each pixel to obtain 3D points, we say we used a "Winner-take-all" (WTA) matching approach. The result of such procedure is usually represented as a "disparity image", a 2D image where the value of each pixel is the distance in pixels to the matched pixel on the second image; or it could be represented as a "depth image", a 2D image where the value at each pixel is the depth - i.e. the distance in meters from the camera to that point in the world.

4.2.2 Stereo confidence measures

The functions used to compute "cost-curves", usually called stereo confidences measures, are responsible to model the confidence or uncertainty associated with a pixel match. It is clear from the stereo procedure itself that these are of crucial importance for 3D reconstruction performance. These functions are of high interest not only for WTA methods but also for global [153–155], fusion [48, 156, 157] and progressive stereo methods [158] which also use costs at several matching hypotheses before making a depth estimate.

Traditionally, uncertainty of stereo matches has been modeled by cost-functions of pixel neighborhoods, also called windows. The cost function computes the cost of matching a pair of pixels between images and assumptions regard to noise distributions, continuity and local smoothness. Common cost functions include Sum of Squared Differences (SSD), Sum of Ab-

solute Differences (SAD) and different variants of Correlation. Other more elaborate cost functions have been proposed, some of which can be implemented as a filter to the images followed by one of the previously mentioned costs [42]. For a thorough comparison of cost functions refer to [42].

Based on these cost functions several models of stereo uncertainty, or confidence measures, have been proposed since the late 1980s. Some of them assume a winner-take-all approach, refining a disparity estimate around the least cost disparity, others take all costs into consideration. Models targeting WTA stereo are usually only defined at the highest-confidence (i.e. lowest-cost) match and do not provide confidence measures on the rest of the disparity range. Examples include left-right consistency checks, uniqueness or curvature tests (how much the highest-confidence is higher than others), texture thresholds, among others. Some of these WTA confidence measures were recently reviewed in [44, 45]. Other confidence measures include statistical models that compute a variance of the disparity estimate. Some models do so by polynomial fitting [159], others by modeling disparity and texture fluctuation inside windows [160], or even by directly computing the variance of WTA disparity between different window sizes [161].

Global methods, however, usually require a likelihood function over disparity to be propagated in order to obtain a final 3D reconstruction. This asks for confidence measures that are defined along the whole disparity range and that model the confidence on each stereo match hypothesis in a reliable way. Specifically, it is not only important that the highest-confidence disparity is of high accuracy but also that when this estimate is wrong, a high confidence is still attributed to the true disparity. Figure 4.1 shows an example of a good confidence function, or confidence measure, in these terms. A few stereo confidence measures have been proposed that are defined at all disparities within the disparity range, although they are only evaluated at WTA disparity in recent benchmarks [44]. For example, in [162], Matthies and Okutomi assume normally distributed image noise and model the probability of the measured pixel differences inside a window according to that model. Sun et. al use a pixel-wise likelihood function [153] in a global stereo method, propagating these likelihoods to neighboring pixels in a Markov Random Field formulation of stereo. The cost function used was the pixel dissimilarity function proposed by Birchfield and Tomasi in [163], chosen for its invariance to image sampling. Also, Mordohai recently proposed the SAMM measure [164] which computes a confidence for each disparity

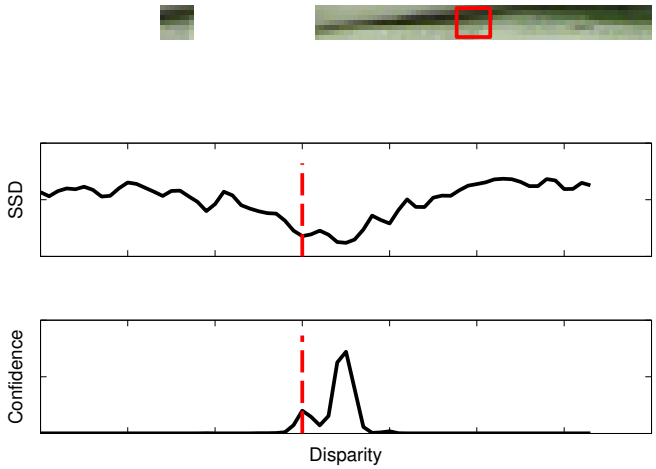


Fig. 4.1 Stereo matching confidence. Top: Matching a pixel in one image to pixels at different disparities in another image. Middle: Cost for each disparity. Bottom: Confidence measure computed from the cost values. Dashed line indicates true disparity. Even if the minimum cost is wrong, true disparity should still be attributed some confidence.

based on the correlation between the left-right stereo cost curve and the self-matching (i.e. left-left) cost curve. No explicit probability distribution assumptions are made. Although promising, the function scores poorly for large support windows when used with SAD costs [164]. Merrell et. al [156] assumes costs to be normally distributed with the mean parameter equal to the best cost value. It is also evaluated in [44].

Researchers have recently benchmarked several of these stereo confidence measures [44, 45, 165, 166]. Such benchmarks typically compare different methods for detection of correspondence errors [45, 165]; or evaluate whether stereo confidence measures can accurately rank matches on a WTA scenario [44, 45]. The latter make use of Receiver Operating Characteristic (ROC) curves for the evaluation, which have been frequently used in the stereo community [164, 167]. ROC curves are obtained by plotting the error-rate of a WTA strategy from the highest confidence matches, for different confidence thresholds. Using ROCs as the comparison criterion, a notable contribution to the state of the art of stereo confidence measures was made by Hu et. al [44]. In that article the authors analyze 17 different confidence functions both in terms of detection of correct WTA matches, occlusions and performance on discontinuities. Nevertheless, the influence of parameter choice on the performance of parametric functions was not discussed. In this chapter

we will study this problem and show that parameter choice drastically influences performance both in WTA stereo and global methods. Finally, these recent benchmarks were conducted mostly for confidence measures defined only at WTA disparity. Even when measures were well defined across the whole disparity range, evaluation was only made on WTA disparity. Such evaluations are hence useful for WTA methods but less so for global methods which integrate the information at all disparities, such as those targeted in this thesis. They leave out possible global and semi-global stereo approaches using multiple disparity hypotheses [153–155, 157, 167, 168].

4.2.3 Issues with common stereo reconstruction methods

Although WTA approaches to stereo are frequently preferred due to their higher computational speed, they are more susceptible to problems with occlusions, discontinuities, noise and lack of texture. Such problems can be avoided by discarding matches that could have happened by chance (a contrario models [169]), or that are ambiguous given the confidence measure (e.g. confidently stable matching [170], training of confidence thresholds from ground-truth [171]). However, these methods come at the cost of lower density. Global methods, by considering the whole disparity range and certain geometry assumptions, have the potential to better overcome such problems. Popular examples of these methods include dynamic programming [167], optimization methods using Markov network representations of stereo matching [153–155], among others.

Occupancy grids [172] provide an excellent tool for world mapping from sensor measurements, which is particularly useful for robot navigation and motion planning as we focus on in this thesis. Through this framework, a world map is built given sensor measurements, sensor position in the world and a sensor model. The map is defined as a grid of cells which can be in an occupied or free state. This is done with a probabilistic approach, accounting for uncertainty in the sensors. The framework is useful in order to accumulate information obtained from stereo cameras over time, as the robot "looks around" or navigates the environment. The concept was initially proposed for use with sonar sensors [172, 173] and later on also applied to stereo vision [174–178]. However, such attempts to integrate stereo into occupancy grids have opted to update cells based on the least-cost match at

each pixel (i.e. WTA approach) and then using the same sensor models as the ones used for sonar.

The existence of a confidence measure for each distance means more information is available than just a pixel-wise distance. As we will show later in Section 4.4, occupancy grid algorithms using stereo sensors can also improve performance by integrating confidence measures at all disparities instead of WTA disparity alone.

4.3 Improving stereo confidence measures

4.3.1 Considered parametric measures

We consider two images $I_1(x, y)$ and $I_2(x, y)$ coming from the same underlying image $I(x, y)$, displaced along the x axis with added Gaussian noise. Therefore,

$$I_2(x, y) - I_1(x + d(x, y), y) = \mathcal{N}(0, \sigma_i^2) \tag{4.1}$$

where $\mathcal{N}(0, \sigma_i^2)$ represents Gaussian white noise with variance equal to the sum of noise variances of each image $\sigma_i^2 = \sigma_1^2 + \sigma_2^2$. Here $d(x, y) \in \{0, 1, ..., D-1\}$ represents the disparity at each pixel. We define also a window with $M \times N$ pixels where (x, y) is the anchor pixel in the center of the window.

Different confidence measures model stereo matches differently. For example, one can model the probability of a disparity value $d(x, y)$ conditioned on a cost function of the pixels inside a window, but another option is to condition disparity on the whole set of pixel differences inside that window. We then define for each pixel (x, y) a matrix of measurements $E \in \mathbb{R}^{S \times D}$, where the D columns are disparity hypotheses and the rows are measurements used for the stereo confidence model (e.g. $S = 1$ for a single cost value per disparity, or $S = MN$ pixel differences per disparity). We will use the notation $E_{:,d}$ to represent all rows taken at disparity d. We will also refer to the disparity with minimum cost by $d_{mincost}$. Finally, in this work we assume independence of measurements at different disparities such that

$$p(E) = \prod_d p(E_{:,d}). \tag{4.2}$$

In this thesis we will deal with a special class of stereo confidence mea-

sures defined along the whole disparity range such that

$$C(d) = \frac{p(E_{:,d} \,|\, d)}{\sum\limits_{d'} p(E_{:,d'} \,|\, d')} \tag{4.3}$$

is the confidence of assigning disparity d to a certain pixel, and $p(E_{:,d} \,|\, d)$ is the probability density of measurements assuming d is the true disparity. Such formulation is used implicitly in other benchmarks [44] and will also be convenient for the integration into probabilistic frameworks described in Section 4.4.

We will evaluate and compare different confidence measures with two different stereo cost functions:

i. Sum of Squared Differences (SSD)

ii. Sum of Absolute Differences (SAD) using Birchfield and Tomasi's pixel dissimilarity function [163], which we will call BTSAD.

These are widely used cost functions, adopted by recent computer vision libraries [179] for local and global stereo methods. The implementations used in this work were those found in OpenCV [179], which also apply a 9x9 Sobel filter as a prefilter to the images. Sobel prefiltering is a common procedure seen in other stereo methods as well (e.g. [180]).

a) Matthies' model

Matthies and Okutomi [162] propose a probabilistic model of stereo that assumes pixel differences inside a window to be i.i.d. and zero-mean Gaussian distributed. The joint probability of all pixel differences is given by

$$p(E_{:,d} \,|\, d) \overset{i.i.d.}{=} \prod_s p(E_{s,d} \,|\, d) \propto exp\left(-\frac{1}{2\sigma_{Mat}^2} \sum_s E_{s,d}^2\right), \tag{4.4}$$

where $E \in \mathbb{R}^{S \times D}$ with $S = MN$. Each element $E_{s,d}$ holds one of the MN pixel differences inside a window at disparity d. Note that the joint distribution is related to a SSD ($\sum_s E_{s,d}^2$). Similarly to recent literature [44], we normalize the SSD by the number of window pixels[1] by setting $\sigma_{Mat}^2 = MN\sigma_i^2$.

[1]Note that the original model [162] sets $\sigma_{Mat}^2 = \sigma_i^2$. While the normalization by MN was not used in that publication, we still refer to the model as used in this thesis as "Matthies' model" for acknowledgment.

To obtain a similar model for a SAD cost function we can assume the i.i.d. pixel differences to follow a zero-mean Laplace distribution. The joint distribution is then given by

$$p(E_{:,d} \,|\, d) \overset{i.i.d.}{=} \prod_s p(E_{s,d} \,|\, d) \propto exp\left(-\frac{1}{b_{Mat}} \sum_s |E_{s,d}|\right). \tag{4.5}$$

In this case the joint distribution is related to a SAD ($\sum_s |E_{s,d}|$). Likewise the SSD case and since it lead us to better performance, we set $b_{Mat} = MNb_i$ where b_i is the parameter of the zero-mean Laplacian of single pixel differences.

b) Merrell's model

Merrel et. al [156] assume costs themselves to be normally distributed. The mean is set to the minimum cost of the corresponding pixel and variance is a parameter σ^2_{Mer}. Confidence is in this case defined by

$$p(E_{1,d} \,|\, d) \propto exp\left(-\frac{(E_{1,d} - E_{1,d\_mincost})^2}{2\sigma^2_{Mer}}\right), \tag{4.6}$$

where $E \in \mathbb{R}^{1\times D}$ and each element $E_{1,d}$ is a window cost value, e.g. $E_{1,d} =$ SSD or BTSAD.

c) The exponential distribution

The exponential model [153–155] assumes costs to be exponentially distributed and is given by

$$p(E_{1,d} \,|\, d) \propto exp\left(-\frac{E_{1,d}}{\mu}\right), \tag{4.7}$$

where $E \in \mathbb{R}^{1\times D}$ and each element $E_{1,d}$ is a window cost value, e.g. $E_{1,d} =$ SSD or BTSAD. Note that this model's expression is similar to Matthies'. However, while the exponential model is a pdf of the cost values, Matthies' is a joint pdf of all window pixel differences.

Note also that in other literature μ is often omitted from the equations, thus $\mu = 1$ is often assumed. The underlying problem of that assumption is that, for $\mu << E_{1,d}$ equation (4.7) will approximate $min(E_{1,d})$ and thus $p(E_{1,d\_mincost} \,|\, d_{mincost}) = 1$ will hold for all $d_{mincost}$. Such choice of parameter could hence lead to low performance of the confidence measure.

4.3.2 Parameter estimation

The parametric confidence measures introduced so far depend on the estimation of a probability distribution's parameter (σ^2_{Mat}, σ^2_{Mer}, μ). In this section we propose to estimate the parameters in a systematic way without ground-truth data, from each stereo pair being matched: through maximum likelihood (ML) estimation of the distribution's parameters computed directly from cost values. The method does not require ground-truth data but assumes cost functions provide relatively low error-rates (low number of bad pixels). To achieve this, *in our study we compute ML parameters from costs at all image pixels where left-right disparity consistency is verified.*

In a nutshell, we:

1. Compute cost values at all pixels and disparities;
2. Compute $d_{mincost}$ and perform a left-right disparity consistency check;
3. For all (x,y) with consistent disparities we compute the mean and variance of the costs at $d_{mincost}$;
4. Compute model parameters from those means or variances.

a) Matthies' model

Matthies' model for the SSD cost function assumes pixel differences to be zero-mean Gaussian. The Gaussian's parameter σ_i^2 can be computed by maximum likelihood from the variance of the data. For convenience we estimate this variance from the SSD cost values instead of the individual pixel differences. We do this by the following heuristic[2], which we found best performing:

$$\hat{\sigma}_i^2 = \frac{\sqrt{Var_{x,y}(SSD(x, y, d_{mincost}(x, y)))}}{MN\sqrt{2}}. \tag{4.8}$$

[2]Note that from the moments of the normal distribution we know that a variable X^2 has variance $2\sigma^4$ for $X = \mathcal{N}(0, \sigma^2)$. We compute the variance of an SSD by $Var(\sum_{s=1}^{MN} E_s^2) = 2\sigma_i^4 MN(1 + \rho(MN - 1))$, where ρ is the average correlation between the squared pixel differences E_s^2. Our heuristic assumes $\rho = 1$. While the original i.i.d. assumption of the model [162] would lead to $\rho = 0$, assuming $\rho = 1$ lead us to better performance results. Finally, note that another option for estimating σ_i^2 would be $\sigma_i^2 = Mean(\sum_{s=1}^{MN} E_s^2)/(2MN)$, which would make the estimated model's expression equal to that of the exponential.

As mentioned in Section 4.3.1 we set $\hat{\sigma}^2_{Mat} = MN\hat{\sigma}^2_i$, which is effectively eliminating the MN normalization in (4.8).

On a SAD (or BTSAD) cost function, we assume pixel differences are zero-mean Laplace-distributed, for which the maximum likelihood parameter is the mean of the absolute value of the data. As done in the SSD case, we compute this estimate from the cost values themselves:

$$\hat{b}_i = \frac{Mean_{x,y}(BTSAD(x, y, d_{mincost}(x, y))))}{MN}, \tag{4.9}$$

and we set $\hat{b}_{Mat} = MN\hat{b}_i$. Please note that using this normalization makes $\hat{b}_{Mat}$ equal to the costs' mean, leading to the same model expression and parameter as the exponential model (see (4.7) (4.11)). In this thesis, results obtained by maximum likelihood will then be the same for BTSAD Matthies' and the BTSAD exponential models.

b) Merrell's model

Merrell's model is a Gaussian distribution of costs with mean $E_{1,d\_mincost}$. The maximum likelihood parameter is estimated from the variance of the data,

$$\hat{\sigma}^2_{Mer} = Var_{x,y}(E_{1,d\_mincost}(x, y)), \tag{4.10}$$

where $E_{1,d\_mincost}$ is an SSD or BTSAD.

c) The exponential distribution

Given an exponential distribution of costs, the maximum likelihood estimate of the distribution's parameter μ is given by

$$\hat{\mu} = Mean_{x,y}(E_{1,d\_mincost}(x, y)), \tag{4.11}$$

where $E_{1,d\_mincost}$ is an SSD or BTSAD.

4.3.3 Histogram Sensor Model

We finally propose our new confidence measure - the HSM - which consists of a histogram trained with costs at true disparity. Confidence is modeled from the cost values and as such $E \in \mathbb{R}^{1\times D}$. In Figure 4.2, we show these histograms for SSD and BTSAD costs with different window sizes, taken from

true disparity d of all images in the 2003 and 2006 Middlebury datasets. We populated the histograms with costs measured at all un-occluded pixels of all images, while true disparity was retrieved from the ground-truth disparity maps provided by the datasets. The dimension of bins was chosen at $3.5\sigma_h/N^{1/3}$ according to Scott's normal reference rule [181], where σ_h represents the standard deviation of the costs and N the number of samples.

Stereo confidence is in this case defined as

$$p(E_{1,d} \,|\, d) \propto hist(E_{1,d}), \tag{4.12}$$

where $E_{1,d}$ is a window cost value, e.g. $E_{1,d} =$ SSD or BTSAD, and $hist(E_{1,d})$ refers to the frequency of the histogram bin associated with $E_{1,d}$.

4.3.4 Results

In this section we make use of stereo datasets and their ground-truth data to evaluate and compare the introduced stereo confidence measures. We base our comparison on two criteria:

i. Performance on a WTA strategy (selecting maximum confidence disparity at each pixel). For easy comparison with other literature, we make use of ROC curves [44, 164, 167]. These curves are obtained by plotting the error-rate of a WTA strategy from the highest confidence matches, for different confidence thresholds. The area under this curve, AUC, is used to measure the quality of the function as a confidence measure. Concretely, whether correct matches are given higher confidence than incorrect ones. *Lower values of AUC mean better performance.*

ii. We consider the cases where WTA disparity is different from true disparity by more than one pixel (we will call these "bad pixels"). We compute, at all bad pixels, the sum of the confidence attributed to a neighborhood around ground-truth disparity d^* given by the dataset: $C(d \in GT)_{badpx} = \sum_{d \in GT} C(d)$. Here GT represents the interval $[d^* - 1; d^* + 1]$. A single performance indicator for each image is then given by the average of $C(d \in GT)_{badpx}$ over all bad pixels. *Higher values of $C(d \in GT)_{badpx}$ indicate higher probability given to true disparity and, as we will argue, better performance of some global algorithms.*

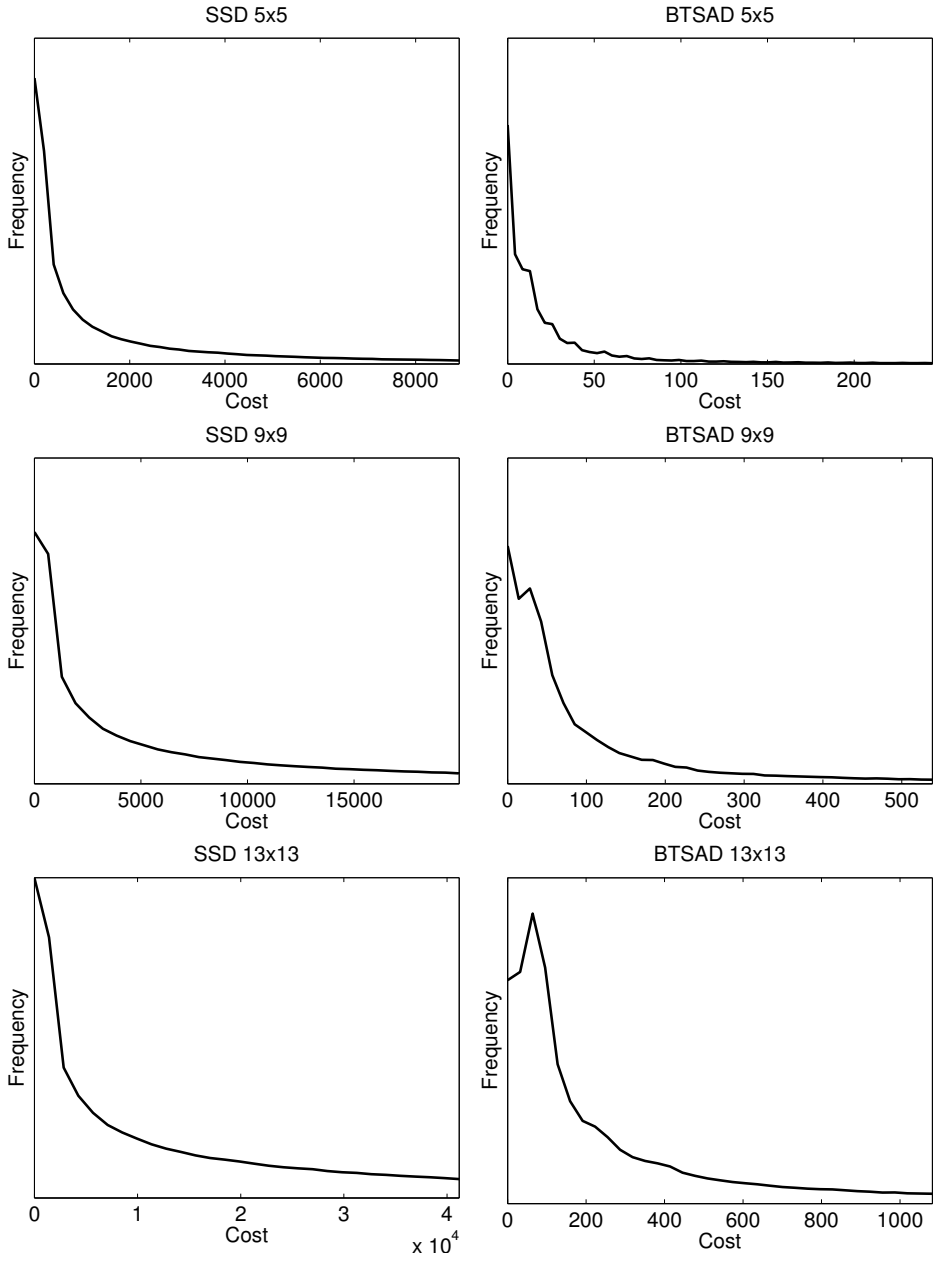


Fig. 4.2 Distribution of costs at true disparity (E_{1,d^*}) for SSD (left) and BTSAD (right) cost functions on a 5x5, 9x9 and 13x13 window. Horizontal axis represents the values of E_{1,d^*}.

We evaluated all models in two sets of data:

i. Indoors set: 23 stereo pairs (all pairs from Middlebury 2003 and 2006 [182–184])

ii. Outdoors set: 10 stereo pairs (KITTI stereo dataset [185], first 10 images).

