UNIVERSITY OF CALIFORNIA
Los Angeles

Perception, Information Acquisition, and Prediction in Visual Tasks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Vasiliy Karasev

2015

ABSTRACT OF THE DISSERTATION

Perception, Information Acquisition, and Prediction in Visual Tasks

by

Vasiliy Karasev

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2015

Professor Stefano Soatto, Chair

This thesis describes a research program towards enabling mobile visual agents to successfully solve visual decision problems. We identify perception, information acquisition, and prediction as the three areas to focus on, and develop methods to facilitate these stages, which are then applied to a sample selection of visual tasks. We first focus on segmentation and categorization of objects in video, and propose techniques for doing so. We then describe approaches for information-acquisition, useful whenever the data collection process can be controlled, or when one can choose which data to process. Finally, we describe an approach that leverages video to predict the long term behavior of specific objects of interest.

The dissertation of Vasiliy Karasev is approved.

Ying Nian Wu

Lieven Vandenberghe

Stefano Soatto, Committee Chair

University of California, Los Angeles

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express gratitude to all who made my research at UCLA possible and enjoyable. First, my advisor Professor Stefano Soatto, for agreeing to take me on as a student when I knew nothing about vision, and providing encouragement and support throughout my life as a PhD student. Stefano's principled approach to vision, the ability to approach problems with mathematical rigor, and seeing through the details to the underlying issue have been a continuing source of inspiration. His familiarity with an armada of topics in mathematics and statistics was a source of motivation for taking on multiple courses. Finally, his guidance style of making concrete suggestions, while encouraging me to explore and seek out topics that interested me has allowed me to learn about numerous facets of vision.

I am also grateful to my thesis committee members, Professors Lieven Vandenberghe, Suhas Diggavi, and Ying Nian Wu, for their suggestions and support. Lieven Vandenberghe's courses on convex optimization were invaluable to my research, and his eagerness to help everyone who walked into his office was also inspiring. Ying Nian Wu's lectures on statistics motivated me to pursue the majority of the courses in that department and helped to improve my understanding of those topics. Suhas Diggavi offered interesting research directions and provided a very welcome information theory perspective.

I thank all the members of UCLA Vision Lab for making my graduate school experience pleasant and enjoyable. Kamil Wnuk encouraged me to begin thinking about robotic exploration, at a time when the future looked turbulent. Alper Ayvaci provided advice, suggested projects (Ch. 2, itself an extension of his prior work [AS12], owes a large part of its existence to our initial attempts of solving the problem with the Split-Bregman method), and years later mentored me during my internship at Honda Research (which grew into Ch. 5). Avinash Ravichandran offered strong insights on a wide variety of academic and non-academic topics. One of his many ideas grew into Ch. 3. While we worked on it he provided much needed guidance and exposed me to a sea of recognition tools. I thank Brian Taylor for all the conversations and the thrilling collaborations (among them, Ch. 2). His uncompromising endurance during deadlines and the ability to withstand various challenges remains a source of inspiration. Virginia Estellers was always quick to

offer advice on optimization – she originally pointed me towards the primal-dual method used in Ch. 2 and offered many insights on its performance. Georgios Georgiadis was helpful with providing feedback on my work, and we shared discussions on a wide range of topics. Josh Hernandez provided infrequent and laconic, but extremely insightful and valuable comments. Konstantine enthusiastically offered experience on filtering and estimation. During my first year, I had the opportunity to connect with Alessandro Chiuso – his thoughts on robotic exploration and control with him were always a source of fresh ideas and stimulation, ultimately leading to Ch. 4.

I thank my family for their unconditional love and encouragement over the years. Finally, I thank Jia – it is difficult to imagine these several years without her support.

VITA

2009	B.S. Electrical Engineering and Computer Science, University of California, Berkeley.
2010	Teaching Assistant, University of California, Los Angeles.
2011	M.S. Electrical Engineering, University of California, Los Angeles.
2013	Research intern, Samsung Display.
2015	Research intern, Honda Research Institute.

PUBLICATIONS

1. B. Taylor, V. Karasev and S. Soatto "Causal Video Object Segmentation from Persistence of Occlusions", In *Proc. Computer Vision and Pattern Recognition Conference (CVPR)*, 2015.

2. V. Karasev, A. Ravichandran and S. Soatto "Active Frame, Location, and Detector Selection for Automated and Manual Video Annotation", In *Proc. Computer Vision and Pattern Recognition Conference (CVPR)*, 2014.

3. V. Karasev, A. Chiuso and S. Soatto "Control Recognition Bounds for Visual Learning and Exploration", In *Information Theory and Applications Workshop (ITA)*, 2013.

4. V. Karasev, A. Chiuso and S. Soatto "Controlled Recognition Bounds for Visual Learning and Exploration", In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

CHAPTER 1

Introduction

Consider an autonomous mobile robot equipped with a camera (an agent) autonomously navigating in a physical environment (the scene) and successfully performing common visual decision problems, such as localization, recognition, and categorization of objects. The system interacts with the environment in a closed loop: we may say that its vision-guided behavior forms a perception-action cycle. At the "perception" stage, visual input (the stream of images from the camera sensor) is organized and interpreted to form a task-dependent internal representation of the environment. Subsequently, during the "action" stage, this internal representation is used to hypothesize future changes in the environment, to plan a strategy for solving a task (the visual decision problem) at hand, and to select the next action. Upon executing the action (literally moving to a new vantage point), the system obtains new visual input, which is then interpreted and used to select the next action; the cycle is repeated.