For each set, the AUC and $C(d \in GT)_{badpx}$ results are averaged from all its stereo pairs and occluded pixels are excluded. The images were used in gray-scale. As cost functions we used SSD, and SAD with BT pixel differences (BTSAD) on window sizes 5x5, 9x9 and 13x13, after prefiltering the images with a Sobel 9x9 filter (OpenCV implementation [179]). This prefilter is adopted in several stereo methods (e.g. [179, 180]) and we also found both AUC and $C(d \in GT)_{badpx}$ performance to improve significantly with prefiltering for all models.

a) Parameter estimation

For the parametric functions introduced in Section 4.3.1, we evaluated the influence of parameter choice on the two mentioned performance criteria (i.e. AUC and $C(d \in GT)_{badpx}$). In Figure 4.3 and 4.4 we show the performance curves obtained for different window sizes, cost functions and confidence measures. Results are shown for four of the indoors stereo pairs. Other stereo pairs have similar curves, although we do not display all to keep figures understandable. The results show that performance of the confidence measures, with respect to parameter choice, has one clear maximum followed by a slow exponential decay of performance. However, a performance "cliff" exists as the parameter tends to zero (i.e. is under-estimated). One important observation is that $\mu = 1$ or $\mu = MN$, common parameter choices for the exponential model [44], could easily fall into the "performance cliff" by underestimating noise, thus drastically reducing performance. We believe this to be the reason why that model scores poorly in recent benchmarks [44] (it is there called Negative Entropy Measure). Furthermore, we argue that measuring parameter sensitivity through an analysis such as the one in Figure 4.3 and 4.4 or similar, should be used in future benchmarks and confidence measure proposals for more complete evaluations.

Another interesting observation is that these parameter performance curves have some inter-image variability. For each combination of cost func-

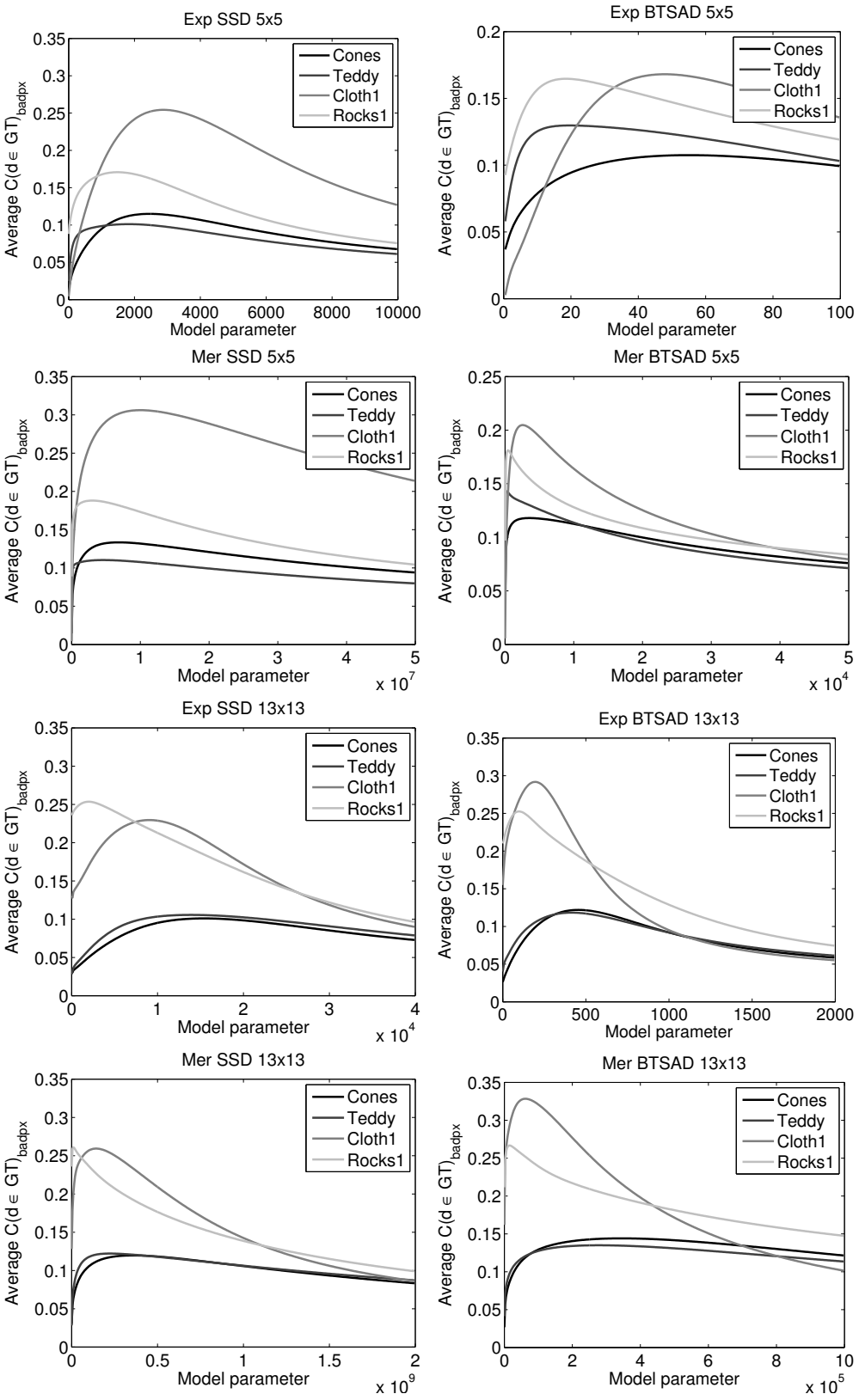


Fig. 4.3 The parametric models' cliff-maximum-and-tail of performance ($C(d \in GT)_{badpx}$). Results with the different cost functions and window sizes are shown. Note how the curves and optimal parameters vary both between images and cost functions. Figures for Matthies' model are not shown since they can be obtained by linearly rescaling the horizontal axis of the exponential model's figures (see equations (4.4), (4.5) and (4.7)).

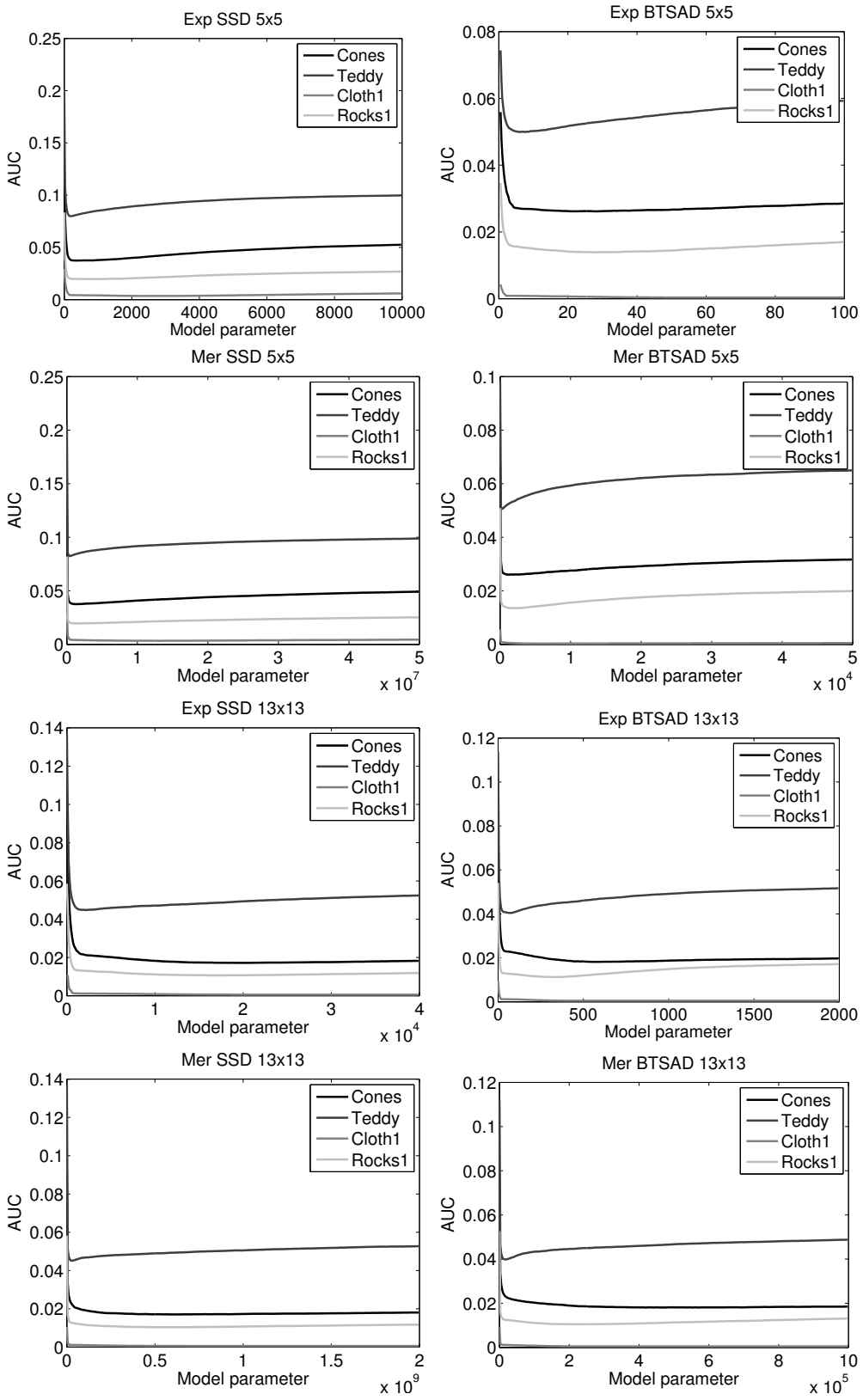


Fig. 4.4 The parametric models' cliff-maximum-and-tail of performance (AUC). Results with the different cost functions and window sizes are shown. Note how the curves and optimal parameters vary both between images and cost functions. Figures for Matthies' model are not shown since they can be obtained by linearly rescaling the horizontal axis of the exponential model's figures (see equations (4.4), (4.5) and (4.7)).

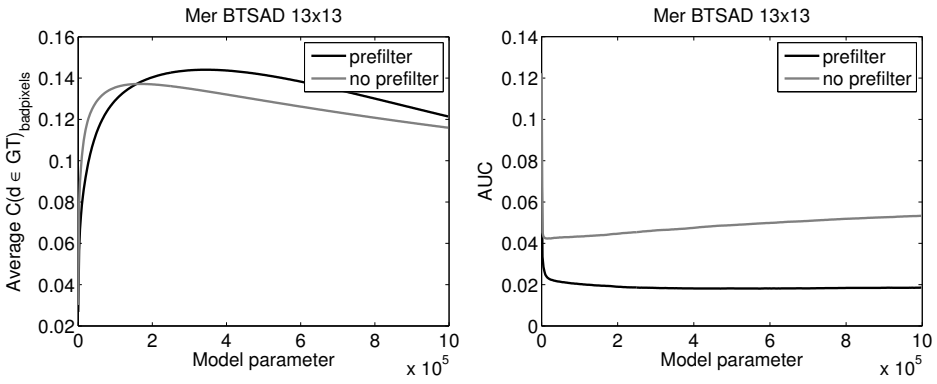


Fig. 4.5 Performance of models with parameter values changes with prefiltering conditions. Results obtained from the Cones image of the indoors set.

tion and window size, we computed the standard-deviation of the optimal parameter values across the 23 images of the indoors set. The average standard deviation of parameters was 131% when optimizing AUC and 84% when optimizing $C(d \in GT)_{badpx}$. On the other hand, optimal parameters also highly depend on the chosen cost function: for a fixed image the average standard-deviation across all combinations of cost function and window size was 352% in the AUC case and 338% in the $C(d \in GT)_{badpx}$ case. Even the fact that a prefilter is applied to the images, in our case the commonly used Sobel filter [179, 180], leads to an average displacement of the parameter with optimal AUC by 60% or optimal $C(d \in GT)_{badpx}$ by 167%. Figure 4.5 shows such a comparison, taken from the Cones image in the indoors set. Still, note that the AUC curves are relatively flat after the performance cliff and so optimal parameter variabilty does not pose a problem as long as parameters are not strongly under or overestimated.

Such performance variability between image conditions and between cost function options has strong implications for researchers working on stereo. During the design stage of a stereo algorithm, such as the experimentation with different cost definitions, prefiltering options and different datasets, the optimal value of the confidence measure's parameter should be recomputed each time. In Hu et. al's important contribution to confidence measure benchmarking [44], the authors compute an optimal parameter value for each measure on a subset of the images in the dataset: which requires recomputing all confidences and a performance value (e.g. AUC) for each parameter sample during an optimization process. The parameters were

there selected such that they lead on average to high performance within a subset of the dataset images, although the procedure is not described in detail. Besides the fact that averaging solves inter-image variability sub-optimally, such methodology (of optimal parameter estimation from datasets with ground-truth) could be a bothersome process when designing a stereo algorithm and considering a large number of cost function or prefiltering options. Automatic, fast estimation of stereo confidence parameters for a given image and cost function design, for example through maximum likelihood as proposed in this thesis, is then of high importance.

Optimal parameters for the confidence measures can only be computed when ground-truth disparity is available. Practically, on unknown stereo pairs, stereo methods have to either assume certain fixed parameter values (as discussed previously), or automatically estimate them from each image without ground-truth data. In this section we evaluate two different parameter estimation strategies for the parametric models:

i. Fixed parameters, computed using a slow offline optimization procedure on training datasets where ground-truth is available. Methodology used was similar to [44]: we estimated parameters by averaging the optimal parameters across train set images. For each image in the indoors set we first computed densely sampled parameter-performance curves such as the ones shown in Figure 4.3 and 4.4, and then averaged the curves' optima across all images. We will call these "average best performing" (ABP) parameters.

ii. Per-stereo-pair, maximum likelihood (ML) parameter estimation as proposed in this thesis, which does not require any ground-truth data. We will call these "ML" parameters.

Table 4.1 shows the ABP parameters that we used in this section, computed from the indoors set. Since these can be chosen to optimize either AUC or $C(d \in GT)_{badpx}$, we display both in the table. As we already observed, ABP parameters optimizing AUC (column "minAUC") have more variability than those optimizing $C(d \in GT)_{badpx}$ (column "maxC"). This suggests that a strategy of offline selection of parameters by averaging on a training set could be more reliable if the criterion being optimized is C.

We then computed the AUC and $C(d \in GT)_{badpx}$ metrics for each model using ML and ABP parameters. Table 4.2 shows the average and standard

Table 4.1 Average best performing parameters computed from the indoors set (total 23 images)

Cost	Model	minAUC param	maxC param
SSD 5x5	Mat	$2.95 \cdot 10^2 \pm 151\%$	$5.99 \cdot 10^2 \pm 92\%$
SSD 9x9	Mat	$1.91 \cdot 10^3 \pm 126\%$	$2.36 \cdot 10^3 \pm 47\%$
SSD 13x13	Mat	$4.17 \cdot 10^3 \pm 117\%$	$4.83 \cdot 10^3 \pm 42\%$
SSD 5x5	Mer	$2.59 \cdot 10^6 \pm 197\%$	$3.49 \cdot 10^6 \pm 103\%$
SSD 9x9	Mer	$5.49 \cdot 10^7 \pm 146\%$	$3.92 \cdot 10^7 \pm 65\%$
SSD 13x13	Mer	$2.82 \cdot 10^8 \pm 147\%$	$1.55 \cdot 10^8 \pm 59\%$
SSD 5x5	Exp	$5.94 \cdot 10^2 \pm 150\%$	$1.20 \cdot 10^3 \pm 93\%$
SSD 9x9	Exp	$3.67 \cdot 10^3 \pm 130\%$	$3.15 \cdot 10^3 \pm 98\%$
SSD 13x13	Exp	$8.27 \cdot 10^3 \pm 118\%$	$8.70 \cdot 10^3 \pm 56\%$
BTSAD 5x5	Mat	$1.18 \cdot 10^1 \pm 106\%$	$1.18 \cdot 10^1 \pm 88\%$
BTSAD 9x9	Mat	$5.64 \cdot 10^1 \pm 110\%$	$4.24 \cdot 10^1 \pm 94\%$
BTSAD 13x13	Mat	$1.12 \cdot 10^2 \pm 105\%$	$1.40 \cdot 10^2 \pm 67\%$
BTSAD 5x5	Mer	$1.88 \cdot 10^3 \pm 173\%$	$1.25 \cdot 10^3 \pm 126\%$
BTSAD 9x9	Mer	$3.89 \cdot 10^4 \pm 130\%$	$1.94 \cdot 10^4 \pm 124\%$
BTSAD 13x13	Mer	$1.81 \cdot 10^5 \pm 132\%$	$1.91 \cdot 10^5 \pm 101\%$
BTSAD 5x5	Exp	$2.37 \cdot 10^1 \pm 106\%$	$2.37 \cdot 10^1 \pm 88\%$
BTSAD 9x9	Exp	$1.13 \cdot 10^2 \pm 110\%$	$8.49 \cdot 10^1 \pm 94\%$
BTSAD 13x13	Exp	$2.24 \cdot 10^2 \pm 105\%$	$2.81 \cdot 10^2 \pm 67\%$

deviation of the distances between the obtained and the optimal performance taken from all 23 images of the indoors set. The table compares two situations: a typical scenario where ground-truth (GT) is not available on the image set, and another when it is available. In the "No GT" scenario, ABP parameters are computed from a different set (same images but without the use of image prefiltering with a Sobel prefilter). It is noticeable how in both situations ML parameters lead to values of AUC and $C(d \in GT)_{badpx}$ which are similar but slightly closer to the optimal value than ABP. This was expected from the high variability of optimal parameters, thus again stressing the importance of ML estimation or the use of parameter-insensitive confidence measures. The table also shows results obtained with the ML method ran on GT disparity instead of WTA (see columns ML-GT). It performed similarly to the no-ground-truth version and better than ABP on average. Importantly, these results mean that the tedious process of obtaining datasets with ground-truth for model training is unnecessary. Model parameters can be computed using our proposed ML strategy, without ground-truth data. Naturally, ABP had slightly higher performance when trained with GT than in the "No GT" condition.

To exemplify the better results of ML seen in Table 4.2, we also compare the shape of $C(d)$ at a given pixel of Middlebury's Teddy image which favors the ML method. In this example, shown in Figure 4.6, Merrell's model with ABP parameters behaves in a uni-modal way (i.e. single maximum), which exemplifies the effect of the "performance-cliff". We remind that as σ tends to 0, a normalized $exp(-\frac{x}{\sigma})$ becomes an approximation to $min(x)$, thus leading to a confidence of 1 on the best match and 0 otherwise. The model using ML parameters has two maxima: one on WTA disparity and another on ground-truth.

b) Benchmark of winner-take-all confidence

We evaluated each models' performance, including the HSM's, in the indoors and outdoors set using the two parameter selection strategies already discussed. In this section we focus on the AUC criterion. We remind that AUC measures whether higher confidence WTA assignments are more likely to be correct assignments or not. The models' AUC, averaged across all images in each dataset, is shown in Table 4.3. Each model's performance is shown with ML and ABP parameters. In case of the HSM, we also compare two

Table 4.2 On average, how close to optimal performance do models get?

	Distance to minAUC				Distance to maxC			
	No GT available		GT available		No GT available		GT available	
Model	ML	ABP-DS	ML-GT	ABP	ML	ABP-DS	ML-GT	ABP
Mat SSD	**0.08 ± 0.07**	0.12 ± 0.22	**0.11 ± 0.09**	0.11 ± 0.13	**0.11 ± 0.14**	0.19 ± 0.15	0.19 ± 0.16	**0.11 ± 0.12**
Mat BTSAD	**0.10 ± 0.22**	0.14 ± 0.29	**0.08 ± 0.17**	0.11 ± 0.14	**0.11 ± 0.09**	0.14 ± 0.10	**0.09 ± 0.08**	0.11 ± 0.11
Mer SSD	**0.06 ± 0.05**	0.12 ± 0.22	**0.06 ± 0.06**	0.09 ± 0.08	**0.04 ± 0.05**	0.10 ± 0.09	**0.07 ± 0.09**	0.07 ± 0.10
Mer BTSAD	**0.13 ± 0.27**	0.15 ± 0.29	**0.09 ± 0.18**	0.11 ± 0.10	**0.10 ± 0.08**	0.13 ± 0.08	**0.09 ± 0.08**	0.14 ± 0.17
Exp SSD	**0.06 ± 0.05**	0.12 ± 0.22	**0.08 ± 0.06**	0.11 ± 0.13	**0.12 ± 0.13**	0.19 ± 0.15	0.15 ± 0.15	**0.11 ± 0.12**
Exp BTSAD	**0.10 ± 0.22**	0.14 ± 0.29	**0.08 ± 0.17**	0.11 ± 0.14	**0.11 ± 0.09**	0.14 ± 0.10	**0.09 ± 0.08**	0.11 ± 0.11

Note: Distances computed as $|AUC_{Method}(img) - minAUC(img)|/minAUC(img)$ and $|C_{Method}(img) - maxC(img)|/maxC(img)$ averaged over all indoors images. ABP are average best performing parameters trained on the same image set given GT disparity; ABP-DS are average best performing parameters trained on a different set - same images different filtering conditions; ML parameters computed for each image given WTA disparity; ML-GT parameters computed using the same method on ground-truth disparity.

Table 4.3 Performance in AUC for all models and window cost functions, averaged over a test set

Test set: indoors (ABP/AGT is trained on the same set and requires GT disparity)									
	Optimal AUC	Mat		Mer		Exp		HSM	
Cost	(parametric)	ABP	ML	ABP	ML	ABP	ML	AGT	ML
SSD 5x5	0.083	0.087	0.088	0.091	0.087	0.087	**0.086**	0.088	0.106
SSD 9x9	0.058	0.063	0.063	0.065	0.063	0.063	0.062	**0.062**	0.085
SSD 13x13	0.056	0.060	0.061	0.062	0.060	0.060	0.060	**0.060**	0.084
BTSAD 5x5	0.066	0.069	0.067	0.070	0.068	0.069	0.067	**0.058**	0.065
BTSAD 9x9	0.051	0.055	0.054	0.056	0.054	0.055	0.054	**0.045**	0.058
BTSAD 13x13	0.050	0.054	0.053	0.056	0.053	0.054	0.053	**0.046**	0.064
Test set: outdoors (ABP/AGT is trained on a different set - indoors)									
	Optimal AUC	Mat		Mer		Exp		HSM	
Cost	(parametric)	ABP-DS	ML	ABP-DS	ML	ABP-DS	ML	AGT-DS	ML
SSD 5x5	0.223	0.230	0.233	0.233	0.229	0.230	0.232	**0.225**	0.256
SSD 9x9	0.175	0.180	0.184	0.183	0.181	0.180	0.183	**0.176**	0.230
SSD 13x13	0.202	0.205	0.207	0.206	0.206	0.205	0.207	**0.200**	0.273
BTSAD 5x5	0.147	0.152	0.153	0.155	**0.152**	0.152	0.153	0.153	0.157
BTSAD 9x9	0.117	**0.121**	0.123	0.124	0.121	0.121	0.123	0.122	0.136
BTSAD 13x13	0.145	0.148	0.149	0.149	0.148	0.148	0.149	**0.145**	0.168

Note: lower AUC is better. ABP are average best performing parameters computed from the indoors set using ground-truth; AGT are average ground-truth histograms as proposed in Section 4.3.3 i.e. HSMs trained on the whole indoors set using ground-truth; ML parameters are estimated for each image from WTA disparity, without ground-truth. Optimal AUC values are shown for comparison and were computed by a slow offline optimization procedure given ground-truth (minimum AUC across all parametric models and whole parameter space).

Table 4.4 Performance in $C(d \in GT)_{badpx}$ for all models and window cost functions, averaged over a test set

Test set: indoors (ABP/AGT is trained on the same set and requires GT disparity)									
	Optimal C	Mat		Mer		Exp		HSM	
Cost	(parametric)	ABP	ML	ABP	ML	ABP	ML	AGT	ML
SSD 5x5	0.108	0.083	0.090	0.097	**0.097**	0.083	0.090	0.077	0.083
SSD 9x9	0.091	0.076	0.072	0.084	**0.086**	0.076	0.074	0.061	0.066
SSD 13x13	0.101	0.086	0.073	0.093	**0.094**	0.086	0.073	0.060	0.072
BTSAD 5x5	0.109	0.087	0.086	0.088	**0.095**	0.087	0.086	0.076	0.094
BTSAD 9x9	0.099	0.084	0.083	0.090	**0.090**	0.084	0.083	0.067	0.085
BTSAD 13x13	0.112	0.095	0.094	**0.104**	0.103	0.095	0.094	0.070	0.088
Test set: outdoors (ABP/AGT is trained on a different set - indoors)									
	Optimal C	Mat		Mer		Exp		HSM	
Cost	(parametric)	ABP-DS	ML	ABP-DS	ML	ABP-DS	ML	AGT-DS	ML
SSD 5x5	0.065	0.053	0.049	0.052	**0.062**	0.053	0.050	0.031	0.043
SSD 9x9	0.059	0.047	0.036	0.045	**0.051**	0.047	0.036	0.025	0.028
SSD 13x13	0.046	0.037	0.029	0.036	**0.039**	0.037	0.029	0.022	0.020
BTSAD 5x5	0.084	0.063	0.060	0.055	**0.072**	0.063	0.060	0.040	0.061
BTSAD 9x9	0.079	0.055	0.045	0.048	**0.061**	0.055	0.045	0.030	0.050
BTSAD 13x13	0.069	0.048	0.039	0.043	**0.051**	0.048	0.039	0.027	0.040

Note: higher C is better. ABP are average best performing parameters computed from the indoors set using ground-truth; AGT are average ground-truth histograms as proposed in Section 4.3.3 i.e. HSMs trained on the whole indoors set using ground-truth; ML parameters are estimated for each image from WTA disparity, without ground-truth. Optimal C values are shown for comparison and were computed by a slow offline optimization procedure given ground-truth (maximum C across all parametric models and whole parameter space).

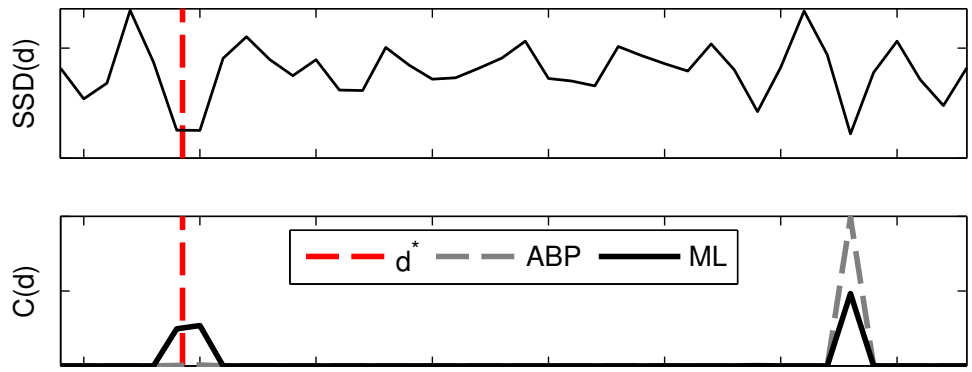


Fig. 4.6 Confidence $C(d)$ using Merrell's model with ABP and ML parameters. Dashed red line indicates true disparity d^* as indicated by the dataset. Results taken from pixel (364,150) of the Teddy image, as an example of ML's better performance seen in Table 4.2. ML does not require ground-truth and leads here to higher $C(d^*)$.

versions of the model, roughly corresponding to ML and ABP. The first version is a no-ground-truth single-stereo-pair model to which we will call "ML HSM". This histogram is trained from WTA disparity costs where left-right disparity is consistent, for each stereo pair. The second is the ground-truth-trained model as described in Section 4.3.3, computed from the costs at true disparity of all stereo pairs in the indoors set. We refer to it as "average ground-truth" (AGT) HSM.

Table 4.3 also shows the optimal AUC across parametric models, for each cost function. These values were obtained by a slow offline optimization procedure given ground-truth data, searching the minimum AUC across all parametric models and whole parameter space for each image. Values shown in the table are the average over all test set's images.

Arguably the most noticeable result is that the AGT HSM model ranks 1st in most conditions, both indoors (where it is trained) and outdoors. This indicates the HSM model to be a good choice when training on a dataset with ground-truth is acceptable. Expectedly, a histogram can better model the real distribution of costs than the parametric models here compared - we remind that distributions in Figure 4.2 are not purely exponential or Gaussian. This can also be seen clearly in the table results (indoors set, BTSAD cost function) where the HSM performs better than the parametric models' maximum possible performance (minAUC column). On the other

hand, the ML version of the HSM had poor performance, meaning the data available on a single stereo-pair may be insufficient to train the HSM for good AUC.

It is interesting to note, however, that cost function choice is crucial: note how it had higher impact on the AUC than model choice itself. We argue that the reason for this is that the models presented here are well estimated, rendering their fit to the real distribution, and performance, very similar to each other. Note again in Table 4.2 and 4.3 that obtained AUCs are very close to their optimal values, both in the indoors and outdoors set. Since optimal AUC depends on the error rate achieved by each cost function, as shown in [44], then as long as close-to-optimal AUCs are obtained on each model, performance will depend mainly on the cost function. The HSM seems to achieve AUC values that are closer to the optimal for each cost function.

Importantly as well, the results show once more that the usage of the datasets with ground-truth to train parametric models is (not only tedious but also) unnecessary, and our proposed ML strategy for parametric models leads consistently to high performance without the need for GT.

c) Benchmark on winner-take-all failure

We now present all models' performance regarding $C(d \in GT)_{badpx}$: the confidence given to true disparity when WTA fails. We compare the different models using this criterion in Table 4.4.

There is a different ranking of models in terms of AUC and C, which suggests that the appropriate choice of model for stereo applications strongly depends on which criterion is to be optimized. However Merrell's model, which had already scored high in the AUC criterion, performed highest in the C criterion using ML estimation (i.e. without the need for training with ground-truth datasets). Such consistency and convenience of ML-estimated Merrell's model makes it a good candidate model for stereo applications.

Regarding the HSM model, its AGT (ground-truth-trained) version performed quite low. Its ML (no-ground-truth) version performed higher, even though it was poor on AUC (Table 4.3). In the next section we will see how this balance between AUC and C is actually reflected on high performance of both versions of the HSM in practice.

4.4 Integrating stereo over time

4.4.1 Cost-curve occupancy grids

a) Definition

Consider a grid of cells which can be in one of two states: occupied O or free $\overline{O}$. The objective of an occupancy grid algorithm is to compute or update the probabilities $p(O_i|z_{0...t}, x_{0...t})$ for each cell $i \in 1, 2, ..., C$, at each time instant t, given measurements $z_{0...t}$ and sensor locations $x_{0...t}$ until time t. This is implemented as a Bayes filter at each cell, which updates occupancy probabilities every time a new measurement is taken [31].

In this section we propose a new Cost-Curve Occupancy Grid method to compute occupancy at each cell from stereo cost measurements at the whole disparity range. The method computes occupancy of cell i as

$$\begin{aligned} P(O_i|E) = P(O_i|V_i, E)P(V_i|E) + \\ P(O_i|\overline{V}_i, E)(1 - P(V_i|E)), \end{aligned} \tag{4.13}$$

where the event $V_i = \overline{O}_{i-1}, ..., \overline{O}_2, \overline{O}_1$ represents visibility of cell i. For the sake of readability and compactness, the equations shown here are for a one-dimensional grid aligned with the sensor - correspondent to the intersection of a camera ray with the three-dimensional grid. Also, the order of cells is reversed from that of pixel disparity: for example $i = 1$ is the closest cell to the camera, equivalent to $d = D - i = D - 1$.

The probability $P(V_i|E)$ can be computed by recursively applying the definition of conditional probability,

$$\begin{aligned} P(V_i|E) &= P(V_{i-1}\overline{O}_{i-1}|E) \\ &= P(\overline{O}_{i-1}|V_{i-1}, E)P(V_{i-1}|E) \\ &= ... = \prod_{j=1...i-1} P(\overline{O}_j|V_j, E). \end{aligned} \tag{4.14}$$

On the other hand, $P(O_i|V_i, E)$ is given by

$$P(O_i|V_i, E) = \frac{p(E|O_i, V_i)P(O_i, V_i)}{P(V_i|E)p(E)}, \tag{4.15}$$

where $P(O_iV_i)$ is a prior on world geometry.

The denominator of (4.15) can also be computed recursively as

$$\begin{aligned} &P(V_i|E)p(E) = \\ &= P(O_i, V_i|E)p(E) + P(\overline{O}_i, V_i|E)p(E) \\ &= p(E|O_i, V_i)P(O_i, V_i) + P(V_{i+1}|E)p(E) \\ &= ... = \sum_{j=i...C} p(E|O_j, V_j)P(O_j, V_j), \end{aligned} \tag{4.16}$$

where we assume that $P(V_{C+1}|E) = 0$, as we will explain next.

The method makes the following assumptions:

- A target surface exists for any 1D grid, or in other words, there exists at least one occupied cell. Thus $P(V_{C+1}) = 0$ and $P(V_{C+1}|E) = 0$;
- The target is equally probable to be at any of the cells along the 1D grid. Thus $P(O_i, V_i) = 1/C \ \forall_i$;
- Measurements E can give no information about occupancy on invisible cells $\overline{V}_i$. Thus $P(O_i|\overline{V}_i, E) = P(O_i|\overline{V}_i)$, which corresponds to a prior on world geometry. In our work we model this prior as a constant 0.5 for all i, so that occupied and free cells are equally probable. Thus $P(O_i|\overline{V}_i) = 0.5 \ \forall_i$;
- Measurements are independent between disparities (see (4.2)).
- $p(E_{:,d})$ is uniform.
- Occupancy or visibility on a cell i gives no information on match measurements taken on other cells. Thus $p(E_{:,D-k}|O_i, V_i) = p(E_{:,D-k}) \ \forall_{k \neq i}$;

From (4.15), (4.16) and the second assumption follows that

$$P(O_i|V_i, E) = \frac{p(E_{:,D-i}|O_i, V_i)}{\sum_{j=i...C} p(E_{:,D-j}|O_j, V_j)}. \tag{4.17}$$

Note that (4.17) is similar to our definition of stereo match confidence (4.3) if disparity is seen as a position (i.e. cell) which is both occupied and visible.

b) Traditional occupancy grids as a special case

In traditional occupancy grids, a single metric distance to a target is directly or indirectly measured [18]. Since no other information is available, the

real distance to the target is modeled as a normal distribution around the measured distance. Uncertainty on the measurement is modeled using the distribution's variance. Such range sensors can be seen as a special case of cost-curve sensors, but where a single cost is measured. If a target is measured to be at cell k, then in our formulation E_k is minimum and $E_i \forall_{i \neq k}$ are equal and maximum. In traditional range-measurement occupancy grids we then have $p(E|O_i, V_i) \propto exp\left(-\frac{(i-k)^2}{2\sigma^2_{range}}\right)$, to which all equations we just defined apply. As we discussed in the Background section, such models in computer vision are referred to as "winner-take-all" (WTA) models - where the distance with minimum cost is selected and the rest of the cost-curve discarded.

4.4.2 Results in visually repetitive environments

A straight-forward application of the cost-curve occupancy grid formulation is a scene with vertical repetitive characteristics. The Peak Ratio (i.e. second-best cost over best cost) of the cost curves will be low, therefore leading to either false positives or holes in the reconstruction depending on whether the Peak Ratio is thresholded or not in a WTA approach. On the other hand, a whole-cost-curve approach is expected to keep the occupancy probability at repetitions high enough, and eliminate false-positives with time, as the viewing angle changes.

To empirically confirm this hypothesis we simulated a simple environment with thin vertical bars, camera moving around them. Figure 4.7 shows the resulting reconstruction of the scenario after 20 frames of camera motion using the cost-curve occupancy grid with Merrel costs and ML-estimated parameters. Blue regions indicate occupied cells, which should form parallel bars. On the figure, cells are drawn on top of the point cloud obtained from least-cost stereo matches.

Figure 4.8 shows the results after the same number of frames from the winner-take-all approach using three different stereo filtering thresholds (Peak Ratio). High confidence restrictions lead to holes in the reconstructions. Less filtering however leads to more errors and intensive post-filtering is needed. We did not find a threshold leading to a reconstruction with no holes and no outliers. The image in the center reveals that there are still holes in the reconstruction when outliers start appearing on a traditional winner-take-all approach.

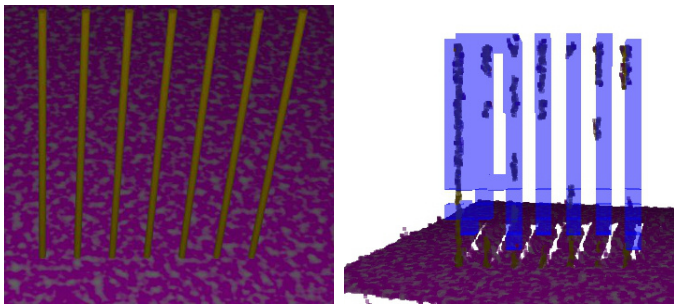


Fig. 4.7 Virtual repetitive scenario and cost-curve occupancy grid result. Left: virtual scenario with vertical bars to induce similar cost minima. Right: resulting occupancy grid using the cost-curve approach. Occupied cells are marked with blue. Result should be 7 parallel bars.

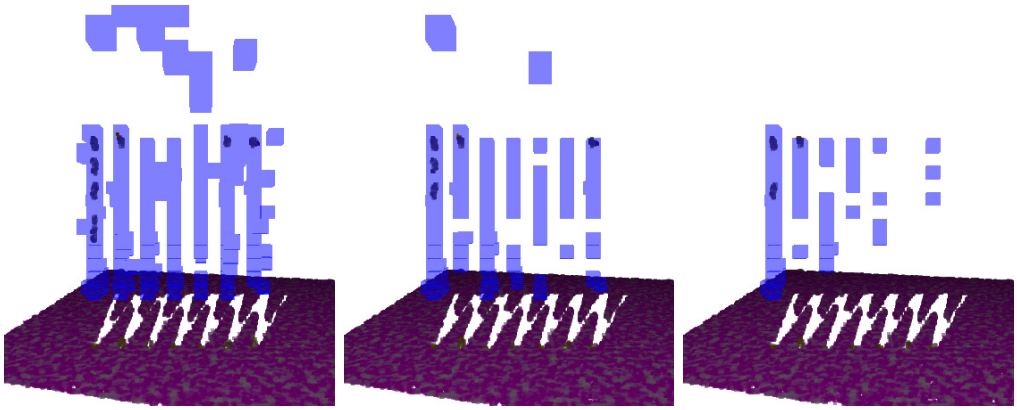


Fig. 4.8 Resulting occupancy grid computed in a traditional winner-take-all approach when using three different filtering thresholds (Peak Ratio). Left: 1.4. Center: 1.5. Right: 1.6. Occupied cells are marked with blue. The result should be 7 parallel bars.

Cost-curve occupancy grids with automatically estimated parameters achieved full reconstruction without outliers and did not require manual calibration.

4.4.3 Reconstruction results in the real world

We now evaluate the cost-curve occupancy grid method using the stereo confidence measures compared previously. In this section we will describe the setup and results, as well as discuss the relation between grid performance and the AUC and C criteria results.

Our grid method assumes static scenes and so the experimental evaluation was also conducted on a dataset with no moving objects: the KITTI residential area dataset "2011_09_26_drive_0079" [185]. The dataset contains 100 synchronized stereo pairs, laser rangefinder measurements and localization data taken from a moving car, while no moving people or moving cars can be seen. An image of this dataset is shown in Figure 4.9.

In order to obtain a ground-truth grid, a simple grid algorithm for range data was implemented and run on all frames using the available laser rangefinder data: cells that were occupied with point data in more than a single frame were considered occupied and the rest as free. The localization data, given by the dataset, was assumed to be correct. Cell size used was 20cm x 20cm x 20cm and the resulting grid 60m x 12m x 3m. Generated ground-truth is shown in Figure 4.9.

To quantitatively evaluate performance of the occupancy grid method we take two measures: "precision" and "recall". Precision measures the fraction of cells classified as occupied which are correct. It is defined as $\frac{tp}{tp+fp}$, where tp (true positives) refers to the number of cells correctly classified as occupied (i.e. occupancy $P > 0.5$) and fp (false positives) refers to the number of cells incorrectly classified as occupied. Recall measures the fraction of occupied cells correctly classified. It is defined as $\frac{tp}{n}$, where n refers to the total number of occupied cells on ground-truth data.

a) Precision, recall, AUC and confidence on ground-truth

We computed reconstruction performance with all models, including the HSM, using both ABP/AGT and ML parameter estimation. Results are shown in Figure 4.10. For the ABP parameters of parametric models, we ran the experiment with both maxC and minAUC parameters (see Table

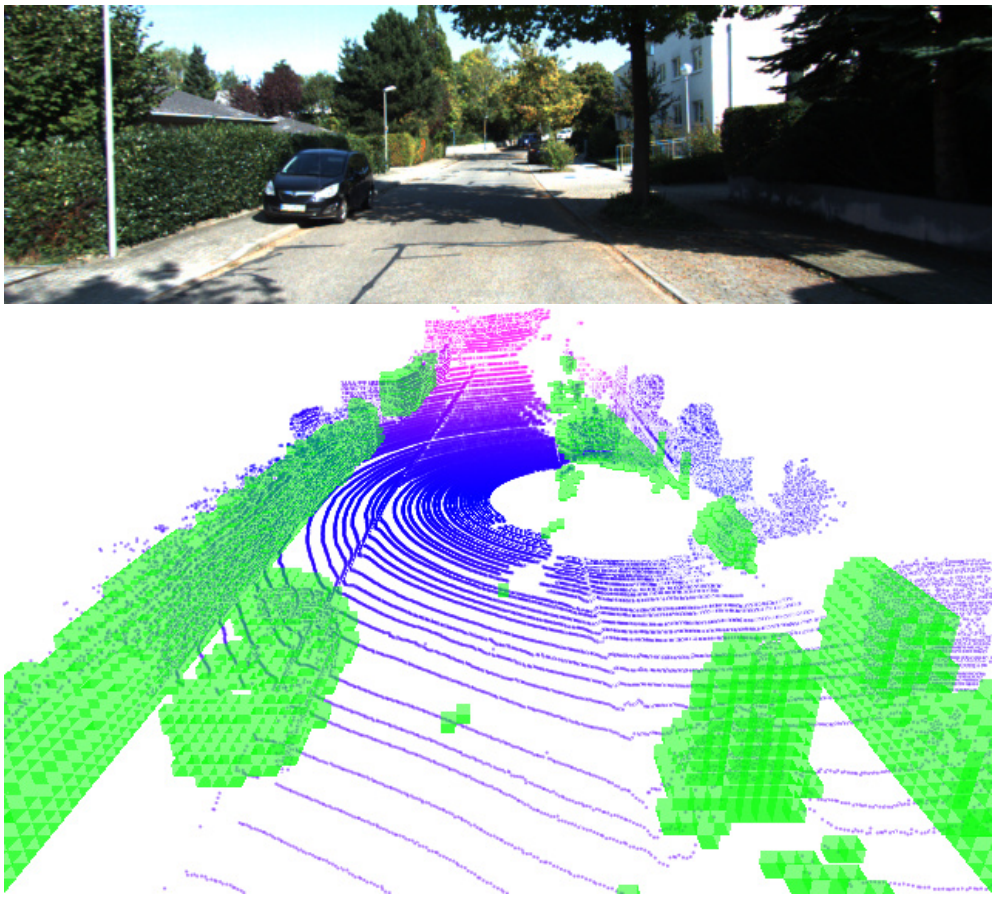


Fig. 4.9 The KITTI residential area dataset [185] used for occupancy grid evaluation. Green regions on the bottom image represent ground-truth occupied cells. Blue points represent laser data at one of the frames.

4.1). Their curves are similar, though, and so we include only one of them (minAUC) in Figure 4.10. Each dot in the figure represents one instant of time of the image sequence (i.e. frame) and hence an update of the occupancy grid. The first frames are marked with "t=0". Frames used were: 0, 5, 10, etc, in multiples of 5.

The curves in Figure 4.10 show how the occupancy grid algorithm leads to increasingly higher recall and precision rates as new frames are processed. Precision rates of around 0.9 and recall 0.5 are achieved by most models by the end of the experiment. Another observation is that precision increases slightly with window size, which is consistent with the results in Section 4.3.4.

Importantly, the HSM and Merrell models lead to the highest final precision results across most cost function and window size combinations, with the exception of BTSAD 5x5. The ML-estimated exponential had slightly higher precision in that case, however at the cost of low recall. Also note that the HSM model's curve is above other curves during most of the image sequence, showing highest precision, although this distance decreases as the number of used images increases. Models with ML and ABP parameters perform similarly for each model-cost combination, with the exception of Matthies' and the exponential models where ML leads to higher precision but lower recall. These results are consistent with Tables 4.3 and 4.4: HSM and Merrell were best performing in either the AUC or C criterion, also ML Exp and Mat had lower C score than their ABP versions, corresponding to the lower recall in the grid application. Overall, higher C criterion is related to higher final grid recall (correlation $r = 0.29$), but not related to precision in our method. Lower AUC is also related to higher final grid recall (correlation $r = -0.35$) and higher final precision (correlation $r = -0.48$).

An interesting observation is how the ML HSM lead mostly to the same performance as AGT HSM, even though AUC in the ML case was poor. As we discussed in Section 4.3.4, the fact that an ML HSM is computed from a single stereo pair could lead to a sparsely populated histogram: thus leading to a poor AUC because the confidence function is not continuous (and ranking of pixels as a function of error rates will also not be continuous). However, the ML histogram is trained from costs at WTA disparity where left-right disparity is consistent. Thus the reason for the ML model's poor AUC could be its bad conditioning near cost values where errors are common (and thus left-right consistency is often not met), even though conditioning

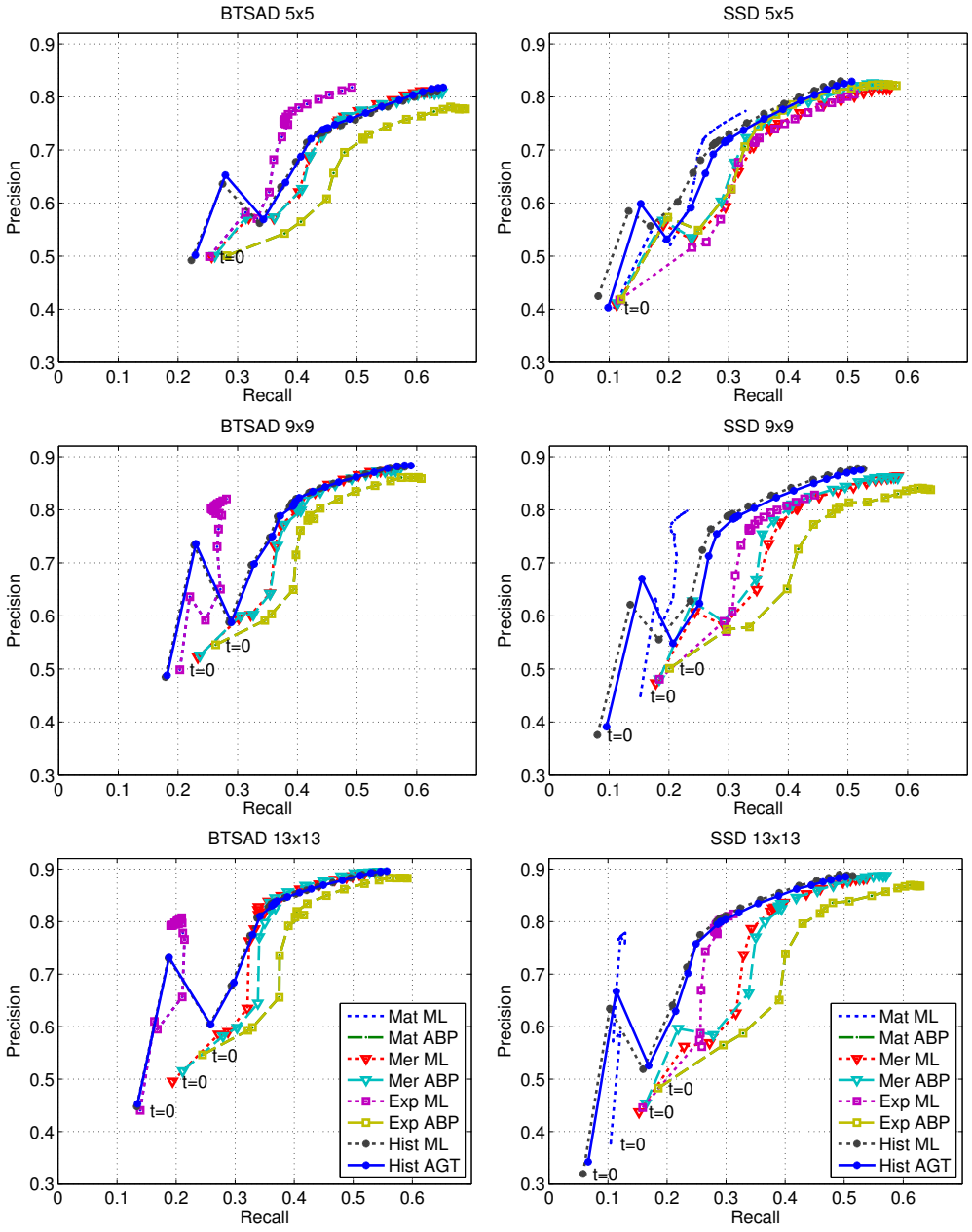


Fig. 4.10 Comparison of the performance of all models along time when used with the occupancy grid algorithm. Each point represents a different instant of time, while the first frame of the image sequence is marked with t=0. "Mat ABP" overlaps perfectly with "Exp ABP" on both cost functions, and "Mat ML" overlaps perfectly with "Exp ML" for the BTSAD cost function.

is good around common cost values of true disparity. This would explain the still high $C(d \in GT)_{badpx}$ result of the model (see Section 4.3.4, Table 4.4), as well as its good performance in the occupancy grid application. Such observations again stress the need for criteria other than AUC for stereo confidence model evaluation, depending on the application.