This thesis is concerned with development of algorithms toward making such computer vision systems possible and practical. We identify perception, acquisition of information, and prediction as the three main areas of focus. Perception involves the design of representations or features that enable completion of a particular task, and has traditionally been the mainstay of computer vision. Indeed, some of the most common algorithms can be viewed as constructing a representation tailored to a particular task. In our work, we focus on developing representations that are useful for accurate object localization and categorization tasks. Information acquisition encompasses control and planning, and involves selecting a sequence of actions that minimizes uncertainty about the parameter of interest (object location, its category, etc), while accounting for uncertainty and ambiguity of visual data, the latter stemming, for example, from occlusions and finite sensor resolution. Prediction ability becomes crucial if the environment is dynamic, or if it is populated by

other intelligent entities. In these cases, if the system is expected to behave safely, it must possess the ability to predict others' behavior, and use these predictions to adjust its own behavior. In the following section we outline our specific contributions with regard to these areas.

1.1 Thesis outline

Chapter 2: Causal Video Object Segmentation from Persistence of Occlusions

Finding objects in video is important for numerous tasks, e.g. action recognition, subsequent object categorization, video summarization, enhancement, and content-based retrieval. We are interested in accurately localizing objects beyond what is afforded by bounding boxes. This prompts us to consider pixelwise masks which annotate points that backproject onto objects in the scene; this is commonly called "video object segmentation".

Commonly, discovery of objects in video is performed using a combination of appearance and motion cues[1]. A number of methods exist that provide "foreground" / "background" partition, but lack the ability to separate individual objects in the "foreground". In addition, most methods that do not require manual initialization are "batch" in nature, requiring a whole video for processing – this permits the use of long-term motion information to reliably separate objects from background.

In Chapter 2 we propose an alternative approach that processes a video stream causally and provides pixelwise segmentations for any number of objects, while providing a coarse depth ordering as a byproduct. The method crucially relies on occlusions (and disocclusions) – parts of the scene disappearing from (or appearing into) the view, as a result of the object or camera moving through the scene. These phenomena allow us to construct local pairwise foreground/background relations, which are then bootstrapped into a global depth order partition by solving a convex optimization problem. Object labels are obtained by examining connected components of the discrete depth layers.

[1] Of course, if one has priors on the categories of objects present in the video, one can also use specific object detectors. We consider this situation in Chapter 3. If priors are not available, or if the penalty for a missed detection of an unexpected object is high (as is the case, for example, in autonomous/assisted driving) one needs to use more generic cues, namely a combination of appearance and motion.

Chapter 3: Active Frame, Location, and Detector Selection for Automated and Manual Video Annotation

Often, merely knowing that an object is present at a given location is not sufficient – we need the ability to reason about its semantic category (e.g. whether it is a cat or a tiger). This need arises in scene understanding, autonomous driving, surveillance, content-based retrieval, and various generic recognition applications. As above, we are interested in localizing objects accurately with pixelwise masks, commonly referred to as "semantic video segmentation".

Accurate categorization relies on object detectors which require significant computation resources; running a battery of detectors on every frame of the video is possible but is usually prohibitively expensive. Yet, as the video contains a lot of redundancy, in Chapter 3 we describe an approach that exploits it, reducing the computational burden, without degrading performance. While our approach is intended for "automated" processing (using object detectors), as a special case it can be used in the "manual" scenario (where object detectors are replaced with human annotators).

We cast the problem within the sequential decision making framework, where the objective is to minimize uncertainty in the underlying object identities in the video, and the choice at each stage is which region within a video frame to submit to a battery of object detectors. The selection is guided by maximization of information (appropriately defined) between the detector responses and the underlying category labels. We show that this problem is submodular, which allows us to use an online, greedy strategy with a known performance guarantee. Further, we extend the framework to also allow for selection of a subset of detectors, based on learned context. The method is able to process the video at a fraction of the original cost, and as we empirically show achieves the same performance as when the video is processed at full cost.

Chapter 4: Active Control for Visual Search and Control-Recognition Tradeoffs

Many common vision tasks are ill-posed when the data available to reach a decision consists of a single image. Even with multiple images, the task may remain ill-posed if the images are gathered passively. However, if the system can exercise control – that is, if it can influence which future data

is acquired – the tasks become well-posed. Thus, the ability to produce correct answers benefits from an active control mechanism.

Consider a detection tasks "is an object X in the scene?". From a single image, this cannot be answered reliably; for example because we are unable to distinguish a three-dimensional scene and a painting of the same scene. Outside this straw-man scenario the problem still remains challenging. This is primarily due to occlusions and finite sensor resolution ("quantization-scale") phenomena. These phenomena produce sensing ambiguities that cannot be resolved from a single image. While object detectors are robust to small partial occlusions, severe or full occlusions are inherently ambiguous. A similar ambiguity arises if the object of interest (that is, X) projects onto area smaller than a pixel; in this case, even the best detectors are powerless.

To answer reliably in presence of these ambiguities, quantization-scale and occlusion nuisances need to be "inverted'. This can be done only by physically changing the vantage point of the camera – approaching putative targets to resolve scaling ambiguity, and looking behind obstacles to resolve ambiguities due to occlusions. This requires the ability to control of the sensing process.