Finally, in Figure 4.11 we show the reconstruction of ML HSM and Merrell's models (using BTSAD 13x13). The HSM's higher recall can be seen quite clearly (e.g. the car and tree are better reconstructed), although the number of false positives is also slightly higher (since recall is higher and precision rate is not 1).

b) Robustness to noise

We also analyzed the influence of image noise in the performance of the occupancy grids, by adding different levels of Gaussian noise to the image sequence. Taking into consideration the original noise estimate of $\sigma^2 = 13$, the resulting pixel intensity noise variance levels tested were $\sigma^2 = 13$, 14, 15, 18, 25, 43, 83, 177 and 397, where pixel intensity is in the range $[0; 255]$.

We run the algorithm with a WTA grid and with a ML-estimated Merrell model. The performance of the occupancy grids quickly deteriorates in both cases. Specifically, grid precision is a function of the power of noise $precision(\sigma^2) = a * (\sigma^2)^b + c$, (SSE $\in [0.0003; 0.004]$). However, cost-curve occupancy grids were more robust to noise than WTA - allowing for higher noise variances to obtain the same grid precision. In Table 4.5 we show the maximum allowed image noise for different values of minimum precision.

4.5 Discussion

We will organize the discussion of this chapter according to the objectives we set in the beginning.

4.5.1 Stereo confidence measures

Performance of parametric stereo confidence measures varies drastically with parameter choice, concretely showing a cliff-maximum-and-tail of performance with parameters. This observation forces stereo algorithm designers and users in robotics to consider doing parameter estimation before applying the algorithm to stereo reconstruction. The reason for performance drop

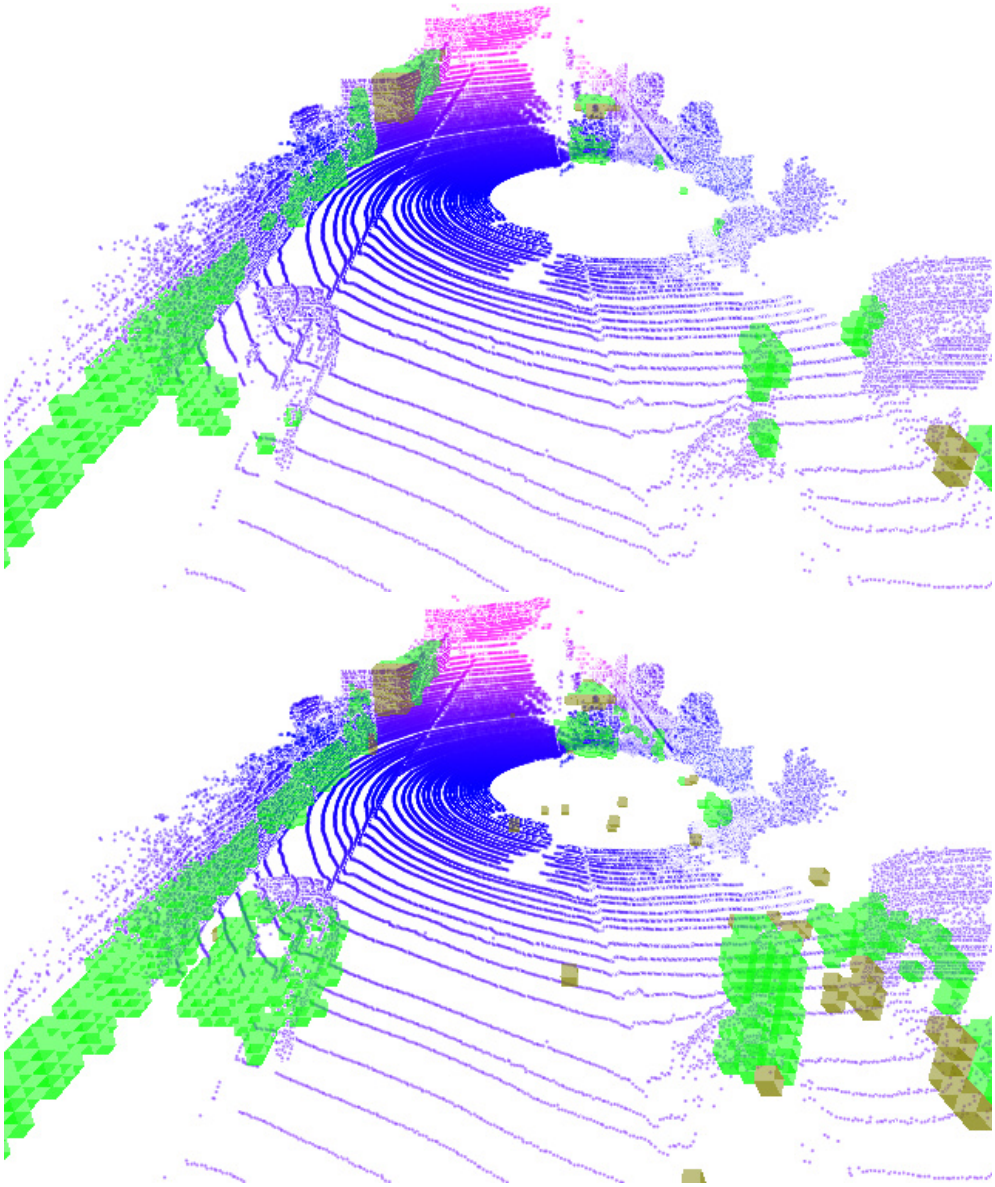


Fig. 4.11 Reconstruction results obtained using a BTSAD 13x13 cost function with the two top models. Top: Merrell's model. Bottom: HSM. Green squares represent true-positives (i.e. cells correctly classified as occupied), brown squares represent false-positives (i.e. cells incorrectly classified as occupied).

Table 4.5 Maximum acceptable image noise variance σ^2 for desired grid precision

		Mer	WTA
SSD 5x5	Min. precision 0.65	–	–
	Min. precision 0.75	–	–
	Min. precision 0.85	–	–
SSD 9x9	Min. precision 0.65	83	15
	Min. precision 0.75	43	–
	Min. precision 0.85	–	–
SSD 13x13	Min. precision 0.65	177	83
	Min. precision 0.75	83	14
	Min. precision 0.85	43	–

when parameters are under-estimated is clear: since the analyzed confidence functions are normalized exponentials of costs, they tend to a *min* function as the cost normalizer tends to zero (is under-estimated) - leading to a single confidence maximum equal to 1. As a general rule, we showed that over-shooting of parameters is safer than under-shooting.

We hope to have made clear that more research into methods for online (no ground-truth) estimation of model parameters has the potential for high impact on stereo and its applications - such as robot locomotion. Other approaches to training the HSM without ground-truth may also be worth investigating, as is the combination of different confidence measures [186].

4.5.2 Improving their performance

Our results indicate that it is possible in certain applications to train parameters of the parametric stereo confidence models from off-the-shelf datasets with ground-truth disparity (i.e. using average best performing parameters, ABP). However, care should be taken such as to re-train the parameters every time costs, prefilters or dataset conditions are changed.

We proposed a systematic parameter estimation method for parametric models using maximum likelihood (ML), eliminating the need for any ground-truth or offline training. Our results indicated that these parameters lead to performance in stereo which is similar but slightly closer to the optimum when compared to ABP parameters - which require training datasets with ground-truth. At the same time, the proposed method is triv-

ial to implement and computationally inexpensive. ML should allow for better compensation of environment changes and be more practical when different cost or prefiltering options are applied during the design stage of algorithms.

The HSM was the best performing model in terms of AUC and occupancy grid precision when trained on off-the-shelf datasets with ground-truth. As seen by the shape of the HSM (Figure 4.2), the distribution of costs at true disparity is not well approximated by a distribution of the exponential-family. We believe this to be a good sign for a push in stereo research towards non-parametric confidence models. For applications where AUC is an important criterion, our results show however that the HSM should not be trained on WTA disparity with few data. Merrell's model with ML parameters is a good choice when ground-truth datasets are not available for training, since it scores high in terms of AUC, $C(d \in GT)_{badpx}$ and grid performance.

One important conclusion of this chapter is that one way to improve performance of 3D reconstruction algorithms for stereo, is to look at non-parametric models of stereo confidence, or models with low parameter sensitivity.

4.5.3 Uncertainty-aware stereo reconstruction

We showed that integrating the uncertainty of stereo matching into occupancy grids, through stereo confidence measures, leads to better reconstruction performance. Namely, from a simulation experiment we concluded that the proposed approach better deals with visually repetitive (thus high uncertainty) patterns. Whole-cost-curve integration brings more evidence to the right matches, eventually leading to better reconstruction: without pre or post discarding of any matches. The method, combined with automatically estimated model parameters, leads also to higher robustness to noise.

In terms of stereo confidence measures, the HSM and Merrell's model performed best in terms of grid precision. The HSM actually achieved higher precision earlier on (i.e. using a fewer number of stereo pairs). On the other hand, the exponential and Matthies' models with ABP parameters lead to overall high recall rates but lower precision.

The AUC criterion is usually used to evaluate the quality of stereo confidence measures in benchmarking literature. However, this criterion is less

informative than desirable when used to choose the best model for a global method integrating confidence measures, such as our Cost-Curve Occupancy Grid. We proposed another criterion, $C(d \in GT)_{badpx}$, which is related to the recall of the grid and ML HSM's performance. Training of parameters by optimizing $C(d \in GT)_{badpx}$ is also subject to lower inter-image variance than AUC.

4.6 Summary

In this chapter we explored areas of computer vision closely related to the robot locomotion problem: stereo vision, and also mapping through occupancy grids. We first tried to find which functions best model stereo matching uncertainty. We evaluated several existing models of confidence which are defined at the whole disparity range, and we introduced a way to improve their performance through automatic parameter estimation. We proposed a new stereo confidence measure, the Histogram Sensor Model (HSM), which better models stereo matching uncertainty and improves reconstruction performance in several criteria. Then we applied these stereo confidence measures to occupancy grids, a 3D reconstruction method which estimates world geometry from sequences of stereo pairs - as an agent looks around or navigates the environment. This world representation is characterized by fast access for collision checking and robot motion planning algorithms such as the ones we are interested in here. The key feature of the method we introduced is that occupancy is computed not from the least-cost estimate of distance given by stereo, but from the likelihoods of all costs along the cost-curve. Such an approach has higher performance robustness to environment texture and also image noise.

Chapter 5

Vision-based hierarchical planning in the real world

5.1 Introduction

We have until now discussed how to plan humanoid locomotion accounting for world friction and geometry, how to estimate friction from vision and how to estimate geometry from vision. We now turn into a more practical problem of how to integrate all these components into a single architecture, and implementing it on real locomotion scenarios.

With the tools we have described so far, we can estimate world geometry fairly accurately. Recent stereo sensors, such as the Carnegie Robotics Multisense-SL which we use in our real-robot experiments, have around 3mm error at a distance of one meter, and 3cm at a distance of 10 meters. With additional time filtering using occupancy grids such as we proposed in Chapter 4, error could become even lower. Such small errors can arguably be dealt with at the feedback control level of robots, and would hardly have any influence on planned trajectories even if they were taken into account. However, the same cannot be said about the coefficient of friction. As we saw in Chapter 3, errors are expected to have a standard deviation of 0.13, which is a huge difference in terms of locomotion requirements. As we can see from Figure 2.5, in Chapter 2, a change of 0.13 in coefficient of friction could require a twice lower speed. It is then of utmost importance to include this uncertainty into the planning algorithm we presented in order to produce robust plans.

In addition to robustifying our algorithms from Chapter 2, it is also important to evaluate the performance of the whole architecture on a variety of scenarios. These two points will be the focus of this chapter. Concretely, our objectives are:

a) To integrate the uncertainty of friction from vision into our planning algorithm for robust trajectories (Section 5.2)

b) To evaluate the whole system (planning and perception) on a variety of scenarios (Sections 5.2.3 and 5.2.4).

5.2 Perception-planning architecture

5.2.1 Robust planning using chance constraints

In Chapter 2 we introduced an extended footstep planning algorithm which considers surface friction by using it as a constraint when computing minimum state transition costs. The algorithm assumed the coefficients of friction were known, so some slight changes are necessary before applying the algorithm to a real robot.

We remind that the only point at which the algorithm depends on friction is the state transition cost $c(s, s')$ used during the tree-based search of stances. This was given by equation (2.7), which was:

$$\begin{aligned} c(s, s') = & \min_{p} \ \hat{f}_{\text{cost}}(s, s', p) \\ & \text{subject to} \\ & \Psi(s, s', p) < 0 \\ & \hat{f}_{\text{RCOF}}(s, s', p) < \mu \\ & a < p < b. \end{aligned}$$

Assuming that the functions $\hat{f}_{\text{cost}}$ and $\hat{f}_{\text{RCOF}}$ are deterministic and the noise in the system is only in the measurement variable μ, it is easy to convert this optimization problem to a robust one without increasing problem complexity by using chance constraints [187]. First, we rewrite the state

transition cost function as

$$\begin{aligned} c(s,s') = & \min_{p} \ \hat{f}_{\text{cost}}(s,s',p) \\ & \text{subject to} \\ & \Psi(s,s',p) < 0 \\ & P(\hat{f}_{\text{RCOF}}(s,s',p) < \mu^{(c)}) \geq \eta \\ & c = 1, ..., C, \end{aligned} \tag{5.1}$$

where c is an index of the contacts of s and s', while $\mu^{(c)}$ is the friction at these contacts. This way, we enforce that the Coulomb friction inequality holds with at least probability η. The constraints can also be rewritten using the cumulative distribution function $p(\mu^{(c)}|\theta, I)$ (equation (3.9)), which we denote by $F_{\mu^{(c)}|\theta,I}$,

$$F_{\mu^{(c)}|\theta,I}(\hat{f}_{\text{RCOF}}(s,s',p)) \leq 1 - \eta. \tag{5.2}$$

Since each $\mu^{(c)}$ is one-dimensional then F can be inverted and the constraints rewritten in deterministic form

$$\hat{f}_{\text{RCOF}}(s,s',p) \leq Q^{(c)}_{1-\eta}, \tag{5.3}$$

where $Q^{(c)}_{1-\eta}$ is the $(1-\eta)$-quantile of $F_{\mu^{(c)}|\theta,I}$.

In practice we compute the quantiles at all image pixels inside the friction perception algorithm, and so the friction-annotated point cloud is actually a friction-quantile-annotated point cloud. Therefore, obtaining the friction quantile for a certain contact c of a stance during search simply involves accessing its value in memory.

We efficiently obtain the friction quantiles at all pixels $k = 1, ..., n$, by querying the c.d.f.s at certain friction values μ until we (approximately) find $\mu : F_{\mu^{(k)}=\mu|\theta,I} = 1 - \eta$. Let us define

$$\mathbf{F}(\mu) = \begin{bmatrix} F_{\mu^{(1)}=\mu|\theta,I} \\ \vdots \\ F_{\mu^{(n)}=\mu|\theta,I} \end{bmatrix}, \tag{5.4}$$

$$\mathbf{W}_{\theta,I} = \begin{bmatrix} P(x_1 = l_1|\theta, I) & \dots & P(x_1 = l_m|\theta, I) \\ \vdots & \ddots & \vdots \\ P(x_n = l_1|\theta, I) & \dots & P(x_n = l_m|\theta, I) \end{bmatrix}, \tag{5.5}$$

$$\mathbf{F}_{prior}(\mu) = \begin{bmatrix} F(\mu|l_1) \\ \vdots \\ F(\mu|l_m) \end{bmatrix}, \tag{5.6}$$

so that querying the c.d.f.s at all pixels corresponds to the matrix operation:

$$\mathbf{F}(\mu) = \mathbf{W}_{\theta,I} \cdot \mathbf{F}_{prior}(\mu). \tag{5.7}$$

Then, we can estimate the quantiles through uniform search, according to the simple Algorithm 1. We use Python's NumPy library [188] for an efficient implementation of matrix operations. Using high accuracy requirements ($\Delta\mu = 0.02$), the algorithm takes a couple hundred milliseconds to complete.

Algorithm 1 Estimating friction quantile images through search

```
1:  procedure QuantileSearch(η, Δμ)        ▷ Δμ is the error tolerance
2:      μ ← 0
3:      finished ← false
4:      while not finished do
5:          F(μ) ← W_{θ,I} · F_prior(μ)
6:          mask ← F(μ) ≤ 1 − η
7:          Q[mask] ← μ                              ▷ Update quantiles
8:          μ ← μ + Δμ
9:          finished ← mask= 0          ▷ Did we compute all quantiles?
10:     return Q
```

An alternative to the search algorithm is to approximate the Gaussian mixture distribution by a single Gaussian distribution. In that case, at each pixel only the mean and variance of the distribution need to be computed, which in our experiments was an order of magnitude faster than Algorithm 1. However, the loss in terms of accuracy is high (RMSE=0.1), and it might be preferable to simply increase the error tolerance in Algorithm 1 for a better trade-off between accuracy and computation speed.

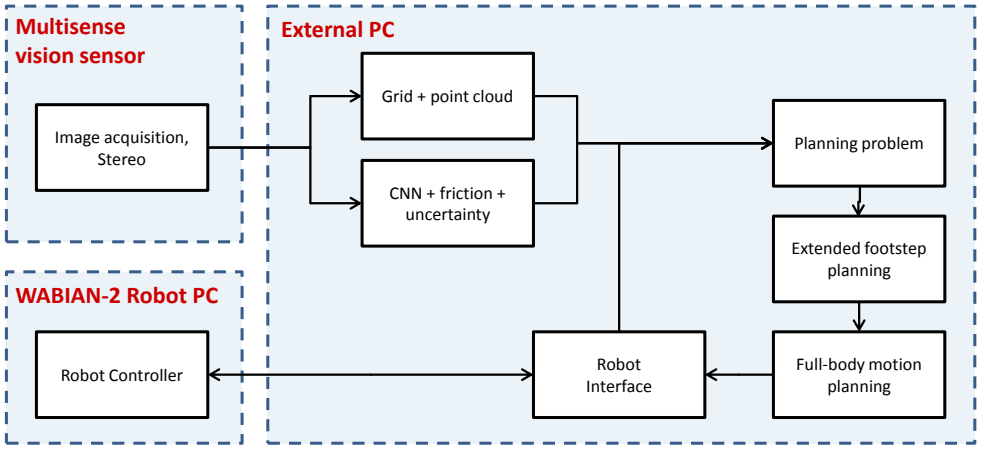


Fig. 5.1 System architecture

5.2.2 System integration

We integrated the planning and perception algorithms described in this thesis into a single system architecture, shown in Figure 5.1.

a) Hardware and communication

Robot. We use the humanoid robot WABIAN-2, which is described in Section 2.3.2. We replaced the original neck and head by the Carnegie Robotics Multisense-SL sensor-head and a neck support. All robot motors and sensors (joint encoders, photosensors, force sensors) are monitored and controled from a computer which is mounted on its backpack - to which we call *robotPC*.

robotPC. Has access to sensors and motors through an interface board. Is responsible for executing full-body trajectories. Sends the state of the robot and gets trajectories from the *externalPC* through a LAN connection. Uses a real-time operating system (QNX) for reliable robot control.

robotMultisense. Consists of two cameras, inertial measurement unit, and laser rangefinder. Stereo matching is computed onboard and data is sent to the *externalPC* through a LAN connection.

externalPC. Is connected to *robotPC* and *robotMultisense* through LAN connections. Is reponsible for collecting robot and vision data, planning full-body trajectories and sending them to *robotPC* for execution.

b) Modules and libraries

Stereo. Runs on *robotMultisense* as a ROS node. Publishes images, stereo matching costs and WTA depth maps.
Friction. Runs on *externalPC* as a ROS node. From input images, uses a deep CNN with trained parameters to estimate the p.d.f. of material at each pixel, then estimates the $(1 - 0.95)$-quantile of friction per pixel and publishes this image. Uses libraries Caffe, CUDA, OpenCV.
OccupancyGrid. Runs on *externalPC* as a ROS node. From input stereo data, updates and publishes an occupancy grid as a point cloud of the grid's centers. Uses the OctoMap library / ROS node.
ProblemPublisher. Runs on *externalPC* as a ROS node. Collects the robot state, point cloud, friction image and locomotion target, combines the data into a "planning problem" - consisting of a friction-annotated point cloud in the world reference frame, start and goal positions. Uses libraries OpenCV and PCL.
WabianPlanner. Runs on *externalPC* as a ROS node. From input planning problems, it plans a full-body trajectory using the robust hierarchical planning algorithm, then publishes the trajectory. Uses libraries SBPL, trajopt, OpenRAVE.
WabianInterface. Runs on *externalPC* as a ROS node. Collects robot state data from a TCP connection with RobotController (*robotPC*) and republishes it as ROS topics. Sends input trajectories to RobotController (*robotPC*) through a TCP connection.
RobotController. Runs on *robotPC* as a process. Collects sensor data and updates motor commands on a real-time 1ms-loop thread. Sends the robot state to WabianInterface (*externalPC*) on a separate thread. Obtains trajectories for execution from WabianInterface (*externalPC*), along with the instant of time at which the trajectories should start. These are checked for consistency with the current trajectory in execution and added to an execution queue.

5.2.3 Real-robot results on a mock-up scenario

We prepared a mock-up scenario in the laboratory which demonstrates the capabilities of our planner. The scenario consists of a floor with two areas of different materials. One is made of wood and the other is a high-friction flooring resembling ceramic tiles both in appearance and coefficient of fric-

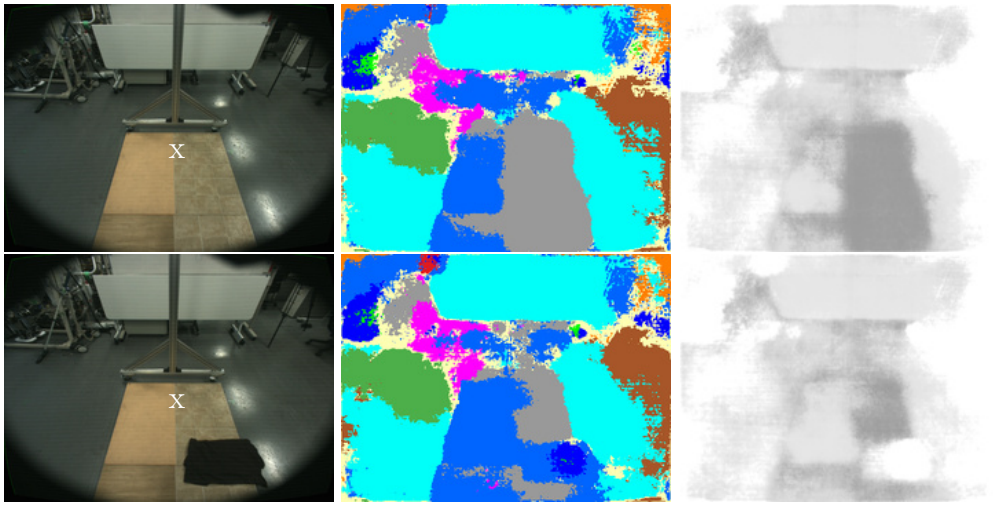


Fig. 5.2 Mock-up scenario and friction from vision. Left: the view from the robot's camera of the mock-up scenario built in the laboratory. The locomotion target is one meter ahead, marked with a white "x". Middle: material segmentation. Right: coefficient of friction quantiles $Q_{1-0.95}$ of equation (3.9). Darker shades of gray correspond to higher friction, such that white is $\mu = 0$ and black $\mu = 1$.

tion. The perception-planning algorithms were run on this scenario, and then a piece of cloth (T-shirt) was laid flat on one of the surfaces to provoke changes in friction and force a different plan. See Figure 5.2 for the scenario, segmentation and friction as seen from the robot's camera at the initial condition. Once again, the figures show the advantage of using the full probability distribution of materials given by the CNN. While *cloth* is the highest-ranking material only in part of the object region, friction is low on a larger region which is highly consistent with object borders.

The robot starts in double-support, with one foot on each surface. The goal stance is one meter ahead, also with a foot on each surface. After the robot is placed at the initial state, the perception and planning algorithms run without any human input except the push of a button to execute the planned full-body trajectory open-loop. Trajectory optimization parameters are the collision penalty weight α of equation (2.9), which is set to 50, and the distance at which the collision penalty starts being applied (for all links except those in contact), which we set to 2.5cm. The obtained full-body trajectory is tracked by position control at the joint level.

For these experiments we customized the WABIAN-2 robot (described

in Section 2.3.2) with a Carnegie Robotics Multisense SL sensor-head. The perception pipeline predicts pixel-wise material label distributions and pixel-wise friction using SegNet [149] and equation (3.9), and combines them with the stereo depth maps computed onboard by the Multisense. It produces friction-annotated point clouds at 2Hz. For collision checking, the point cloud is converted into a mesh using the fast surface reconstruction algorithm of [189] as implemented in PCL [124]. All perception and planning computation ran on the external PC (connected to the robot's onboard PC), and we used ROS [190] for communication.

We show the results of the perception-planning experiments in Figure 5.3. From left to right, we show the material and friction point clouds, the footstep plan, the collision-checking bounding boxes used by the footstep planner and the final planned full-body trajectory after optimization and stabilization. In the first situation there are only wood and ceramic surfaces, but the predicted lower bound of friction of the wood surface is lower than that of the tiles ($Q_{1-0.95}$ =0.1 vs 0.4). The footstep planner returns a sequence of stances that reduces the amount of times wood is stepped on. This behavior comes naturally from the *extended footstep planning* approach [191], since walking on low friction ground requires higher stance times (slower motion) and thus more energy cost. Furthermore, note that the trajectory optimization uses all degrees-of-freedom to satisfy the constraints (e.g. trunk roll use is clear in the image sequence, important mainly for the stability constraints), and that the knees are relatively stretched in order to reduce torque consumption but still satisfy stability constraints. Also note that swing leg clearance happens automatically due to the use of collision costs.

In the second situation we laid a flat piece of cloth on a ceramic spot used by the previous trajectory. The cloth was correctly classified and its friction was practically zero. The footstep planner returned a trajectory around the cloth and on the wood surface, which led to a slightly longer time and energy cost of the full-body trajectory (63 vs 60 seconds, 5% longer than on the first situation). Note that while the times are long they correspond to 25 stances because of step length limits, and thus the average time per stance is approximately 2.5 seconds.

Footstep planning took approximately 20 seconds in the first situation and 10 in the second. The reason for the difference is clear from the scenario: while in the first situation stances on both surfaces are expanded by A* in order to guarantee optimality, in the second situation no stances are

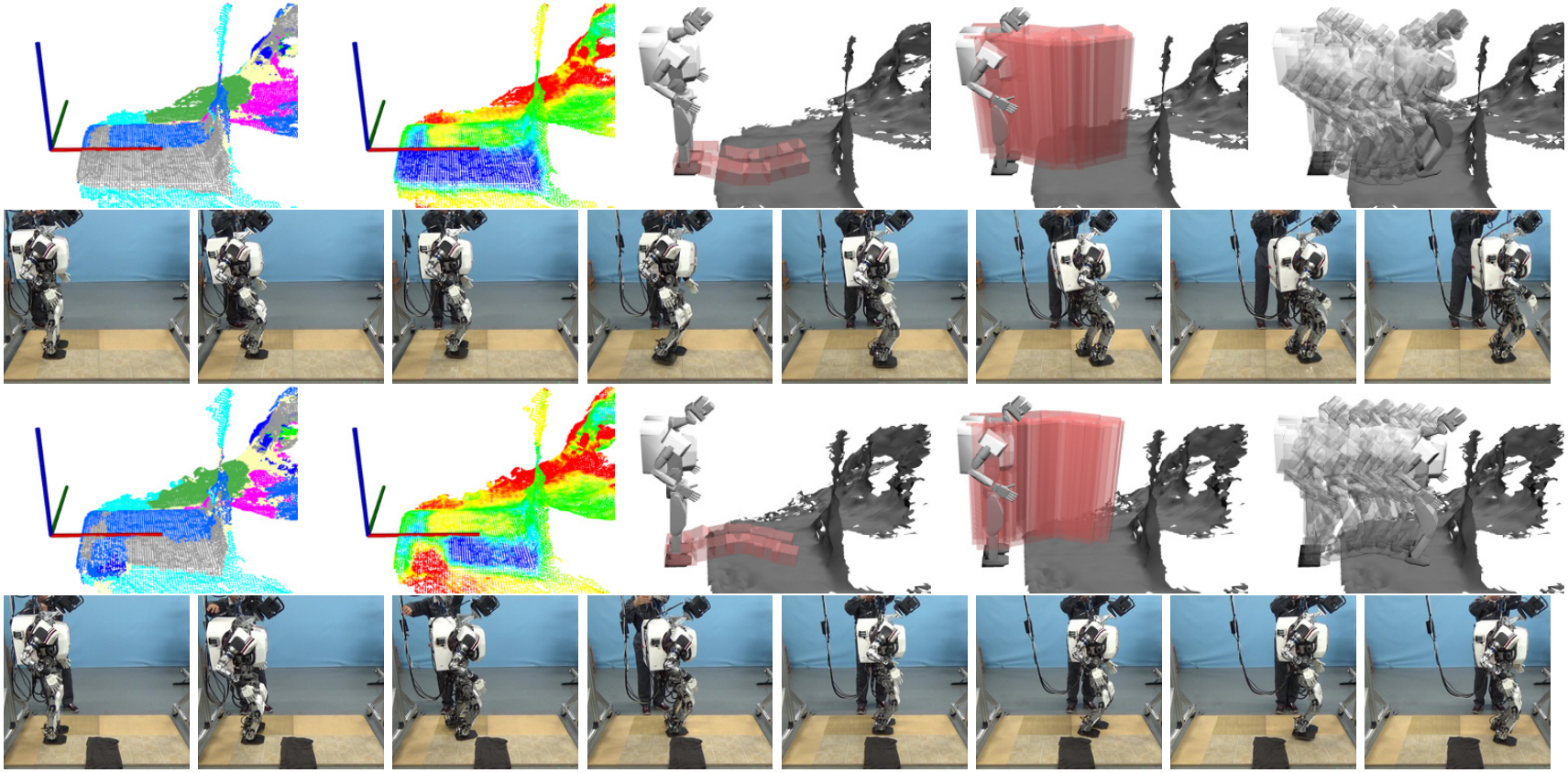


Fig. 5.3 Perception, planning and locomotion results on the mock-up scenario. First two rows: two surfaces (wood and ceramic). Last two rows: three surfaces (wood, ceramic and cloth). We show the material segmentation (same colour codes as in Figure 5.2), friction (cold colours are high friction, warm are low), footstep plan, collision bounding boxes, full-body plan and finally the walking sequence on the real robot.

expanded on the surface with cloth since friction zero has infinite cost. Full-body trajectory optimization took approximately 40 seconds and dynamic stabilization 2 seconds. Note that these are for 25-stance, 60 second trajectories, and therefore they should be considerably faster in case planning is done one or two steps at a time.

5.2.4 Simulation results on a real-world outdoors dataset

We also applied the friction-from-vision and planning algorithms to larger and realistic outdoor scenarios. We did this while avoiding the logistics of transporting the real robot (and its crane, battery charger, computers, etc) by acquiring a 3D dataset of real outdoor environments. We ran the same perception and planning algorithms on the data, and then analyzed the resulting trajectories without actually executing them on the real robot.

The gathered dataset consists of stereo pairs acquired with a consumer-level stereo camera. We used a FujiFilm FinePix REAL 3D W3, a compact-sized camera with 10 megapixels and a 75mm baseline. The camera was calibrated using a 10-by-10 squared chessboard pattern and the stereo calibration functions of the OpenCV library [179].

We started by acquiring several high-quality stereo pairs on the streets and parks of Tokyo, Japan. Then, we obtained 3D reconstructions from the pictures by applying the block-matching stereo algorithm implemented in OpenCV. To recover the direction of gravity we assumed that the largest plane in the scene was always a horizontal ground plane - which we tried to guarantee during dataset collection by appropriate framing. That plane was automatically segmented from the 3D reconstruction through RANSAC-based plane segmentation using PCL [124], and the world reference frame thus placed with the vertical direction aligned with gravity (i.e. the Z vector perpendicular to the largest plane). The CNN-based friction estimation algorithm of Section 3.5 was applied to the left image of the stereo pair, and the resulting friction quantiles merged with the 3D construction to form friction-annotated point clouds for planning.

Figures 5.4 to 5.9 show the dataset pictures and results. Each figure consists of the original picture, the 3D-reconstructed scenario, the friction-colored scenario together with the footstep plan, and an image sequence of the planned trajectory. The friction-colored scenario represents coefficient

of friction values in hue space, so that cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

In the scenario of Figure 5.4, grassy and leafy areas are predicted to have lower friction. We set the locomotion target to the inside of the grass patch, on the right side of the tree. The planned trajectory avoids the dirt on the way to the target, and makes a curve when approaching the grass in order to reduce the distance covered over grass. This curve is especially visible on the footstep plan image. In Figure 5.5, the robot climbs dirt at a construction site. Due to matching errors in stereo, traffic cones were not correctly reconstructed as obstacles but as being part of the ground surface, and thus the planned trajectory erroneously crosses them. In Figure 5.6, the robot climbs down a ramp and goes up stairs at low angles, while avoiding collision with the handrail. In Figure 5.7, the robot starts in the middle of a dirt patch and gets out of it to the asphalt by a trajectory that takes the shallower slope (instead of a steep step). In Figure 5.8, the robot starts in front of a metal pole, and walks to a traffic cone directly behind it. This scenario is similar in concept to our real robot experiment of Figure 5.2. The target can be approached either through the surface on the right (high friction ceramic) or through the surface on the left (lower friction asphalt). The final trajectory opts for the higher-friction path while still avoiding collision with the poles. Finally, in Figure 5.9 the robot avoids a straight path that would involve walking over a wet surface of low friction, by taking a curved path through grass.

5.3 Discussion

We will organize the discussion of this chapter according to the objectives we set in the beginning.

5.3.1 Robust planning

We showed that planning can be made robust to uncertainty in friction estimates using chance constraints. Using the method described in Section 3.5 a full p.d.f. of friction is available, from which we can also estimate the probability of satisfying Coloumb friction constraints. That probability can be used as a parameter for conservativeness.