We will show that an optimal control strategy minimizes uncertainty in the variable of interest, or equivalently maximizes information in the acquired images (to be appropriately defined). Performance in the task depends not only on the control strategy but also on the agent's inherent ability to influence the sensing process – its control authority. In the static environments, this quantity is related to volume and geometry of the agent's reachable space.

In Chapter 4 we focus on the problem of "visual search" of an unknown object in an otherwise known and static scene (otherwise known as "intruder detection"). We propose the simplest model that captures the phenomenology of image formation, including scaling and occlusions. Complexity of computing an optimal control strategy in this setting is exponential in planning horizon, and even in the simple scenarios turns out to be intractable. This motivates us to develop efficient heuristics, and verify their performance. Finally, we empirically characterize the relationship between the amount of control one has and performance in the search task.

Chapter 5: Intent-Aware Long-Term Prediction of Pedestrian Motion

To safely operate in dynamic scenes populated by other intelligent agents, a visual system needs to avoid collisions: avoid regions of space likely to be concurrently occupied by another agent (or a human). This requires an ability to predict other agents' behavior. To avoid a game-theoretic conundrum, we assume that other agents are not influenced by our own behavior. The evolution of the behavior can be described by a dynamical system. Being interested in long-term prediction, we are precluded from using linear models based on derivatives of position (i.e. n^{th} order random walk). These models are not sufficiently rich, and as we show, are unsuitable for all but the shortest horizons. Instead, in Chapter 5, we construct a richer alternative that allows us to leverage global environment information and account for environment dynamics.

Our proposed method posits a finite set of behaviors, each of which is characterized by a solution to an optimal control problem. High level knowledge about the environment and its dynamics can be naturally incorporated into these problems, which are formulated and solved to yield policies at training time. At test time, observations of the agents' behavior allow us recognize their decision-making process – to infer which policy is being executed. Knowing the policy completely characterizes future behavior and therefore permits long-term behavior prediction. This model of agents behavior can be interpreted as a jump-Markov dynamical system, with the latent variable (index of the policy) governing the switches, and the policy describing the nonlinear motion dynamics.

We focus on the task of pedestrian path prediction in the assistive driving scenario. We evaluate the approach on a driving dataset augmented with environment maps and demonstrate that the method outperforms a number of baselines.

Common themes

While the applications discussed in each chapter may seem diverse and perhaps unrelated, the underlying ideas are connected. In Chapter 4 (chronologically, the starting point for the work), we focus on specifying an optimal behavior strategy for a mobile visual agent, tasked with finding a novel object in the scene. This has a natural formulation as an optimal control problem, with

the "reward" at each stage being the amount of information provided by the measured image, and the "goal" being the minimization of uncertainty on the variable of interest. The framework is related to partially observable Markov decision processes, "value of information", and "experiment design" in statistics. In Chapter 3, we use the same framework on a different problem. Whereas in Chapter 4, the "control" is the choice of which future data to acquire, here, in Chapter 3, the "control" is the choice of which data to process. We aim to minimize uncertainty of the semantic categories assigned to each pixel, while running as few object detectors as possible, on as few frames as possible. This, again, is formulated as an "experiment design" problem. In Chapter 5, we focus on the scenario where the environment consists of other intelligent, rationally behaving agents. Here, the "experiment design" aspect is missing: we are unable to control other agents. Instead, we posit that *they* are solving optimal control problems. By observing their behavior, we infer which optimal control problem is being solved, which then allows us to predict their future behavior.

The work described in Chapter 2 addresses an application similar to Chapter 3 – both deal with video segmentation. The similarities end at the application-level. In Chapter 2, the focus is on the ability to solve a large scale convex optimization problem. We do so by exploiting the recent first-order method, which permits us to solve the problem by performing sequences of simple and essentially componentwise operations.

CHAPTER 2

Causal Video Object Segmentation from Persistence of Occlusions

Occlusion relations inform the partition of the image domain into "objects" but are difficult to determine from a single image or short-baseline video. We show how long-term occlusion relations can be robustly inferred from video, and used within a convex optimization framework to segment the image domain into regions. We highlight the challenges in determining these occluder/occluded relations and ensuring regions remain temporally consistent, propose strategies to overcome them, and introduce an efficient numerical scheme to perform the partition directly on the pixel grid, without the need for superpixelization or other preprocessing steps.

2.1 Introduction

Partitioning the image domain into regions that correspond to "objects" is elusive absent an explicit definition of objects that has a measurable correlate in the image. Gestalt principles [Wer39] provide grouping criteria: continuity, regularity, proximity, compactness, the last of which (figure/ground, or occlusion) is best informed by video. Occlusions have been used extensively for grouping [WA94, BM98, BE99, AS12]. A nice feature of [AS12] is that grouping is obtained via a linear program: local ordering constraints provided by *occluder/occluded relations* are integrated to globally partition the image domain into *depth layers*. The challenge is that errors in determining occlusion relations can have a cascading effect.