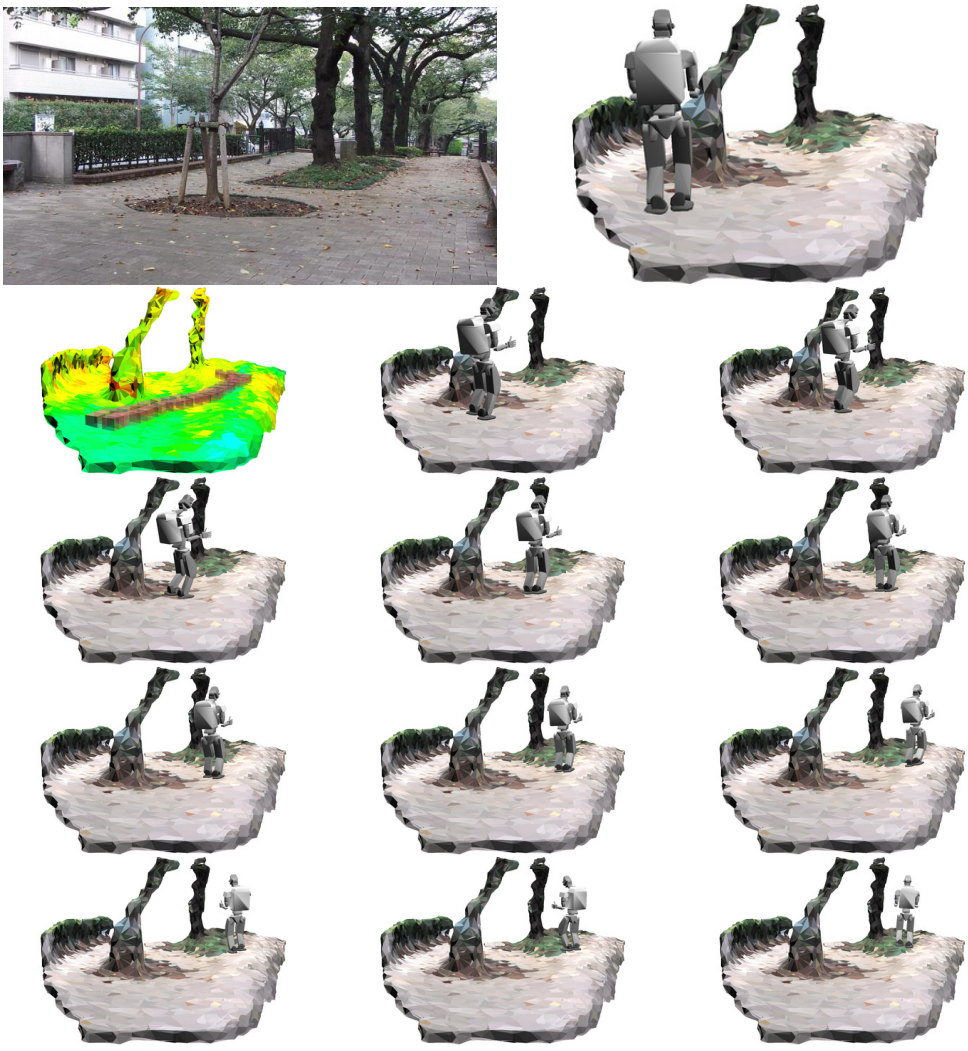


Fig. 5.4 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

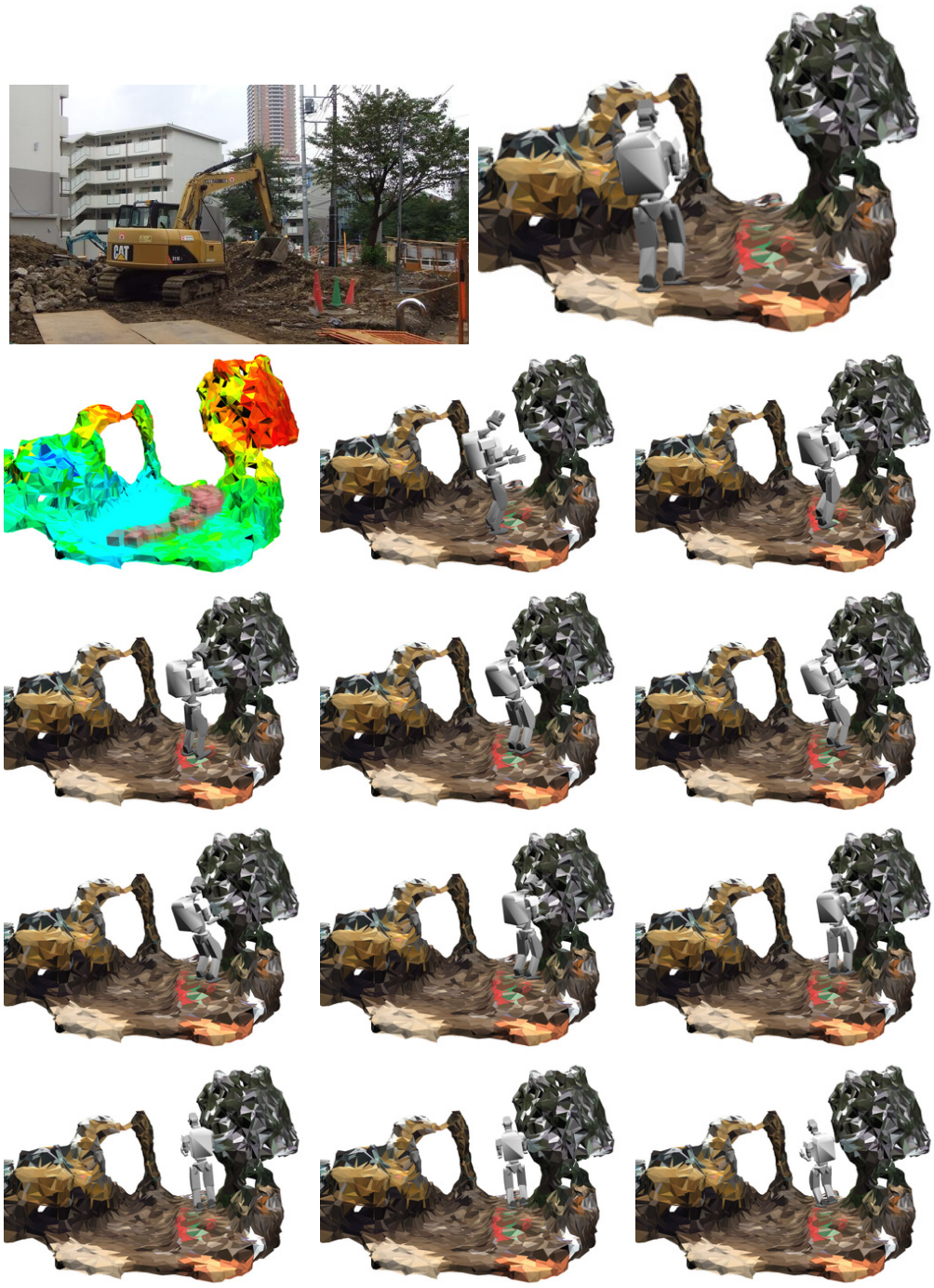


Fig. 5.5 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

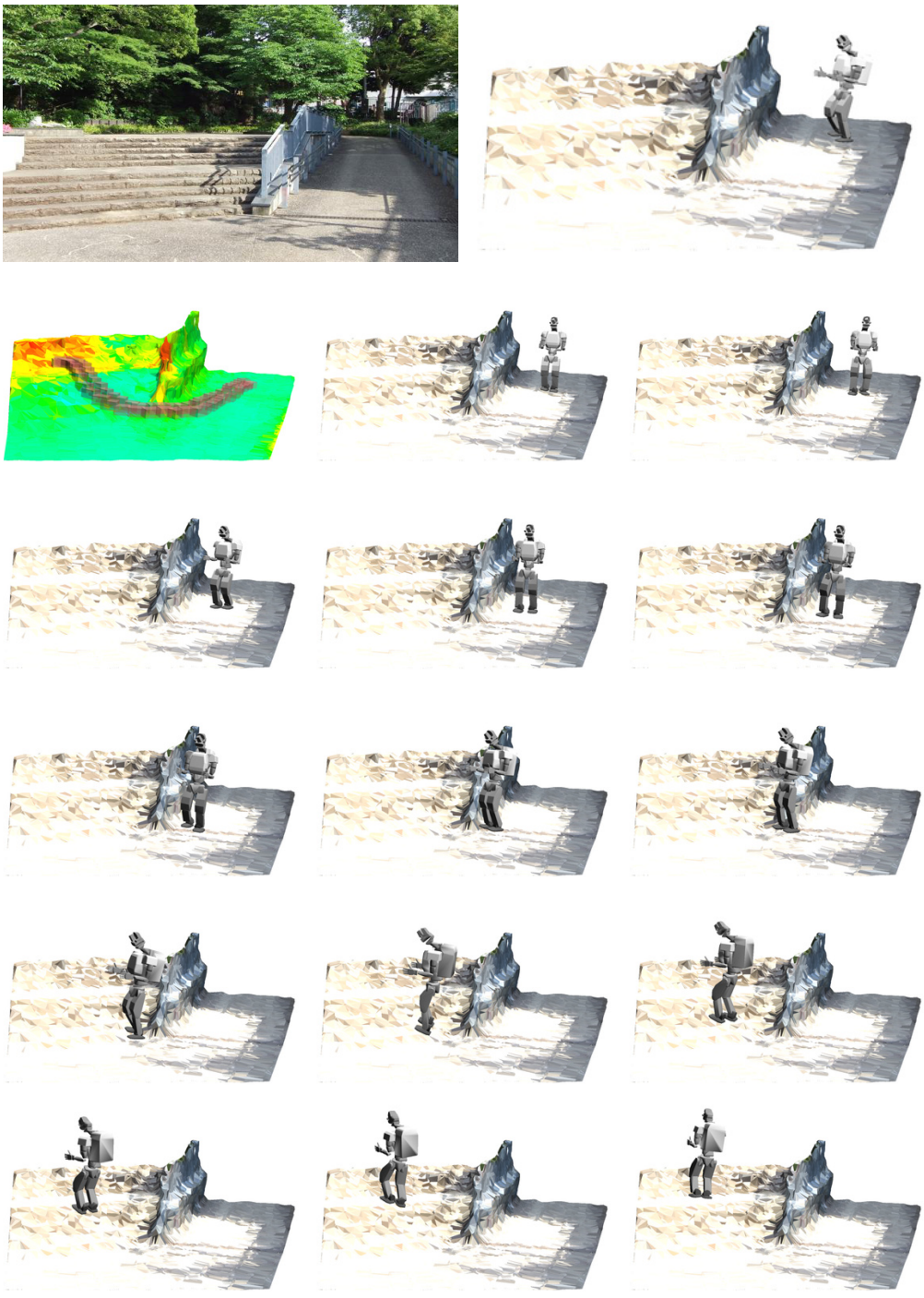


Fig. 5.6 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

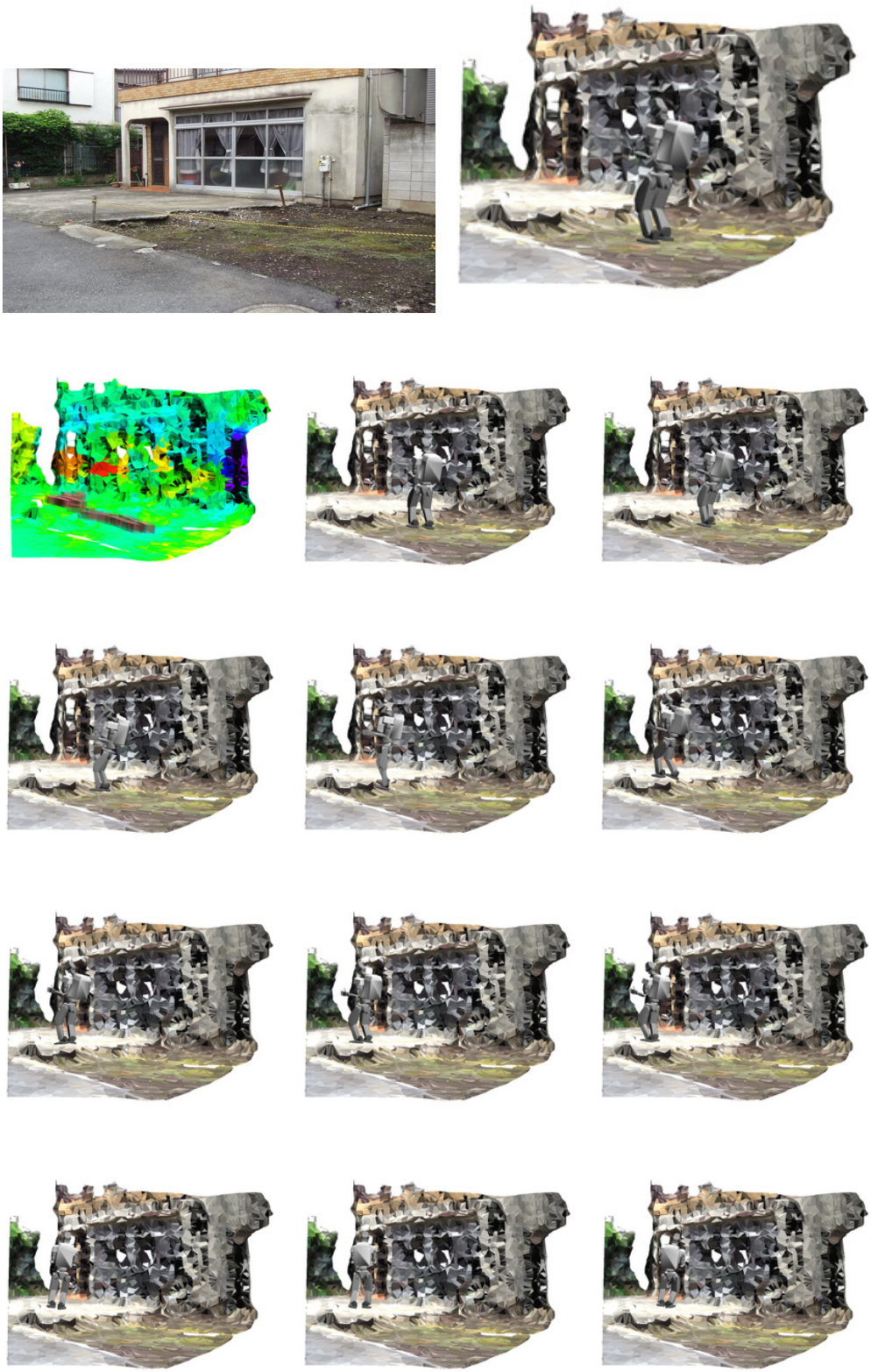


Fig. 5.7 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

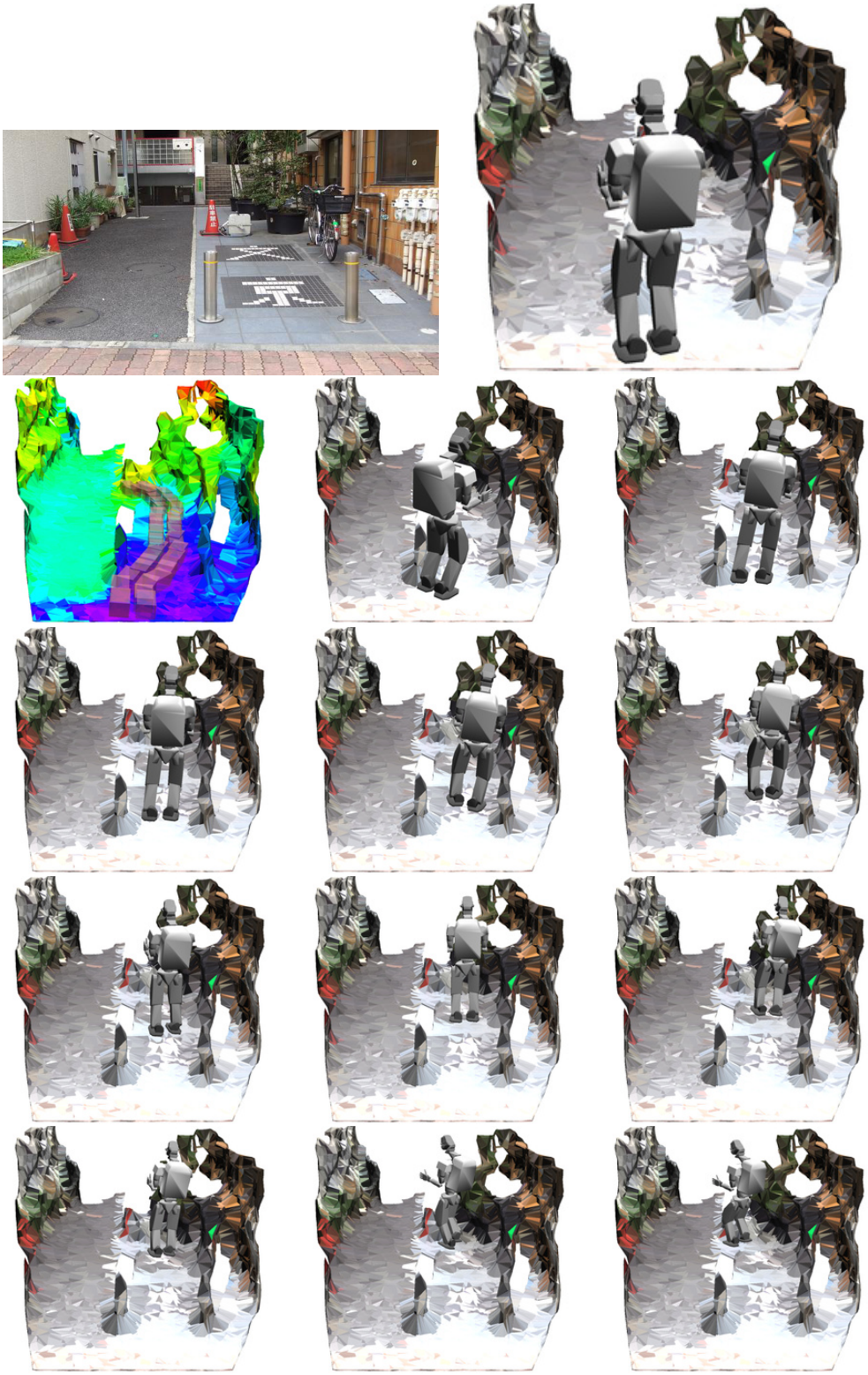


Fig. 5.8 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

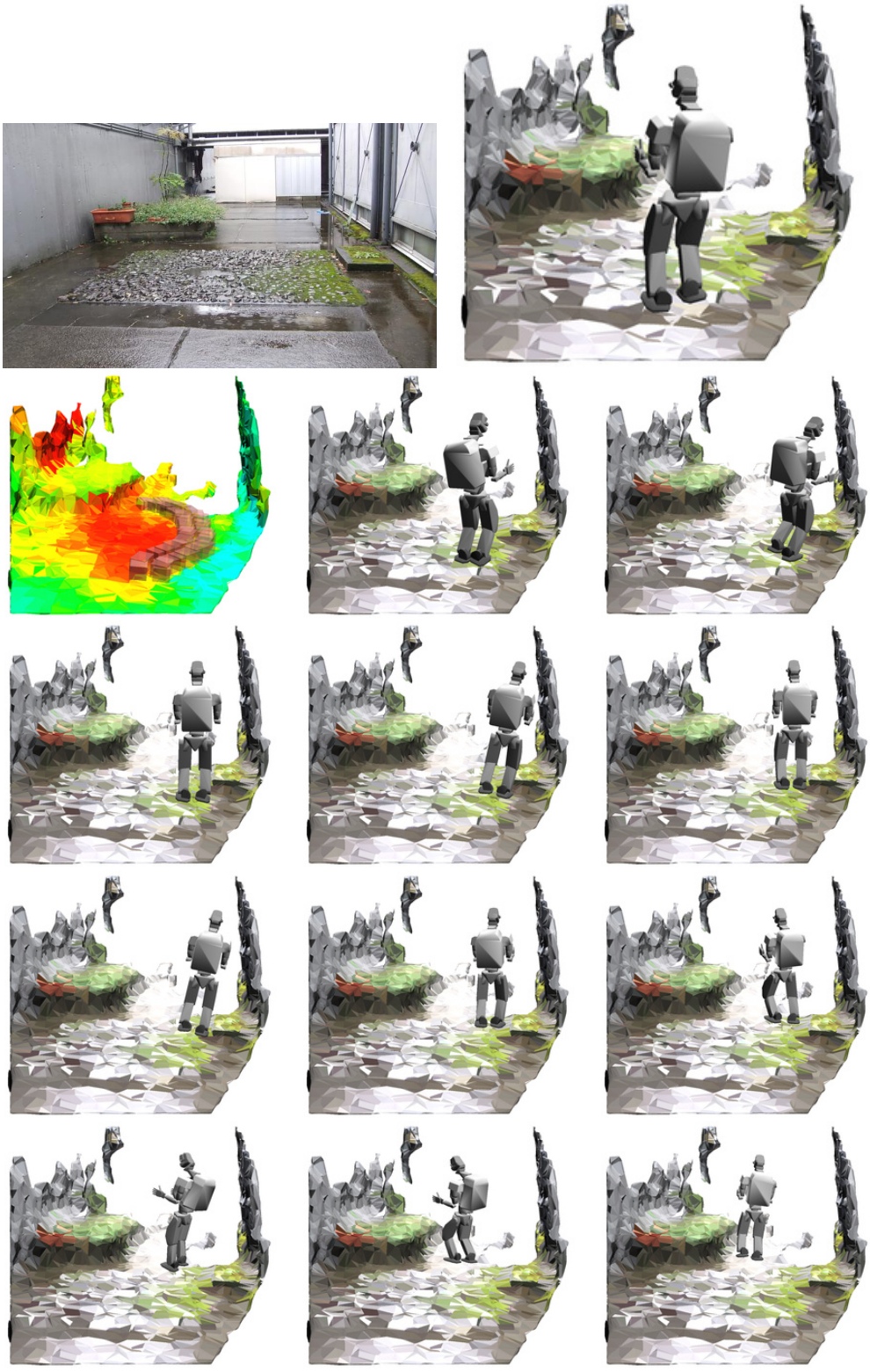


Fig. 5.9 Simulation results on real stereo data. In order: original picture, 3D-reconstructed scenario together with initial pose, friction-colored scenario together with footstep plan, image sequence of planned trajectory. Friction coloring: cold colors are high friction and warm are low (i.e. red is $\mu = 0$, blue $\mu = 1$).

Our formulation assumes that RCOF and energy models are not corrupted by noise, but model uncertainty could be considered using chance constraints as well. This would come at the extra cost of computing a convolution between the two p.d.f., or using a convex approximation of the constraints.

5.3.2 Whole system evaluation

We showed that locomotion planning considering world geometry and friction can be fully automated on a real robot at both perception and planning levels. These friction-aware algorithms are relevant since not only obstacles but also different terrain types abound in the real world, and locomotion choices should take them into account - whether for safety or energetic considerations.

Our implementation of the whole perception pipeline can be executed at 2Hz, which is arguably sufficient for high-level planning. Footstep plans are provided every 10 seconds, and the full-body planning stage takes around 1.6 seconds per stance. Planning time is considerably slow, and so speeding up computation is one important direction of research. There is also a clear hierarchy in both representation and computation time, which can be exploited by successively lowering the planning horizon at each level. For example, with the current implementation timings, if stances last on average 2 seconds then for continuous walking the footstep planner should plan 5 stances at a time, and the full-body planner one or two stances at a time. Ideally, there should also be one extra layer of planning at the COM level to guide footstep planning.

Importantly, another problem with the proposed perception and planning approach is that wrong material classifications can lead to there being no solution to the footstep planning problem. An example of such a situation is when a material the robot cannot walk on, such as water in our case, is mistakenly given very high confidence. Our view is that the solution could be semi-supervision where a teleoperator can correct a segmented region's material label. However, as we concluded in Chapter 3, inexperienced teleoperators should not directly annotate COF.

The results of Section 5.2.4 show that the perception and planning algorithms introduced can not only be applied to simple laboratory scenarios but also complex outdoor scenarios. In general, paths avoided obstacles and

dealt with slippery terrain and slopes. However, the results also show the consequences of not considering stereo uncertainty during planning: one of the paths (Figure 5.5) actually traverses obstacles as if they were not there because of a mismatch in stereo. An important avenue for future research then, is the integration of 3D-reconstruction uncertainty into planning. For example, using stereo matching likelihoods or occupancy probabilities to frame probabilistic collision constraints.

5.4 Summary

In this chapter we extended the planning algorithm of Chapter 2 in order to enforce robust friction constraints. We showed that uncertainty in the coefficient of friction estimates can be integrated into extended footstep planning without increasing problem complexity by using chance constraints. Then we introduced an architecture for the full perception-planning system on a real robot, and demonstrated the system on a real scenario. The real robot was able to correctly perceive surface friction, as well as optimally plan footstep placement and motion to reduce energy under friction constraints. We finished with an evaluation of the perception-planning system on a set of real-world scenarios to demonstrate the applicability of the system to a wide range of environments. Overall, this chapter showed that the perception and planning algorithms of this thesis are applicable to challenging scenarios with varying surface friction, slopes and obstacles - both in laboratory-made environments and realistic outdoor scenarios.

Chapter 6

Conclusion and discussion

6.1 Contributions of this thesis

6.1.1 Technical contributions

In this thesis we proposed a complete solution to the problem of biped robot locomotion on environments with complex geometry and slippery terrain. We developed:

a) a robust and objective-consistent hierarchical planning method which considers energetic trajectory costs, friction, kinematics, collision and stability;

b) a visual friction estimation algorithm which provides uncertainty estimates for robust planning; and

c) a geometry estimation algorithm which accumulates stereo and its uncertainty over time for high precision and visual robustness.

One important contribution of the planning methods we introduced is that they can reason about friction in ways that were not possible before. For example in the ice puddle scenarios we used for evaluation in Section 2.4.2, the planner automatically opts between avoiding or slowly traversing the ice. Such scenarios would not be solvable by previous footstep planners unless slippery surfaces were treated as non-traversable - in which case the trajectory could be energetically inefficient when compared to our friction-aware slow-traversal approach. Friction-aware footstep planning was made possible mainly by the use of step times as variables estimated from step

placement and surface friction. The fact that we plan step times is also innovative - with few exceptions, most state-of-the-art humanoid locomotion planners still assume fixed step times, which can highly limit the motion of the robots. Finally, we also made footstep and full-body motion planning more tightly coupled and consistent when compared with previous contact-before-motion approaches. This was possible by the concept of "oracles" or "learned models" which predict the cost and RCOF of footsteps obtained at the end of the whole (full-body) planning pipeline. These are learned offline by feeding the full-body motion planner with a large number of stance transitions and saving the results on a hash table for online use.

The friction from vision algorithms we described, especially the "material CNNs with material friction p.d.f.s" algorithm of Section 3.5, is also a pioneer in the field. While some state-of-the-art control and planning frameworks now include friction in the models, the authors of these methods do not estimate the coefficients of friction yet, but only assume them to be constant. Our work on friction estimation is thus an important and large step towards the actual and practical use of friction during planning, since we prove not only that it is possible but quantitatively estimate the error expected out of these estimates. Since our method can provide a whole probability distribution of friction, it will hopefully find its way into trending robust planning and control frameworks.

Our treatment of stereo reconstruction in Chapter 4 also follows a similar probabilistic approach. While the common methodology in the robotics community is to discard the uncertainty in stereo matching early on and use disparity images to build occupancy maps, we instead estimate these maps from the full p.d.f. of stereo matching costs. This is actually the main common principle we use for both visual estimation of friction and geometry: to avoid making early decisions on uncertain variables (e.g. on material class or stereo disparity) but instead keep the probability distributions until the end of the perception pipeline. As we showed in both cases, this approach is advantageous. In the friction case, friction quantiles were more consistent with object margins than highest-confidence material images because true materials were still given high confidence by the segmentation algorithm. In the geometry case, we obtained higher grid precision in outdoor scenarios, visually repetitive scenarios and noisy images, because when the least-cost disparity is wrong, true disparity is still given high confidence by the occupancy grid algorithm. We believe this thesis also contributes to demonstrate

this advantage and hope that the community will follow.

6.1.2 Impact and applicability to different fields

We would like to stress that the methods introduced in this thesis have broad cross-disciplinary applications. Starting with the "extended footstep planning" method and in general our approach to locomotion planning with oracles, the approach is applicable not only to humanoids but legged and mobile robots in general. Most robot locomotion algorithms start by some sort of high-level planning representation such as static bounding boxes that discard the different ways the robot can use to get to target states. Using our approach these can be parameterized, the respective energy/required-friction learned by physics simulation of the whole planning pipeline, and final models (i.e. oracles) used online by the high-level planner for energy and friction considerations.

Our friction from vision work can be applied straightforwardly to robot manipulation, where grasp parameters depend on COF estimates which could be provided by our algorithm. Since robots can also repeat manipulation experiments more easily at scale, when compared to walking experiments, the probability distributions of material friction could in that case be more easily obtained through autonomous robot learning behaviour instead of the manual experiments we used in Section 3.5. Furthermore, we can envision the friction algorithm being applied to well-being applications for the blind and people with low-visual-acuity, such as assistive devices that warn or provide safer locomotion paths to the user.

Finally, our new data-driven approach to stereo confidence measures, such as the new histogram-based model trained on stereo datasets with ground-truth, or the maximum-likelihood-estimated parametric models, can be applied to global stereo methods in general, and visual SLAM methods in general. We believe these methods could all benefit from better stereo confidence measures and from the conclusions we reached in Section 4.5.

6.1.3 Insights for robotics and vision

The experiments conducted for the purpose of this thesis brought forward several useful insights for the robotics and vision communities. Namely:

a) Friction should be considered in humanoid robot locomotion planners,

not only controllers. Considering it can lead to both lower falling risk and lower energy consumption

b) Including timing variables in footstep planning is crucial for friction, energy and stability considerations

c) Human-inspired gait principles applied to footstep planning have slippage and electrical energy advantages, as well as leading to human-like motion

d) Humans are not good at the task of predicting friction of a robot foot in various terrain. A single teleoperator system, or even crowd-sourced teleoperation, should not involve friction estimation but, more usefully, material label correction

e) Material is a highly predictive feature for friction estimation. It is important for robotics to build large-scale datasets and algorithms for material segmentation

f) Stereo uncertainty models (i.e. confidence measures) should be carefully chosen, estimated and integrated into time filtering algorithms for better performance and visual robustness

6.2 General discussion

6.2.1 Planning vs control

In this thesis we focused on planning algorithms for humanoid locomotion, generating open-loop full-body robot trajectories that are predicted to be low-energy, slippage-free and collision-free. We did not especially develop control algorithms to track these trajectories with guarantees, or to avoid falling when the robot actually slips, which is another important and active field of research. One of the reasons we opted for this route was to complement existing feedback control-centred locomotion approaches [58, 89, 90]. The motivation is that feedback control's local adaptation of tangential-to-normal force ratios may not be sufficient in very low friction surfaces, and so the global trajectory itself should be better planned so that we can tackle general scenarios. We do this by predicting the performance of the controller as a function of step parameters, using an oracle for footstep planning. The performance of our planning algorithm should thus also improve with the

introduction of better full-body controllers which consider dynamics and friction.

Since we planned full-body trajectories for the whole path from start to goal, even when these were fairly distant (e.g. 1 to 3 meters), planning was relatively slow. This begs the question of whether we are planning too far ahead. More importantly: how far ahead should full-body motion be planned? What about footstep plans? Are we using too much detail at the full-body planning level? These are recurring questions in robot motion planning. We believe deep planning hierarchies are an interesting answer to the computational speed problem. For example, planning at the point or bounding box level, followed by footstep level, followed by full-body. Or instead, planning base motion at first with all joints fixed and collision consideration, then successively refining the path by adding more joint freedom and constraints, in an approach similar to continuation numerical optimization [29]. Of course the issue then is how to define the hierarchy for best performance, and we believe the answer could lie somewhere between good designer intuition and machine learning. Different methods other than graph search for high-level planning are also an interesting option, such computing homotopy classes (i.e. topological constraints) to guide search [192], sampling or optimization [193] methods. In addition to that, it is important to develop efficient synergies between planning layers instead of strict hierarchies, something that is dealt with for example in [194].

Importantly, to reduce computation complexity, planning can be coarse but long-ranged for high-level descriptions and fine but short-ranged for low-level descriptions. While we did it for simplicity in this thesis, there was no need to plan the footstep or full-body trajectories from start to goal - they could be computed for the next few steps successively as the robot is walking and more visual input is received, thus creating several levels of feedback as well.

6.2.2 Model-based vs model-free planning

In this thesis we opted for a model-based approach to planning by using precise robot models which include all link masses and at some point use analytic inverse kinematics for dynamic stabilization. This approach is a popular one in humanoid robots due to the robots' complexity and high cost, which make them hard to control and risky to experiment on without huge

financial and time expenses. If robots were robust enough to fail, fall and autonomously repeat experiments, then alternative model-free reinforcement learning techniques or other black-box machine learning-based techniques would also be a competitive approach to the problem. The main advantage in that case would be to avoid the simplification and modeling errors inherent to the model-based approach. Of course the model-free approach would not need to be run from zero on the real robot, but warm-started from a training stage in simulation. Even in that case, the real platform would need to be ready to fall several times before the policy would converge to something stable. This is something most platforms are not designed for, including the one we used in this thesis (WABIAN-2). However, it is still a promising field of research, especially with recent developments in the reinforcement learning literature [68] and as more fall-robust robots are built.

Contact models in thesis were also model-based. In particular, we used Coulomb's physical contact model to include friction in planning as commonly done in the humanoid literature. Still it is arguable that physical contact models themselves are not precise. As stated in Ruina and Pratap's book [195], "Not only can't you know the coefficient of friction between any two pieces of steel with any certainty, you also can't even trust the concept of a coefficient of friction to be very accurate", and because of that theory and practice typically differ by 5 to 50% [195]. For example, the coefficient of friction between our robot foot and any given surface, measured at the same spot and at the same foot orientation, had a SD of 0.05. Each experiment is slightly different, either due to small contact-area variations, dust, applied force profiles, etc.

So while physical contact models are not precise, they are intuitive and reusable between robots. The alternative could be learning black-box models of physical contact by interaction for each robot, which could have more accuracy in the long run. The same problem of small data we just described applies here once again though, since most current humanoid robot platforms are not ready for the amount of training trials that would require. Despite that fact, this avenue is a crucial one for the improvement of robot robustness, especially after model-based planning and control algorithms have been better understood.

6.2.3 Direct vs indirect perception

This thesis adopts an "indirect perception" model of perception and action. In other words, perception is "top-down" - mediated by knowledge which is coded in the robot [196]. This is opposed to the "direct perception", "bottom-up" model, in which perception is mediated by past experience and data. This discussion is closely related to that of model-based and model-free control.

In indirect perception, exemplified by Marr's important work [197], perception can be studied and implemented independently of action, i.e. the control system. This is what we do here. We develop friction and geometry estimation methods, combine them into a representation (a friction-annotated point cloud), and then plan motion on it. This is a strict hierarchical model of perception and action in which the planner or the controller do not influence perception. This of course need not be the case and is arguably inflexible. If the robot slips or detects the current surface has a different coefficient of friction from that which was predicted by vision, this information could feed back into the visual system to update the whole surface (e.g. using material segmentation regions) with a new p.d.f. or prior for friction. Such an interconnected model of perception and action is closer to the ideas of direct perception. Pure direct perception, as defended by Gibson's work [198], would lead to a model-free controller of locomotion where sensor input translates directly to actions or action modulators (he calls these associations *affordances*) without explicit intermediate blocks of "friction from vision", "geometry from vision", "planning", etc, which are typical of top-down approaches such as ours. The best performance might come out of a combination of both worlds, such as these feedbacks into vision that we just mentioned and other interactions between perception and action layers.

6.3 Limitations

6.3.1 Serial design, strict hierarchy

One of the limitations of this thesis is tightly related to the general discussion of "indirect perception" we touched on the previous section. Our planning approach in this thesis has a strict hierarchy of processing from perception

to plans which can lead to inflexible behaviour. For example, wrong material classifications can lead to there being no solution to the footstep planning problem. In addition to that, there is no feedback from the controller to the vision system in order to update friction predictions of surfaces which "look like" the one the robot is currently walking on, or walked on at some point in time. We believe this kind of feedback and memory mechanisms can greatly improve friction prediction and locomotion performance in the real world.

6.3.2 No dynamics in full-body planning

Another limitation of our planner is the absence of dynamics and friction consideration in the full-body motion planner. Because of this, even if the statically stable motion of the full-body planner is subsequently made dynamically stable by another algorithm, the resulting motion is overly conservative. For example, the COM needlessly lies inside the support polygon and in order to achieve this knees are bent more than necessary as well. In addition to that, since the dynamic stabilization algorithm will locally adapt joint motion, the full-body planner must assume tighter margins on joint angle limits so that these are not crossed after the stabilization stage. This further constrains the final solutions which might seem overly static. In any case, our planning architecture in itself is not limited and simply improving the full-body planner to consider dynamics and friction will suffice to solve this limitation.

6.3.3 Computational speed of planning

The computational speed of our footstep and full-body planners as currently implemented is arguably slow for real applications. In our experiments, the planning pipeline of footstep and full-body planning for 25 stances can take up to 1 minute when solved to optimality. There are several reasons for this. One of them is the currently large branching factor of footstep planning, due to the fact that all feasible neighbour points of a foot in contact will be considered to create a new stance. One way to attenuate this problem is to reduce point cloud resolution, another to introduce randomness by sampling. The low speed of footstep planners is a problem of current methods in general, and the 10 second planning times we obtained in our

experiments are actually common with other search and optimization-based approaches.

Our footstep planner is also relatively slow because it is being solved to optimality ($\epsilon = 1$ in A* search). So another possibility to reduce search time could be to stop search once a solution with a good enough sub-optimality criteria is reached. In addition to that, we current use an admissible heuristic for optimality guarantees. This can lead to slow convergence, and so if sub-optimality is not an issue then more effective heuristics could be used or learned with experience.

Regarding the current full-body planner, its low computational speed is mainly due to the computation of collision constraints which require evaluating mesh penetrations, and the optimization design itself (Sequential Quadratic Programming), which requires successive approximations of the objective and constraints and solving quadric programs until convergence. Planning speed could be improved by approximating the robot and environment meshes with simpler primitives (now the full meshes are used), by investigating other optimization designs, by warm-starting the planner with solutions which are close to a local optimum, and most importantly by planning motion not for all footsteps but for a shorter horizon.

As we discussed in Section 6.2.1, deeper planning architectures with successively shorter horizons could also make planning faster. Importantly, before footstep planning there could be a planning stage based on grids or similar representations of low dimension.

6.3.4 Uncertainty factors in planning

In this thesis we dealt with the problem of high uncertainty in friction estimates by considering that uncertainty during footstep planning using chance constraints. Friction perception is in fact a large source of uncertainty for locomotion according to our experiments. In Section 3.4 we measured the estimation error's SD of coefficient of friction to be close to 0.13 when materials are correctly labeled. Figure 2.5 shows that to accommodate for 1-SD of such errors the robot would have to walk up to 3 times slower, which corresponds to around extra 40J per step on flat terrain (Figure 2.4). Errors from stereo reconstruction are relatively smaller but also present. On the Multisense sensor-head we use, they are around 0.3mm at 1m distances, which is extremely small compared to the scale of footstep distances. At

10m distances the error is 3cm. In that case steps could contact the ground 3cm higher than expected which, ignoring extra energy expenses due to stability control, would lead to around 25J more energy per step (Figure 2.3). Although we did not account for this uncertainty in our footstep planner, that would be an interesting extension of our work.

In addition to that, there is also uncertainty in the energy and RCOF returned by the oracles/models. There is uncertainty due to the gap between simulation and real robot and due to the discretization of the space we make for speed (using hash functions). This uncertainty could also be considered, modeled for example as Gaussian noise added to the energy objective and RCOF constraint.

6.3.5 No motion in friction from vision

One of our conclusions was that humans' judgements of friction for a robot foot are poor. Such an observation matches previous work where COF was difficult to estimate for humans [135], and work which associates human falls to over-reliance on shine [136]. Nevertheless, our friction from vision datasets consisted of images only, from which the detection of specularities and gloss is made difficult due to the absence of motion. Humans' performance could have been overly bad due to a dataset bias which leads humans to make incorrect gloss (and consequently friction) estimates. In order to better quantify the reliance of humans on gloss, and its relationship to friction, we believe future datasets should include video clips of a moving camera over the surface of interest.

Another avenue of improvement is related to the CNN-based friction from vision algorithm, which estimates friction pixel-wise from each frame independently. Filtering the measurements over time, perhaps directly in the occupancy grid, could lead to improvements in smoothness and robustness to visual conditions such as the ones observed in geometry reconstruction (Section 4.4.3).

6.4 Future work and open problems

We will now point out several possible directions for future research related to this thesis.

6.4.1 More physical properties

One interesting research direction to be explored is that of considering more physical properties of contact surfaces, other than friction. For example, how to predict deformation of soft contact surfaces and its impact on stability and energy consumption is an important open problem. Previous research has modeled soft ground as springs with known stiffness and developed controllers for stable walking on such surfaces [199], however, our extended footstep planning or a similar approach could be used to *plan* better reference motion or avoid these surfaces altogether when predicted energy or instability is too high. Another problem is that of predicting terrain stiffness from images. Measuring terrain stiffness is more challenging than measuring coefficients of friction, so it is also important to find out whether an offline way to train predictors is possible at all. The solution might lie somewhere between a simple measurable model of terrain deformation and empirical human knowledge through engineer's coefficient tables, surveys or text mining.

6.4.2 Planning architectures

Several planning approaches to locomotion with contact have been proposed in the humanoid robot community, from search as in this thesis to sampling [26], to mixed-integer optimization [15] and continued optimization [29]. The number of design options grows even higher once we consider also the different representations and planning algorithms at each level of a hierarchical architecture, as well as different interfaces between layers. It is still not clear which out of the large number of possibilities is preferable, mainly because each researcher obtains planning results on a different environment. It is thus of extreme importance to create a common benchmark for planning on legged robots, consisting of a common set of environments, task specifications and robots. Only then comparison of the advantages and disadvantages of each design choice can be made clearer. Finally, due to the exploding number of possibilities for representation, planning and interface of planning layers, another interesting direction of research is that of designing algorithms to automatically evaluate, combine and optimize complete planning *architectures*, instead of parameters for a pre-established architecture.

6.4.3 Large datasets for friction from vision

The datasets we developed for the context of this thesis provided some insights on features with high predictive power for the friction-from-vision task. Still, the conclusions that can be taken from datasets of this dimension are limited, and non-linear end-to-end learning is not possible without over-fitting. One interesting direction of research then is that of developing large-scale datasets with thousands of friction-annotated images, so that algorithms such as CNNs can be trained end-to-end (i.e. pixels to friction). In addition to that, such large-scale datasets would allow to test non-linear models of human perception of friction with reliable significance statistics. Such datasets could include many more materials than those tested in this thesis, as well as labeled surface conditions, such as dry/wet, presence of fixed/rolling stones, etc.

The process of acquiring such dataset would be very time consuming, especially if the same manual-measurement procedure is taken. Other options include developing autonomous mobile robots that explore large environments collecting measurements, or special wearable sensors which capture data from a human's shoes throughout the day without human intervention. Taken to the extreme, a "complete" city could be mapped offline together with measurements of friction or other meaningful physical properties. Such a map could be used online for planning in a similar approach to that used in autonomous cars [200] but with extra material or physical property information.

6.4.4 Text mining for navigation in unseen terrain

One of the promising results of this thesis was the performance of text mining at the material friction prediction task. Our algorithm estimated the average friction between a reference material and all other materials which co-occur with it, but this need not be the case. One more interesting direction of research is thus to further explore this field, by estimating friction between two specific materials, using larger text databases, and estimating more environments properties - such as traversability, fragility, stability, etc.

References

[1] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima, and S. Kawatsuma, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[2] U.S. Bureau of Labor Statistics, "Census of fatal occupational injuries," tech. rep., U.S. Department of Labor, 2014.

[3] K. Tadakuma, R. Tadakuma, K. Nagatani, K. Yoshida, M. Aigo, M. Shimojo, and K. Iagnemma, "Throwable tetrahedral robot with transformation capability," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2801–2808, Oct 2009.

[4] P. R. Allison, "What does a bomb disposal robot actually do?," *BBC Future*, 2016.

[5] IFR Statistical Department, "World robotics report 2016 service robots," tech. rep., 2016.

[6] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-iii: An integrated humanoid platform for sensory-motor control," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 169–175, Dec 2006.

[7] T. Kishi, N. Endo, T. Nozawa, T. Otani, S. Cosentino, M. Zecca, K. Hashimoto, and A. Takanishi, "Bipedal humanoid robot that makes humans laugh with use of the method of comedy and affects their psychological state actively," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1965–1970, May 2014.

[8] D. Li, P. L. P. Rau, and Y. Li, "A cross-cultural study: Effect of robot appearance and task," *International Journal of Social Robotics*, vol. 2, no. 2, pp. 175–186, 2010.

[9] M. Destephe, M. Brandao, T. Kishi, M. Zecca, K. Hashimoto, and A. Takanishi, "Walking in the uncanny valley: importance of the attractiveness on the acceptance of a robot as a working partner," *Frontiers in Psychology*, vol. 6, February 2015.

[10] M. Mori, K. F. MacDorman, and N. Kageki, "The uncanny valley [from the field]," *IEEE Robotics Automation Magazine*, vol. 19, pp. 98–100, June 2012.

[11] W. Huang, J. Kim, and C. Atkeson, "Energy-based optimal step planning for humanoids," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3124–3129, May 2013.

[12] J. Kim, N. Pollard, and C. Atkeson, "Quadratic encoding of optimized humanoid walking," in *13th IEEE-RAS International Conference on Humanoid Robots*, pp. 300–306, Oct 2013.

[13] A. Hornung, A. Dornbush, M. Likhachev, and M. Bennewitz, "Anytime search-based footstep planning with suboptimality bounds," in *12th IEEE-RAS International Conference on Humanoid Robots*, pp. 674–679, Nov 2012.

[14] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *14th IEEE-RAS International Conference on Humanoid Robots*, pp. 279–286, Nov 2014.

[15] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[16] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 473–491, 2016.

[17] S. Feng, X. Xinjilefu, W. Huang, and C. Atkeson, "3d walking based on online optimization," in *13th IEEE-RAS International Conference on Humanoid Robots*, October 2013.

[18] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[19] K. M. Wurm *et al.*, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation*, vol. 2, 2010.

[20] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Slip prediction using visual information," in *Proceedings of Robotics: Science and Systems*, (Philadelphia, USA), August 2006.

[21] J. Baron, *Thinking and deciding*. Cambridge University Press, 2000.

[22] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[23] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[24] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *2005 IEEE International Conference on Robotics and Automation*, pp. 629–634, April 2005.

[25] J. Garimort and A. Hornung, "Humanoid navigation with dynamic footstep plans," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3982–3987, May 2011.

[26] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1325–1349, 2008.

[27] C. Eldershaw and M. Yim, "Motion planning of legged vehicles in an unstructured environment," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, pp. 3383–3389 vol.4, 2001.

[28] K. Hauser, T. Bretl, K. Harada, and J.-C. Latombe, "Using motion primitives in probabilistic sample-based planning for humanoid robots," in *Algorithmic foundation of robotics VII*, pp. 507–522, Springer, 2008.

[29] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, pp. 43:1–43:8, July 2012.

[30] M. Posa and R. Tedrake, *Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact*, pp. 527–542. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[31] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[32] A. J. Davison and D. W. Murray, "Mobile robot localisation using active vision," in *Proceedings of the 5th European Conference on Computer Vision - Volume II*, ECCV '98, (London, UK), pp. 809–825, Springer-Verlag, 1998.

[33] G. Lidoris, K. Kuhnlenz, D. Wollherr, and M. Buss, "Information-based gaze direction planning algorithm for slam," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pp. 302–307, 2006.

[34] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *2005 IEEE International Conference on Robotics and Automation*, pp. 661–666, 2005.

[35] M. Strand and R. Dillmann, "Using an attributed 2d-grid for next-best-view planning on 3d environment data for an autonomous robot," in *International Conference on Information and Automation*, pp. 314–319, 2008.

[36] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 146–151, 1997.

[37] R. Shade and P. Newman, "Choosing where to go: Complete 3D exploration with stereo," *2011 IEEE International Conference on Robotics and Automation*, pp. 2806–2811, May 2011.

[38] J. Seara and G. Schmidt, "Intelligent gaze control for vision-guided humanoid walking: methodological aspects," *Robotics and Autonomous Systems*, vol. 48, no. 4, pp. 231 – 248, 2004.

[39] M. Suzuki, T. Gritti, and D. Floreano, "Active vision for goal-oriented humanoid robot walking," in *Creating Brain-Like Intelligence* (B. Sendhoff, E. KÃ¶rner, O. Sporns, H. Ritter, and K. Doya, eds.), vol. 5436 of *Lecture Notes in Computer Science*, pp. 303–313, Springer Berlin Heidelberg, 2009.

[40] M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Active gaze strategy for reducing map uncertainty along a path," in *3rd IFToMM International Symposium on Robotics and Mechatronics*, pp. 455–466, October 2013.

[41] M. Brandao, K. Hashimoto, and A. Takanishi, "Uncertainty-based mapping and planning strategies for safe navigation of robots with stereo vision," in *14th Mechatronics Forum Conference*, pp. 80–85, June 2014.

[42] H. Hirschmüller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1582–99, Sept. 2009.

[43] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.

[44] X. Hu and P. Mordohai, "A Quantitative Evaluation of Confidence Measures for Stereo Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2121–2133, 2012.

[45] G. Egnal, M. Mintz, and R. P. Wildes, "A stereo confidence metric using single view imagery with comparison to five alternative approaches," *Image and vision computing*, vol. 22, no. 12, pp. 943–957, 2004.

[46] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct 2015.

[47] J. Engel, T. Schöps, and D. Cremers, *LSD-SLAM: Large-Scale Direct Monocular SLAM*, pp. 834–849. Cham: Springer International Publishing, 2014.

[48] R. A. Newcombe, S. Lovegrove, and A. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 IEEE International Conference on Computer Vision*, pp. 2320–2327, 2011.

[49] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. McDonald, and R. Tedrake, "Continuous humanoid locomotion over uneven terrain using stereo fusion," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pp. 881–888, Nov 2015.

[50] O. E. Ramos Ponce, M. Garcia, N. Mansard, O. Stasse, J.-B. Hayet, and P. Souères, "Towards reactive vision-guided walking on rough terrain: an inverse-dynamics based approach," *International Journal of Humanoid Robotics*, vol. 11, June 2014.

[51] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1251–1262, 2012.

[52] J. S. Gutmann, M. Fukuchi, and M. Fujita, "A floor and obstacle height map for 3d navigation of a humanoid robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1066–1071, April 2005.

[53] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. Software available at http://octomap.github.com.

[54] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, (New York, NY, USA), pp. 303–312, ACM, 1996.

[55] R. Deits and R. Tedrake, *Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming*, pp. 109–124. Cham: Springer International Publishing, 2015.

[56] K. Mamou and F. Ghorbel, "A simple and efficient approach for 3d mesh approximate convex decomposition," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 3501–3504, Nov 2009.

[57] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[58] J. H. Park and O. Kwon, "Reflex control of biped robot locomotion on a slippery surface," in *2001 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 4134–4139 vol.4, 2001.

[59] T. E. Lockhart, J. C. Woldstad, J. L. Smith, and J. D. Ramsey, "Effects of age related sensory degradation on perception of floor slipperiness and associated slip parameters," *Safety Science*, vol. 40, no. 7â€“8, pp. 689 – 703, 2002.

[60] K. Gieck, *Engineering formulas*. 1986.

[61] J. R. Davis, *Concise metals engineering data book*. Asm International, 1997.

[62] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "Material recognition in the wild with the materials in context database," *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[63] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 3828–3836, 2015.

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[65] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), pp. 487–495, Curran Associates, Inc., 2014.

[66] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[67] M. Campbell, A. Hoane, and F. hsiung Hsu, "Deep blue," *Artificial Intelligence*, vol. 134, no. 1, pp. 57 – 83, 2002.

[68] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[69] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott, "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1847–1871, Aug 2013.

[70] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157 vol.2, 1999.

[71] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pp. 510–517, Ieee, 2012.

[72] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[73] J. A. R. Marshall, R. Bogacz, A. Dornhaus, R. Planqué, T. Kovacs, and N. R. Franks, "On optimal decision-making in brains and social insect colonies," *Journal of The Royal Society Interface*, vol. 6, no. 40, pp. 1065–1074, 2009.

[74] T. D. Seeley, P. K. Visscher, T. Schlegel, P. M. Hogan, N. R. Franks, and J. A. R. Marshall, "Stop signals provide cross inhibition in collective decision-making by honeybee swarms," *Science*, vol. 335, no. 6064, pp. 108–111, 2012.

[75] M. Brandao, L. Jamone, P. Kryczka, N. Endo, K. Hashimoto, and A. Takanishi, "Reaching for the unreachable: integration of locomotion and whole-body movements for extended visually guided reaching," in *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 28–33, October 2013.

[76] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *International Conference on Robotics and Automation*, pp. 3041–3047, IEEE, 2007.

[77] K. Fiehler, I. Schütz, and D. Y. P. Henriques, "Gaze-centered spatial updating of reach targets across different memory delays," *Vision Research*, vol. 51, pp. 890–897, 2011.

[78] M. Flanders, L. Daghestani, and A. Berthoz, "Reaching beyond reach," *Experimental Brain Research*, vol. 126, no. 1, pp. 19–30, 1999.

[79] R. M. Wilkie, J. P. Wann, and R. S. Allison, "Active gaze, visual look-ahead, and locomotor control," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 34, no. 5, pp. 1150–1164, 2008.

[80] P. Prévost, I. Yuri, G. Renato, and B. Alain, "Spatial invariance in anticipatory orienting behaviour during human navigation," *Neuroscience letters*, vol. 339, no. 3, pp. 243–247, 2003.

[81] M. Zarrugh, F. Todd, and H. Ralston, "Optimization of energy expenditure during level walking," *European Journal of Applied Physiology and Occupational Physiology*, vol. 33, no. 4, pp. 293–306, 1974.

[82] A. Minetti and R. Alexander, "A theory of metabolic costs for bipedal gaits," *Journal of Theoretical Biology*, vol. 186, no. 4, pp. 467 – 476, 1997.

[83] A. D. Kuo, "A simple model of bipedal walking predicts the preferred speed–step length relationship," *Journal of biomechanical engineering*, vol. 123, no. 3, pp. 264–269, 2001.

[84] B. R. Umberger and P. E. Martin, "Mechanical power and efficiency of level walking with different stride rates," *Journal of Experimental Biology*, vol. 210, no. 18, pp. 3255–3265, 2007.

[85] Y. Ogura, K. Shimomura, H. Kondo, A. Morishima, T. Okubo, S. Momoki, H. o. Lim, and A. Takanishi, "Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3976–3981, Oct 2006.

[86] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, no. 3, pp. 369–383, 2010.

[87] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *14th IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, Nov 2014.

[88] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," *IEEE Transactions on Robotics*, vol. 28, pp. 427–439, April 2012.

[89] S. Kajita, K. Kaneko, K. Harada, F. Kanehiro, K. Fujiwara, and H. Hirukawa, "Biped walking on a low friction floor," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, pp. 3546–3552 vol.4, Sept 2004.

[90] M. Nikolic, B. Branislav, and M. Rakovic, "Walking on slippery surfaces: Generalized task-prioritization framework approach," in *Advances on Theory and Practice of Robots and Manipulators* (M. Ceccarelli and V. A. Glazunov, eds.), vol. 22 of *Mechanisms and Machine Science*, pp. 189–196, Springer International Publishing, 2014.

[91] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1620–1626 vol.2, Sept 2003.

[92] K. Hang, F. Pokorny, and D. Kragic, "Friction coefficients and grasp synthesis," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3520–3526, Nov 2013.

[93] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428 – 442, 2013.

[94] "Open dynamics engine."

[95] "Bullet physics library."

[96] E. J. Gibson, G. Riccio, M. A. Schmuckler, T. A. Stoffregen, D. Rosenberg, and J. Taormina, "Detection of the traversability of surfaces by crawling and walking infants.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 4, p. 533, 1987.

[97] K. E. Adolph, A. S. Joh, and M. A. Eppler, "Infants' perception of affordances of slopes under high-and low-friction conditions.," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 36, no. 4, p. 797, 2010.

[98] G. Cappellini, Y. P. Ivanenko, N. Dominici, R. E. Poppele, and F. Lacquaniti, "Motor patterns during walking on a slippery walkway," *Journal of Neurophysiology*, vol. 103, no. 2, pp. 746–760, 2010.

[99] R. Cham and M. S. Redfern, "Changes in gait when anticipating slippery floors," *Gait & Posture*, vol. 15, no. 2, pp. 159 – 171, 2002.

[100] T. L. Heiden, D. J. Sanderson, J. T. Inglis, and G. P. Siegmund, "Adaptations to normal human gait on potentially slippery surfaces: The effects of awareness and prior slip experience," *Gait & Posture*, vol. 24, no. 2, pp. 237 – 246, 2006.

[101] J. G. Buckley, K. J. Heasley, P. Twigg, and D. B. Elliott, "The effects of blurred vision on the mechanics of landing during stepping down by the elderly," *Gait & Posture*, vol. 21, no. 1, pp. 65 – 71, 2005.

[102] H. B. Menz, S. R. Lord, R. S. George, and R. C. Fitzpatrick, "Walking stability and sensorimotor function in older people with diabetic peripheral neuropathy1," *Archives of Physical Medicine and Rehabilitation*, vol. 85, no. 2, pp. 245 – 252, 2004.

[103] J. C. Menant, J. R. Steele, H. B. Menz, B. J. Munro, and S. R. Lord, "Effects of walking surfaces and footwear on temporo-spatial gait parameters in young and older people," *Gait & Posture*, vol. 29, no. 3, pp. 392 – 397, 2009.

[104] A. J. Chambers, S. Margerum, M. S. Redfern, and R. Cham, "Kinematics of the foot during slips," *Occupational Ergonomics*, vol. 3, no. 4, pp. 225 – 234, 2002.

[105] M. G. A. Llewellyn and V. R. Nevola, "Strategies for walking on low-friction surfaces," in *The Fifth International Conference on Environmental Ergonomics. Maastricht*, pp. 156 – 157, 1992.

[106] R. M. Alexander, *Principles of animal locomotion*. Princeton University Press, 2003.

[107] A. Minetti, "Optimum gradient of mountain paths," *Journal of Applied Physiology*, vol. 79, no. 5, pp. 1698–1703, 1995.

[108] D. R. Proffitt, M. Bhalla, R. Gossweiler, and J. Midgett, "Perceiving geographical slant," *Psychonomic Bulletin & Review*, vol. 2, no. 4, pp. 409–428, 1995.

[109] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Optimizing walking controllers for uncertain inputs and environments," *ACM Trans. Graph.*, vol. 29, pp. 73:1–73:8, July 2010.

[110] D. A. Winter, *Biomechanics and motor control of human gait: normal, elderly and pathological*. 2 ed., 1991.

[111] K. Sasaki, R. R. Neptune, and S. A. Kautz, "The relationships between muscle, external, internal and joint mechanical work during normal walking," *Journal of Experimental Biology*, vol. 212, no. 5, pp. 738–744, 2009.

[112] D. A. Winter, "A new definition of mechanical work done in human movement," *Journal of Applied Physiology*, vol. 46, no. 1, pp. 79–83, 1979.

[113] J. S. Matthis and B. R. Fajen, "Humans exploit the biomechanics of bipedal gait during visually guided walking over complex terrain," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 280, no. 1762, p. 2013, 2013.