Occlusions are usually detected from the residual of optical flow, but even assuming this detection is correct, *occluder relations* are non-trivial to determine. As we show in Fig. 2.2, correct determination of the occluder requires either knowledge of the motion of the occluded region (which

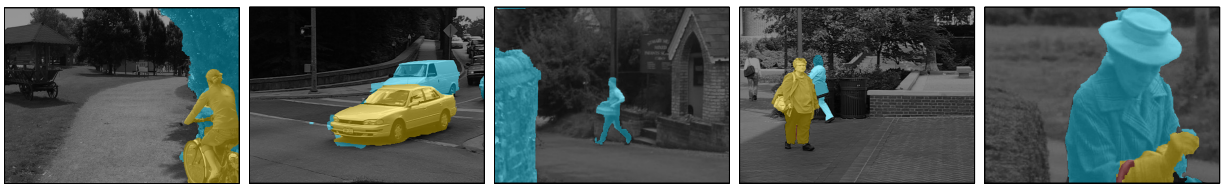


Figure 2.1: *Sample outcomes of our scheme: background* $c(x) = 0$ *(gray) and foreground layers* $c(x) = 1$, $c(x) = 2$, $c(x) = 3$ *indicated by* ■, ■, ■ *respectively. On the far right, our algorithm correctly infers that the bag strap is in front of the woman's arm, which is in front of her trunk, which is in front of the background. Project page:* `http://vision.ucla.edu/cvos/`

is undefined), or knowledge of its partition into regions. Hence the conundrum: to determine occlusion relations, so that objects can be segmented, we need to know the objects in the first place. The *first contribution* of our work is to break the conundrum by leveraging motion and appearance priors to hallucinate motion in the occluded region. With the *occluder/occluded* relations we can obtain a depth-layer partition for the image domain. In video, however, nuisances such as motion blur, quantization, scale, and lack of motion can cause layer segmentation to fail. Thus, the *second contribution* is a causal framework for integrating occlusion cues exploiting temporal consistency priors to partition the video into depth layers. Our *third contribution* is to make the solution of the resulting optimization problem efficient using a primal-dual scheme. Our proposed method is competitive to state-of-the-art approaches qualitatively in visual boundaries and quantitatively in numerical benchmarks, while processing video sequences causally, rather than in batch. Samples from scheme are shown in Fig. 2.1.

The chapter is organized as follows: we set up our problem in Sec. 2.2. We describe our first contribution in determining occluder relations in Sec. 2.2.1 and how we leverage prior work [AS12] in Sec. 2.2.2. Sec. 2.3 explores how we causally integrate cues to construct priors for foreground regions in Sec. 2.3.1, obtain persistent object boundaries in Sec. 2.3.2, and aggregate occluder relations in Sec. 2.3.3—components culminating in our final model in Sec. 2.3.4. Implementation and optimization details are covered in Sec. 2.4–2.5, including our approach for hallucinating motion in the occluded regions in Sec. 2.4.2. Empirical evaluation appears in Sec. 5.3, where we show that the typical failure modes of prior approaches stemming from unreliable occlusion relations are mitigated.

2.1.1 Related work

Our method incrementally processes the video and provides segmentation masks for the objects, along with their relative depth ordering. As such, it is related and directly builds upon work in *video segmentation* and *depth layer estimation*.

Video segmentation aims to partition a video sequence into a collection of non-overlapping regions with unique labels. Plethora of work exists that achieves this goal using motion, appearance, or combinations of the two [RM07, BT09, GKH10, VAP10, LKG11, XXC12, ZJS13, PF13, OMB14]. Many of these methods do not have an explicit model for "objects" in the video, and as the result, the segmentation does not respect object boundaries: multiple regions can span the same object, or regions can leak beyond edges. We refer to this as video oversegmentation.

Video *object* segmentation attempts to mitigate the problem of oversegmentation, and to assign a single regions to objects. The problem can be cast as a multi-label classification, in which a unique label is attached to each object [OMB14], or as a binary "foreground"/"background" classification [LKG11, ZJS13, PF13]. In [PF13], "objects" are regions that move differently from background *throughout the duration of the video*. The framework can be susceptible to objects that move for only a fraction of the video shot. This issue is addressed in the model used in [OMB14], which is able to detect an object, so long as it moves relative to background *once*; our approach is similar. In [LKG11, ZJS13], "objects" are salient and dominant regions – ones that appear frequently and are temporally consistent. This assumption is valid as long as the video is pre-segmented into shots, and the shot contains few objects. It is less justified for streaming video, where objects continuously enter and leave the scene. We make no assumptions on how frequently objects appear in the video, and have no difficulties detecting objects that appear in the video only for a fraction of the frames.

Many of the approaches perform segmentation *offline* (or *non-causally*), with the entire video sequence available for processing [OMB14, LKG11, ZJS13, PF13]. This is problematic in situations where the video is "endless" (as in robotics or surveillance applications), and does not scale well as the video length increases. Streaming video segmentation methods that process small groups of frames in batch have been proposed before [VAP10, XXC12]; however, as our algo-

rithm is truly *online* (or *causal*), it is closer related to tracking-like methods, e.g. [RM07] or [BWS09, CF13] who, unlike us, require manual initialization.

Depth layers estimation work attempts to infer regions' depth ordering, typically along with simultaneously estimating motion and the regions' segmentation masks [DP91, WA94, BM98, JF01, SDC04, JYS08, KTZ08, CF13]. The problem is formulated jointly, with the understanding that estimation of each component helps in estimating the others. The resulting problem is nonconvex, and is optimized with an expectation-maximization scheme or within a Bayesian framework, and requires a substantial computational effort. For scalability reasons, we separate motion estimation from inferring segmentation and depth ordering, and focus on the latter, which can then be formulated as a convex problem solvable with an efficient first-order computational scheme [CP11].