[114] H. B. Menz, S. R. Lord, and R. C. Fitzpatrick, "Acceleration patterns of the head and pelvis when walking on level and irregular surfaces," *Gait & Posture*, vol. 18, no. 1, pp. 35 – 46, 2003.

[115] D. Fong, Y. Hong, and J. X. Li, "Lower-extremity gait kinematics on slippery surfaces in construction worksites.," *Medicine and science in sports and exercise*, vol. 37, no. 3, pp. 447–454, 2005.

[116] S. Collins and A. Ruina, "A bipedal walking robot with efficient and human-like gait," in *2005 IEEE International Conference on Robotics and Automation*, pp. 1983–1988, April 2005.

[117] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. Lim, and A. Takanishi, "Development of a new humanoid robot wabian-2," in *2006 IEEE/RSJ International Conference on Robotics and Automation*, IEEE-RAS, 2006.

[118] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: A versatile 3d robot simulator," in *Simulation, Modeling, and Programming for Autonomous Robots*, vol. 6472 of *Lecture Notes in Computer Science*, pp. 51–62, Springer, 2010.

[119] K. Hashimoto, H. Kondo, H.-O. Lim, and A. Takanishi, *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, ch. Online Walking Pattern Generation Using FFT for Humanoid Robots, pp. 417–438. Springer International Publishing, 2015.

[120] B. Damas and J. Santos-Victor, "Online learning of single-and multivalued functions with an infinite mixture of linear experts," *Neural computation*, vol. 25, no. 11, pp. 3044–3091, 2013.

[121] J. Perry, *Gait analysis: normal and pathological function*. Slack Incorporated, 1992.

[122] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," in *Advances in Neural Information Processing Systems*, pp. 767–774, 2003.

[123] E.-C. Corporation, *DC Motors, Speed Controls, Servo Systems*. Pergamon, 1977.

[124] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, (Shanghai, China), May 9-13 2011.

[125] M. Likhachev, "Search-based planning library," 2010.

[126] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[127] D. Kraft, "Algorithm 733: Tomp–fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software*, vol. 20, pp. 262–281, Sep 1994.

[128] S. G. Johnson, "The nlopt nonlinear-optimization package," 2014.

[129] M. Tremblay and M. Cutkosky, "Estimating friction using incipient slip sensing during a manipulation task," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 429–434 vol.1, May 1993.

[130] H. Liu, X. Song, T. Nanayakkara, K. Althoefer, and L. Seneviratne, "Friction estimation based object surface classification for intelligent manipulation," in *IEEE ICRA 2011 workshop on autonomous grasping, Shanghai*, 2011.

[131] K. Kaneko, F. Kanehiro, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, and H. Hirukawa, "Slip observer for walking on a low friction floor," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 634–640, Aug 2005.

[132] N. Okita and H. Sommer, "A novel foot slip detection algorithm using unscented kalman filter innovation," in *American Control Conference (ACC), 2012*, pp. 5163–5168, June 2012.

[133] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 6058–6064, Nov 2013.

[134] Y.-W. Chao, Z. Wang, R. Mihalcea, and J. Deng, "Mining semantic affordances of visual object categories," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 4259–4267, June 2015.

[135] M. F. Lesch, W.-R. Chang, and C.-C. Chang, "Visually based perceptions of slipperiness: Underlying cues, consistency and relationship to coefficient of friction," *Ergonomics*, vol. 51, no. 12, pp. 1973–1983, 2008. PMID: 19034787.

[136] A. S. Joh, K. E. Adolph, M. R. Campbell, and M. A. Eppler, "Why walkers slip: Shine is not a reliable cue for slippery ground," *Perception & Psychophysics*, vol. 68, no. 3, pp. 339–352, 2006.

[137] K. W. Li, W.-R. Chang, T. B. Leamon, and C. J. Chen, "Floor slipperiness measurement: friction coefficient, roughness of floors, and subjective perception under spillage conditions," *Safety Science*, vol. 42, no. 6, pp. 547 – 565, 2004.

[138] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces: A richly annotated catalog of surface appearance," *ACM Trans. on Graphics (SIGGRAPH)*, vol. 32, no. 4, 2013.

[139] LimeSurvey Project Team / Carsten Schmitz, *LimeSurvey: An Open Source survey tool*. LimeSurvey Project, Hamburg, Germany, 2012.

[140] R. W. Fleming, C. Wiebel, and K. Gegenfurtner, "Perceptual qualities and material classes," *Journal of Vision*, vol. 13, no. 8, p. 9, 2013.

[141] E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Am.*, vol. 61, pp. 1–11, Jan 1971.

[142] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Trans. on Graphics (SIGGRAPH)*, vol. 33, no. 4, 2014.

[143] N. Limare, A. B. Petro, C. Sbert, and J.-M. Morel, "Retinex Poisson Equation: a Model for Color Perception," *Image Processing On Line*, vol. 1, 2011.

[144] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychological review*, vol. 104, no. 2, p. 211, 1997.

[145] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26* (C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, eds.), pp. 3111–3119, Curran Associates, Inc., 2013.

[146] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[147] "Princeton University "About WordNet." WordNet. Princeton University," 2010.

[148] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 0–0, 2015.

[149] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[150] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[151] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[152] J. Gragg, E. Klose, and J. Yang, "Modelling the stochastic nature of the available coefficient of friction at footwear-floor interfaces," *Ergonomics*, vol. 0, no. 0, pp. 1–8, 2016. PMID: 27592564.

[153] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 1–14, 2003.

[154] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *International Journal of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998.

[155] C. J. Pal, J. J. Weinman, L. C. Tran, and D. Scharstein, "On Learning Conditional Random Fields for Stereo," *International Journal of Computer Vision*, vol. 99, pp. 319–337, Oct. 2010.

[156] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistér, and M. Pollefeys, "Real-time visibility-based fusion of depth maps," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[157] M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "On the formulation, performance and design choices of cost-curve occupancy grids for stereo-vision based 3d reconstruction," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1818–1823, September 2014.

[158] J. Cech and R. Sara, "Efficient sampling of disparity space for fast and accurate matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[159] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 236, pp. 209–236, 1989.

[160] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[161] a. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, no. 2, pp. 858–863, 1997.

[162] L. Matthies and M. Okutomi, "A Bayesian foundation for active stereo vision," *Proc. SPIE Sensor Fusion II: Human and Machine Strategies*, pp. 1–13, 1989.

[163] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.

[164] P. Mordohai, "The self-aware matching measure for stereo," in *IEEE International Conference on Computer Vision*, pp. 1841–1848, IEEE, 2009.

[165] R. Mayoral, G. Lera, and M. J. Perez-Ilzarbe, "Evaluation of correspondence errors for stereo," *Image and Vision Computing*, vol. 24, no. 12, pp. 1288 – 1300, 2006.

[166] A. Torabi, M. Najafianrazavi, and G. A. Bilodeau, "A comparative evaluation of multimodal dense stereo correspondence measures," in *2011 IEEE International Symposium on Robotic and Sensors Environments*, pp. 143–148, 2011.

[167] M. Gong and Y.-H. Yang, "Fast unambiguous stereo matching using reliability-based dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 998–1003, 2005.

[168] C. Dima and S. Lacroix, "Using multiple disparity hypotheses for improved indoor stereo," in *IEEE International Conference on Robotics and Automation*, pp. 3347–3353, IEEE, 2002.

[169] N. Sabater, A. Almansa, and J. M. Morel, "Meaningful matches in stereovision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 930–942, May 2012.

[170] R. Sára, "Finding the largest unambiguous component of stereo matching," in *Proceedings of the 7th European Conference on Computer Vision-Part III*, ECCV '02, (London, UK, UK), pp. 900–914, Springer-Verlag, 2002.

[171] D. Pfeiffer, S. Gehrig, and N. Schneider, "Exploiting the power of stereo confidences," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 297–304, June 2013.

[172] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[173] K. Konolige, "Improved occupancy grids for map building," *Autonomous Robots*, 1997.

[174] L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid-based representation," *1988 IEEE International Conference on Robotics and Automation*, pp. 727–733, 1988.

[175] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation," *Autonomous Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[176] F. Andert, "Drawing stereo disparity images into occupancy grids: measurement model and fast implementation," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5191–5197, 2009.

[177] M. Perrollaz, A. Spalanzani, and D. Aubert, "Probabilistic representation of the uncertainty of stereo-vision and application to obstacle detection," *2010 IEEE Intelligent Vehicles Symposium*, pp. 313–318, June 2010.

[178] A. Suppes, F. Suhling, and M. Hötter, "Robust obstacle detection from stereoscopic image sequences using kalman filtering," *Pattern Recognition*, pp. 385–391, 2001.

[179] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, 2000.

[180] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision*, 2010.

[181] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.

[182] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. I–195–I–202, 2003.

[183] D. Scharstein and C. Pal, "Learning Conditional Random Fields for Stereo," *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, June 2007.

[184] H. Hirschmuller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, ed.), pp. 1–8, 2007.

[185] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

[186] R. Haeusler and D. Kondermann, "Synthesizing real world stereo challenges," in *Pattern Recognition* (J. Weickert, M. Hein, and B. Schiele, eds.), vol. 8142 of *Lecture Notes in Computer Science*, pp. 164–173, Springer Berlin Heidelberg, 2013.

[187] W. van Ackooij, A. Möller, R. Henrion, and R. Zorgati, *Chance constrained programming and its applications to energy management*. INTECH Open Access Publisher, 2011.

[188] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science and Engg.*, vol. 13, pp. 22–30, Mar. 2011.

[189] D. Holz and S. Behnke, *Fast Range Image Segmentation and Smoothing Using Approximate Surface Reconstruction and Region Growing*, pp. 61–73. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[190] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[191] M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Footstep planning for slippery and slanted terrain using human-inspired models," *IEEE Transactions on Robotics*, vol. 32, pp. 868–879, Aug 2016.

[192] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.

[193] A. Orthey, V. Ivan, M. Naveau, Y. Yang, O. Stasse, and S. Vijayakumar, "Homotopic particle motion planning for humanoid robotics." working paper or preprint, Mar. 2015.

[194] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Motion planning with dynamics by a synergistic combination of layers of planning," *IEEE Transactions on Robotics*, vol. 26, pp. 469–482, June 2010.

[195] A. Ruina and R. Pratap, *Introduction to Statics and Dynamics*. Preprint for Oxford University Press, 2008.

[196] A. M. Williams, K. Davids, and J. G. P. Williams, *Visual perception and action in sport*. Taylor & Francis, 1999.

[197] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co., Inc., 1982.

[198] J. J. Gibson, *The ecological approach to visual perception*. Houghton, Mifflin and Company, 1979.

[199] K. Hashimoto, H. j. Kang, M. Nakamura, E. Falotico, H. o. Lim, A. Takanishi, C. Laschi, P. Dario, and A. Berthoz, "Realization of biped walking on soft ground with stabilization control based on gait analysis," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2064–2069, Oct 2012.

[200] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 4372–4378, May 2010.

[201] M. Brandao, R. Ferreira, K. Hashimoto, A. Takanishi, and J. Santos-Victor, "On stereo confidence measures for global methods: Evaluation, new model and integration into occupancy grids," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 116–128, Jan 2016.

[202] L. Jamone, M. Brandao, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 556–567, 2014.

[203] M. Brandao, Y. M. Shiguematsu, K. Hashimoto, and A. Takanishi, "Material recognition cnns and hierarchical planning for biped robot locomotion on slippery terrain," in *2016 IEEE-RAS International Conference on Humanoid Robots*, Nov 2016.

[204] M. Brandao, K. Hashimoto, and A. Takanishi, "Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation," in *2016 IEEE-RAS International Conference on Humanoid Robots*, Nov 2016.

[205] M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Optimizing energy consumption and preventing slips at the footstep planning level," in *15th IEEE-RAS International Conference on Humanoid Robots*, pp. 1–7, Nov 2015.

[206] M. Brandao, K. Hashimoto, and A. Takanishi, "Extending humanoid footstep planning with zmp tracking error constraints," in *Proceedings of the 6th International Conference on Advanced Mechatronics*, p. 130, Dec 2015.

[207] M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Gait planning for biped locomotion on slippery terrain," in *14th IEEE-RAS International Conference on Humanoid Robots*, pp. 303–308, November 2014.

[208] M. Destephe, M. Brandao, T. Kishi, M. Zecca, K. Hashimoto, and A. Takanishi, "Emotional gait: Effects on humans' perception of humanoid robots," in *23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 261–266, August 2014.

[209] M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Integrating the whole cost-curve of stereo into occupancy grids," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4681–4686, November 2013.

Appendix A

Friction from vision questionnaires

This appendix shows the questionnaires used to obtain the datasets and results described in Chapter 3.

A.1 OSA+F dataset

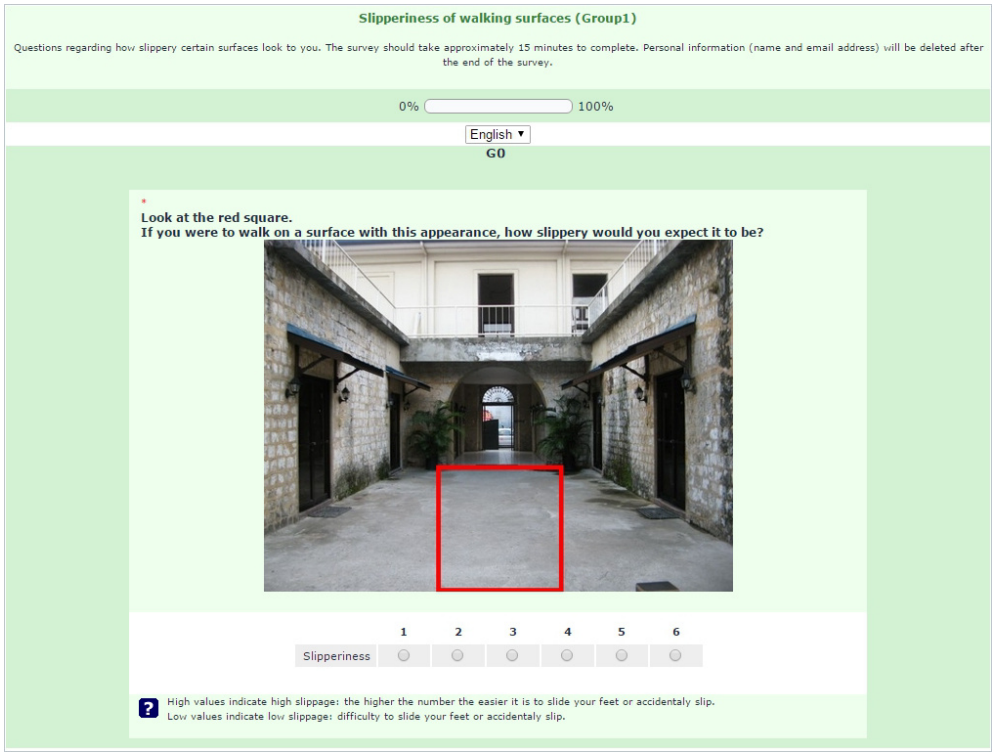


Fig. A.1 Question from the OSA+F survey

A.2 GTF dataset

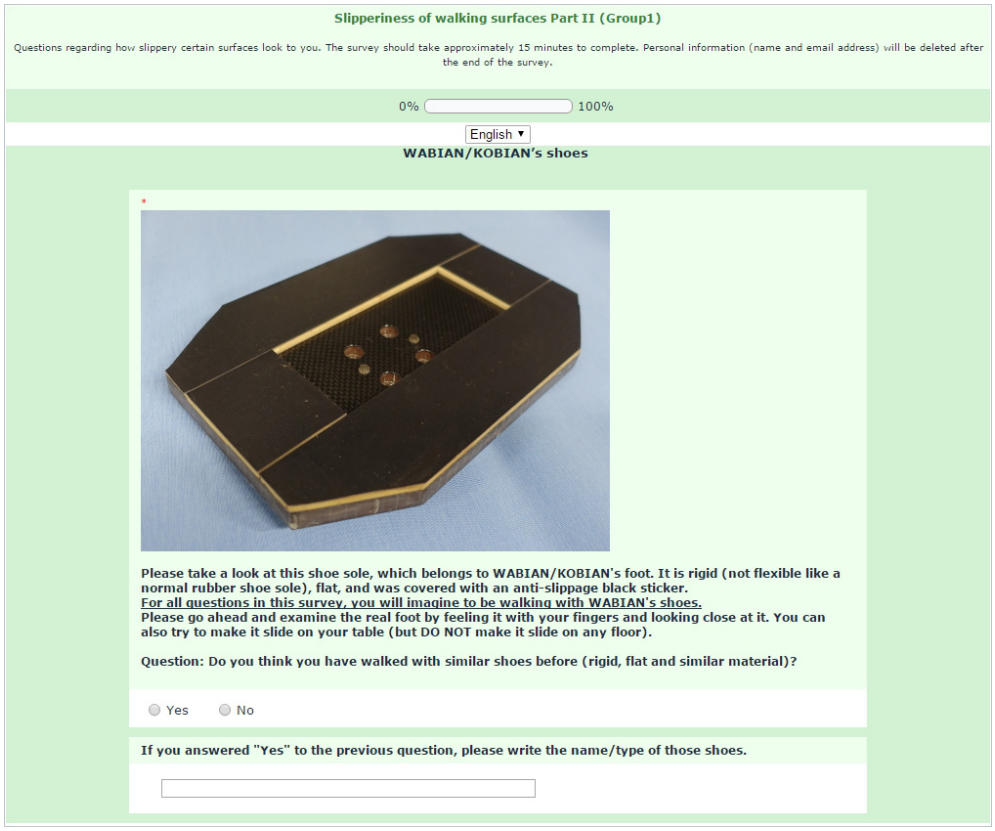


Fig. A.2 Question from the GTF survey

A.3 Material friction

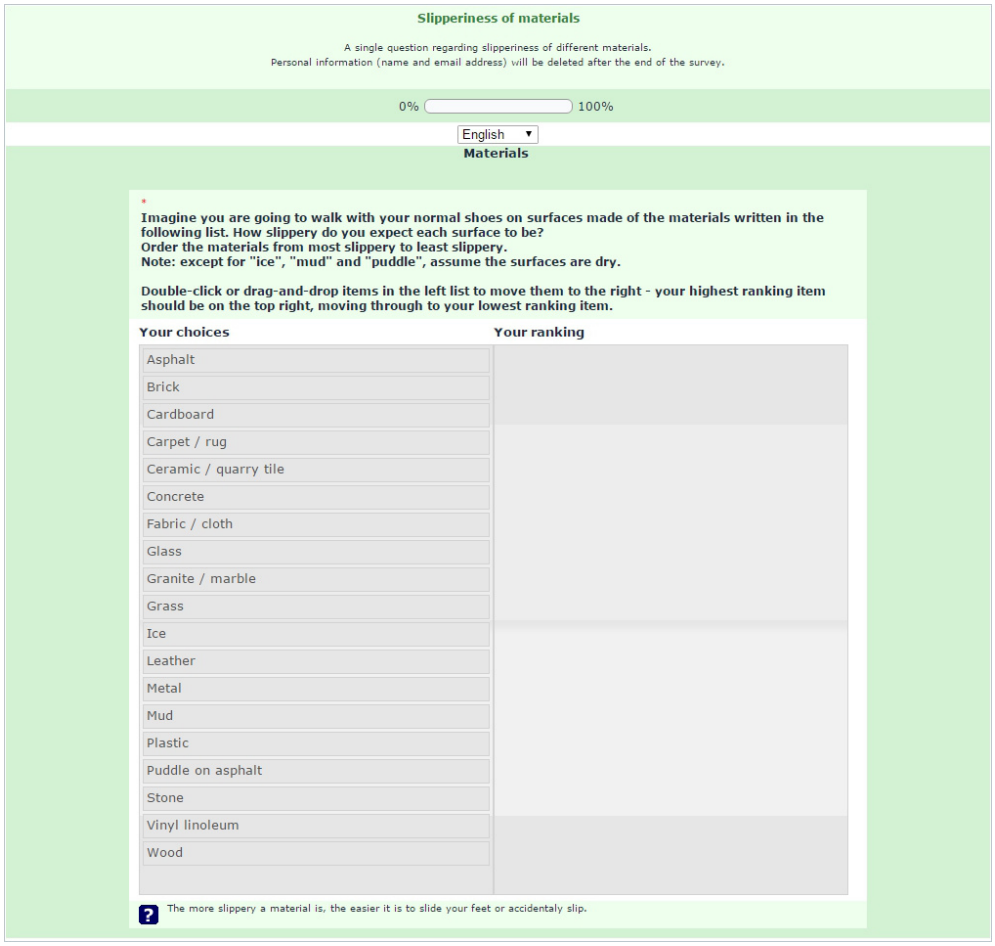


Fig. A.3 Material friction survey

Research achievements

Publications

Journal articles

- M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Footstep planning for slippery and slanted terrain using human-inspired models," *IEEE Transactions on Robotics*, vol. 32, pp. 868–879, Aug 2016

- M. Brandao, R. Ferreira, K. Hashimoto, A. Takanishi, and J. Santos-Victor, "On stereo confidence measures for global methods: Evaluation, new model and integration into occupancy grids," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, pp. 116–128, Jan 2016

- M. Destephe, M. Brandao, T. Kishi, M. Zecca, K. Hashimoto, and A. Takanishi, "Walking in the uncanny valley: importance of the attractiveness on the acceptance of a robot as a working partner," *Frontiers in Psychology*, vol. 6, February 2015

- L. Jamone, M. Brandao, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 556–567, 2014

International conference proceedings

- M. Brandao, Y. M. Shiguematsu, K. Hashimoto, and A. Takanishi, "Material recognition cnns and hierarchical planning for biped robot

locomotion on slippery terrain," in *2016 IEEE-RAS International Conference on Humanoid Robots*, Nov 2016

- M. Brandao, K. Hashimoto, and A. Takanishi, "Friction from vision: A study of algorithmic and human performance with consequences for robot perception and teleoperation," in *2016 IEEE-RAS International Conference on Humanoid Robots*, Nov 2016

- M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Optimizing energy consumption and preventing slips at the footstep planning level," in *15th IEEE-RAS International Conference on Humanoid Robots*, pp. 1–7, Nov 2015

- M. Brandao, K. Hashimoto, and A. Takanishi, "Extending humanoid footstep planning with zmp tracking error constraints," in *Proceedings of the 6th International Conference on Advanced Mechatronics*, p. 130, Dec 2015

- M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Gait planning for biped locomotion on slippery terrain," in *14th IEEE-RAS International Conference on Humanoid Robots*, pp. 303–308, November 2014

- M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "On the formulation, performance and design choices of cost-curve occupancy grids for stereo-vision based 3d reconstruction," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1818–1823, September 2014

- M. Brandao, K. Hashimoto, and A. Takanishi, "Uncertainty-based mapping and planning strategies for safe navigation of robots with stereo vision," in *14th Mechatronics Forum Conference*, pp. 80–85, June 2014

- M. Destephe, M. Brandao, T. Kishi, M. Zecca, K. Hashimoto, and A. Takanishi, "Emotional gait: Effects on humans' perception of humanoid robots," in *23rd IEEE International Symposium on Robot and Human Interactive Communication*, pp. 261–266, August 2014

- M. Brandao, L. Jamone, P. Kryczka, N. Endo, K. Hashimoto, and A. Takanishi, "Reaching for the unreachable: integration of locomotion

and whole-body movements for extended visually guided reaching," in *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 28–33, October 2013

- M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Integrating the whole cost-curve of stereo into occupancy grids," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4681–4686, November 2013

- M. Brandao, R. Ferreira, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Active gaze strategy for reducing map uncertainty along a path," in *3rd IFToMM International Symposium on Robotics and Mechatronics*, pp. 455–466, October 2013

Grants and awards

- 2016: Best Paper Award Finalist (IEEE Humanoids 2016)
- 2015-2016: JSPS KAKENHI Grant Number 15J06497
- 2013-2015: JSPS Strategic Young Researcher Overseas Visits Program for Accelerating Brain Circulation
- 2013: Isao Okawa Scholarship for Information Technology Science
- 2013: Highly Commended Paper Award (IFToMM ISRM 2013)