The idea of using occlusions to help in estimating the depth order is not new, and was used by [WA94, BM98], by [BE99] (who assume static camera), and more recently by [AS12], where segmentation and depth layers were solved within a convex optimization framework, given motion. That work is extended in the present thesis. Another interesting approach is [AYR14], where a very similar approach is taken to solve for depth ordering, assuming that the image segmentation is given and only the regions' depths are unknown. Instead of using occlusions from video to infer depth order, it uses local cues from a single image.

2.2 Video segmentation with layers

Let $I_t : D \to \mathbb{R}^3$ be an image of a video $\{I_t\}_{t=1}^T$ defined on the domain $D \subset \mathbb{R}^2$. We seek to partition D into regions, each associated with an integer *depth order*, represented by a function $c_t : D \to \mathbb{Z}_+$ indicating to which layer each pixel belongs. A layer is then $c_t^{-1}(i) = \{x \in D | c_t(x) = i\}$, where $c_t(x) = 0$ denotes the background and larger values of c_t indicate "foreground" regions $c_t(x) = 1, 2, 3, \ldots$. The connected components of non-zero regions correspond to individual objects. It was shown by [BM98, AS12] that depth layers can be inferred from occlusion phenomena, that occur as a result of object or viewer motion, causing parts of the scene to become hidden and others revealed. These inform local order relations between surfaces in the

scene: when a surface becomes *occluded*, the image region where it projected becomes occupied by the *occluder*, which is therefore closer to the viewer. These occluder-occluded relationships can be used as cues for segmenting regions in the image that back-project to distinct objects in the scene.

2.2.1 The "occluder" and the "occluded"

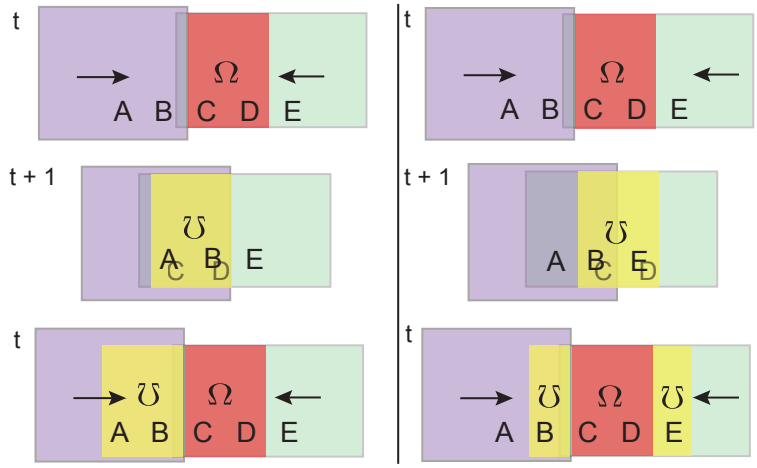


Figure 2.2: *Initial (top) and final (middle) views of a smaller square sliding under a larger one, producing an occluded region* Ω *in red (subscripts dropped for readability), with two alternate hypotheses (left and right) for the occluder (*$\mho$*) in yellow. Via* w^t_{t+1} *in the occluded region, these produce different constraints (bottom).* Left*:* Ω *moves with* E *and slides under* $\mathsf{A} \cup \mathsf{B}$*.* Right*: the occluder is split in two—*B *occludes* C *and* E *occludes* D*. Disambiguation requires either the motion in* Ω*, which is undetermined as it is occluded, or the object segmentation, which is the final goal.*

Under the assumptions of Lambertian reflection, constant illumination, and co-visibility typically implicit in most optical flow algorithms, $I_t(x)$ is related to $I_{t+1}(x)$ by the brightness-constancy equation

$$I_t(x) = I_{t+1}(w_t^{t+1}(x)) + n_t(x),\ x \in D \setminus \Omega_t^{t+1}(x), \tag{2.1}$$

where w_t^{t+1} is the deformation field that warps the domain of I_t into I_{t+1} and n_t lumps together all un-modeled phenomena and violations of the assumptions. Often, w_t^{t+1} is represented by the optical flow field v_t^{t+1} by $w_t^{t+1}(x) = x + v_t^{t+1}(x)$. The above holds on the entire image domain except in the *occluded regions* Ω_t^{t+1}, where surfaces visible at time t are no longer visible at $t+1$.

In this region, the optical flow is not defined, but can be *extrapolated* from the "co-visible" regions via regularization. Occluded regions are easy to find as a byproduct of optical flow estimation [ARS12], as they yield a large residual n_t via backward flow. What is not easy to find is the *occluder*.

The defining characteristic of the occluder point $y_t^c \in \mathcal{V}_t^{t+1}$ (the occluder region) of the point $y_t \in \Omega_t^{t+1}$ (the occluded region) is

$$w_t^{t+1}(y_t^c) = y_t. \tag{2.2}$$

This equation is somewhat unintuitive as the left hand-side lives in the domain of the image at time $t+1$ whereas the right-hand side is defined only at time t. This can be interpreted as

$$y_t^c = w_{t+1}^t(y_t), \tag{2.3}$$

which is completetely agnostic of the motion of the occluded region.

Consider Fig. 2.2: The occluded region, $\mathsf{C} \cup \mathsf{D}$, could slide under the larger rectangle, and become occluded by $\mathsf{A} \cup \mathsf{B}$. However, C and D could also actually correspond to different objects, and move independently. In this case, B could be the occluder of C and E could be the occluder of D. To disambiguate between these two hypotheses, we need to know either the motion of D, which is not possible since it is occluded, or the object partition, which is our goal in the first place. In the example in question, using (2.2) would favor the hypothesis of B occluding C and E occluding D (right half of Fig. 2.2). This would yield two ordering constraints, $c(\mathsf{B}) > c(\mathsf{C})$ and $c(\mathsf{E}) > c(\mathsf{D})$ that hinge on the occluded region and impose no constraints between the visible regions B and E. The latter constraint is also incorrect in the example (Fig. 2.2 bottom right).

However, while the motion in the occluded region is not determined, it can be hallucinated exploiting regularization priors. Even with a coarse estimate of the motion of D, we could determine if it moves similarly to E, in which case it cannot be occluded by it and must instead be occluded by B. Therefore, in our approach we extrapolate motion to the occluded region, so as to attribute it to a possible occluder. In Sec. 2.4.2, we discuss how to exploit natural image and motion priors to achieve this. Of course, one could resort to such priors and photometric characteristics of the

occluded region to directly determine the grouping of C, D, and E. But again if this was easy, we would have already solved the problem of object segmentation.

2.2.2 From local ordering constraints to layers

In [AS12], the following convex model for inferring c_t from occlusion cues was proposed:

$$\begin{aligned} c_t^* = \arg\min_{c_t : c_t \geq 0} \int_D g_t(x) |\nabla c_t(x)| dx \\ \text{s.t. } c_t(y^c) - c_t(y) \geq 1 \ \ \forall (y^c, y) \in O_t \end{aligned} \tag{2.4}$$

where in each pair $(y^c, y) \in O$, y^c lies on the occlud*ing* surface, and y lies on the surface that was (will be) occlud*ed* in the previous (next) frame. The objective $\int_D g_t(x)|\nabla c_t(x)|dx$ is just weighted total variation (TV), with the data-dependent affinity weights (denoted by $g_t(x)$) being small at image and motion boundaries and large otherwise. Note that the "data-term" enters the optimization as a set of constraints which require occluded-occluder pairs to lie in different layers: in particular, the occluder must lie in the closer layer (higher values of c_t). An overview of this approach is shown in Fig. 2.3. While this optimization problem relaxes the integer constraint ($c_t : D \to \mathbb{Z}_+$), empirically the solutions are piecewise constant and integer valued.

Although this model is formulated for a single time instant t, three frames $(t-1, t, t+1)$ are necessary to obtain occlusion cues. However, they are typically not sufficient when small inter-frame motion produces unreliable occlusion constraints. Next, we exploit temporal persistence to overcome this problem.

2.3 Incorporating motion cues causally

Our causal framework leverages a rich history of image frames, the segmentation cues from those frames (occlusions and weights), and previous layer estimates to facilitate segmentation in the current frame. Large-displacement propagation of these cues via w_t^{t-1} is unstable, rendering cues unusable. But when the motion becomes large, occlusions become easier to detect, making the

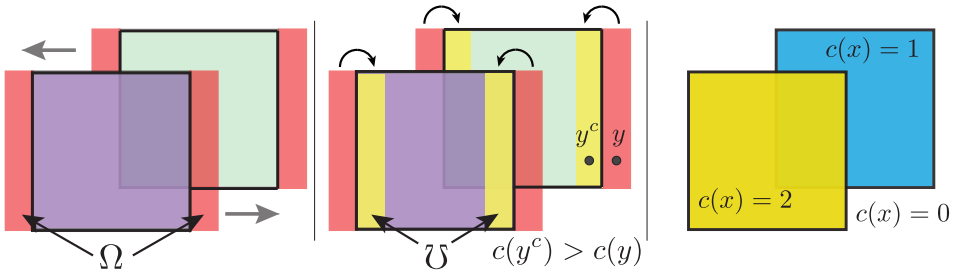


Figure 2.3: Left: *The motion of two objects generate occlusions and disocclusions (both denoted by Ω, shown in red).* Middle: *each occluded region is attributed to a local occluder ($\mho$, shown in yellow). Occluder-occluded relationship constrains objects' depth-order.* Right: *resulting depth layers.*

past unnecessary for segmentation. Thus, these cues are complementary—when the motion is large, sufficient occlusion cues are produced, and w_t^{t-1} is erroneous. When the motion is small, occlusion cues are few, but propagation is reliable. This motivates an adaptive integration of cues based on motion. A weighting function $m_t(x) \doteq \alpha \exp(-|v_t^{t-1}|/\mu_v)$ is used, where $\alpha \in [0, 1]$, v_t^{t-1} is the optical flow, and μ_v is the mean value of v for this frame. The weight decreases with large motion, regardless of how long ago it occurred. The following sections describe the temporal cues leveraged in our framework. The variable being optimized over is always c_t, and c_{t-1} is always available as a result of previous optimization.

2.3.1 Once an object, always an object

Layer values $c_t(x)$ are not constant over time, as objects can move in front of one another and switch order of their distance to the viewer. However, once separated from the background, objects cannot become background again, even if they stop moving and produce no occlusion relations.

This can be enforced causally using the prior segmentation result (c_{t-1}) via a (convex) constraint:

$$c_t(x) \geq 1 \;\; \forall x \in F, \;\; F = \{c_{t-1}(w_t^{t-1}(x)) \geq 1\}, \tag{2.5}$$

where F is the indicator of the previous frame's foreground warped into the current frame[1]. To

[1]Throughout the chapter we will rely on availability of the results and cues at time $t-1$ when computing the solution at time t. The correspondence between points at $t-1$ and t is obtained via optical flow w_t^{t-1} in the following way: a point at location x at time t corresponds to $w_t^{t-1}(x)$ at $t-1$.

mitigate errors in prior segmentations, we relax the constraint and penalize violations with a hinge loss:

$$\int_D \kappa_t(x) \max\big(0, 1 - c_t(x)\big) dx \tag{2.6}$$

with κ_t being the cost of violating the constraint. Choosing $\kappa_t(x) = 0$ for x outside F allows us to write the penalty as an integral over entire image domain D. As $\kappa_t(x) \to \infty$ for $x \in F$, the hard constraint (2.5) is recovered.

The cost of violating the constraint is computed recursively, with initial condition $\kappa_1(x) = 0$, as

$$\kappa_t(x) = m_t(x)\kappa_{t-1}(w_t^{t-1}(x)) + \mathbb{1}\{c_{t-1}(w_t^{t-1}(x)) \geq 1\}$$

where $\mathbb{1}$ is a characteristic function ($\mathbb{1}\{X\} = 1$ if X is true, and is 0 otherwise). This *foreground prior* boosts $\kappa_t(x)$ wherever the corresponding points are labeled as foreground in the previous frame and diminishes it over time and motion as described above. As demonstrated in Fig. 2.4, whenever motion is small, instantaneous occlusion cues are insufficient to perform segmentation, and this notion of temporal consistency is helpful.

To avoid the entire image domain from becoming foreground, we introduce an additional regularization penalizing layer values:

$$\tau \int_D c_t(x) dx. \tag{2.7}$$

This is similar to the regularization used in [AS12], although they use the ℓ_∞ norm, whereas here we use ℓ_1. This term encourages pixels to lie in the background layer, unless sufficient evidence pushes them into the foreground.

2.3.2 Layer unity prior and aggregated TV weights

While depth-layer *values* are not persistent, their boundaries are. Unless objects split or merge, we have

$$\mathbb{1}\{\nabla c_t(x) \neq 0\} = \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) \neq 0\}. \tag{2.8}$$

This is a nonconvex constraint. However, enforcing $\nabla c_t(x) = 0$ wherever $c_{t-1}(w_t^{t-1}(x)) = 0$ is simple (a linear constraint), and its relaxed version with a hinge loss and associated cost $u_t(x)$

Figure 2.4: *c_{t-1} (column 1) is used to compute the foreground prior (κ_t) (column 2). Without κ_t, the resulting c_t completely misses the objects (column 3), however with κ_t, c_t succeeds (last column). Note c_{t-1} and c_t look very similar—κ_t helps most during small-baseline motion when occlusion cues are weak but c_{t-1} easily predicts c_t.*

is equivalent to increasing weights in TV regularization [2]. This leaves the hard part: enforcing $\nabla c_t(x) \neq 0$ wherever $c_{t-1}(w_t^{t-1}(x)) \neq 0$. To remain within a convex optimization framework, we treat this as a bias and set the corresponding $u_t(x)$ to be negative, which *decreases* the corresponding TV weights (which are kept nonnegative to preserve convexity). This cost is also computed recursively, with initial condition $u_1(x) = 0$, as

$$u_t(x) = m_t(x)u_{t-1}(w_t^{t-1}(x)) + \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) = 0\} - \mathbb{1}\{\nabla c_{t-1}(w_t^{t-1}(x)) \neq 0\}.$$

We also perform temporal aggregation of the TV affinity weights. In each frame, we compute the *boundary strength* $\rho_t(x) \in \mathbb{R}_+$, as described in Sec. 2.4. The aggregated boundary strength $\bar{\rho}_t(x)$ is (with $\bar{\rho}_1(x) = 0$):

$$\bar{\rho}_t(x) = m_t(x)\bar{\rho}_{t-1}(w_t^{t-1}(x)) + \rho_t(x). \tag{2.10}$$

[2] Enforcing $\nabla c(x) = 0$ (a linear constraint) is equivalent to enforcing $|\nabla c(x)| \leq 0$ (a convex constraint). Relaxed as a hinge loss penalty with weights $u(x)$, this becomes

$$\int_D u(x) \max(0, |\nabla c(x)|)dx = \int_D u(x)|\nabla c(x)|dx. \tag{2.9}$$

But this is just TV regularization, and the objective already includes a penalty of the same form with weights $g(x)$. We conclude that the new "biased" penalty is $\int_D g'(x)|\nabla c(x)|dx$ with $g'(x) = g(x) + u(x)$.

The aggregated TV weights used in the optimization are:

$$g_t(x) = \max(0, 1 - \bar{\rho}_t(x) + u_t(x)). \tag{2.11}$$

2.3.3 Aggregated occlusion constraints

Instantaneous occlusion constraints (O_t) are accumulated into the aggregated constraints set $\bar{O}_t = w^t_{t-1}(\bar{O}_{t-1}) \cup O_t$, where past constraints $\bar{O}_{t-1}$ are propagated to the current frame by the motion of the occluder $w^t_{t-1}(y^c)$ (see Fig. 2.5). The base condition is $\bar{O}_1 = O_1$. The constraint penalty weights λ, computed by (2.4.1), are adjusted over time by:

$$\lambda_{t,i} = m_t(y^c_i)\lambda_{t-1,i} + \mathbb{1}\{c_{t-1}(w^{t-1}_t(y^c_i)) \geq c_{t-1}(w^{t-1}_t(y_i))\}.$$

Figure 2.5: *Occlusion cues from the current frame alone (O_t), with occluded points (Ω) in red and occluder points ($\mho$) in yellow, (column 1) fail to segment the objects (column 2). However, aggregating constraints over time ($\bar{O}_t$) (column 3) succesfully recovers all of them (last column).*

2.3.4 Overall model

The final model that incorporates occlusion cues, weights, foreground and unity priors is

$$c_t^* = \arg\min_{c_t \geq 0} \int_D g_t(x)|\nabla c_t(x)|dx + \tau \int_D c_t(x)dx + \int_D \kappa_t(x) \max\big(0, 1 - c_t(x)\big)dx + \sum_{\substack{i=1 \\ (y_i^c, y_i) \in \bar{O}_t}}^{N} \lambda_i \max\big(0, 1 - \big(c_t(y_i^c) - c_t(y_i)\big)\big) \tag{2.12}$$

where the first term (weighted TV) ensures that the result is piecewise constant, the second term (foreground prior) encourages regions to have nonzero layer values wherever $\bar{\kappa}_t(x)$ is large, the third (model selection) term prevents the creation of spurious layers, and the fourth is the penalty for violating the occlusion constraints.

2.4 Implementation details

For each frame, we incorporate appearance, motion, and edge information into the weights $\rho_t(x)$ in (2.10) as follows:

$$\rho_t(x) = 1 - \Big(\beta_I h(|\nabla I(x)|) + \beta_E h(E(x)) + \beta_v h(|\nabla v_t^{t+1}(x)|)\Big)$$

where $h(x) = \exp(-x/\mu_x)$, μ_x is the average value of x. $E(x) \in [0, 1]$ is the output of an edge detector [DZ13] with $E(x) \approx 1$ at the boundaries. In our experiments, $(\beta_I, \beta_E, \beta_v) = (0.2, 0.4, 0.4)$. Following [PF13], we also adjust the motion term by the difference in flow angles at the pixels where flow magnitude is small.

2.4.1 Occlusion constraint weights

Often the occluded and occluding surfaces differ in appearance, motion, and are separated by a strong image boundary, suggesting λ be computed in a fashion similar to (2.4):

$$\lambda_i = \eta_i\Big(1 - \Big(\beta_I h(|I(y_i^c) - I(y_i)|) + \beta_E h(\hat{E}(y_i^c, y_i)) + \beta_v h(|v_t^{t+1}(y_i^c) - v_t^{t+1}(y_i)|)\Big)\Big)$$

where the gradient operator is replaced by a difference between appearance, motion, and edge statistics of y^c and y. $E(x)$ is also replaced by $\hat{E}(x_1, x_2)$ – the strongest edge response on the line connecting y^c and y. We additionally validate y^c and y as an occluder-occluded pair with weight η, which measures the degree to which y and y^c move toward each other. Indeed, unless they do so, $\mho_t^{t+1}$ cannot take the place of Ω_t^{t+1}, i.e. when $\eta(y^c, y)$ in

$$\begin{aligned} \Delta(y^c, y) &= (v_t^{t+1}(y^c) - v_t^{t+1}(y))^T (\frac{y^c - y}{\|y^c - y\|_2}) \\ \eta(y^c, y) &= \max(0, 1 - \exp(-\theta\ \Delta(y^c, y))) \end{aligned} \tag{2.13}$$

is small, then y^c is less likely to occlude y. We choose $\theta = 2$ so that $\Delta(y^c, y) = 1$ yields a high score. $\lambda_i \approx 1$ whenever the appearance and motion of y^c and y are "different" and the points are moving toward each other. Finally, assuming that the occluded and occluding surfaces differ in appearance allows us to locally perturb constraints, with the goal of correcting them. This procedure is described in [TKS15]. Altogether, this evidence alters the constraint weights to help us discount potentially erroneous constraints, which occur due to inevitable errors in optical flow and occlusion estimation.

2.4.2 Flow extrapolation over the occlusion region

As noted in Sec. 2.2.1, $v_t^{t+1}(x)$, for $x \in \Omega_t^{t+1}$ is undefined (2.1) and filled in by the regularizer, which corresponds to enforcing priors on motion. The simplest priors rely solely on continuity, tending to smooth motion boundaries, while more sophisticated ones attempt to preserve them. We use the cross-bilateral filter [ED04] to enforce such priors on v_t^{t+1} in the occluded regions based on the backward flow v_t^{t-1}:

$$\hat{v}_t^{t+1}(z) = \frac{1}{V_z} \int_D v_t^{t+1}(x) P(x \notin \Omega_t^{t+1}) \mathcal{G}(\|v_t^{t-1}(x) - v_t^{t-1}(z)\|, \sigma_v) \mathcal{G}(\|x - z\|, \sigma_x) dx, \tag{2.14}$$

where $\hat{v}_t^{t+1}$ is the filtered forward flow, $P(x \notin \Omega_t^{t+1})$ is the probability of x being visible, $\mathcal{G}$ is the gaussian kernel $\mathcal{G}(x, \sigma) = \exp(-\|x\|^2/2\sigma^2)$, and V_z is a normalization term. We can filter the backward flow $\hat{v}_t^{t-1}$ by changing the $t+1$ ($t-1$) indices to $t-1$ ($t+1$). Extrapolating flow is

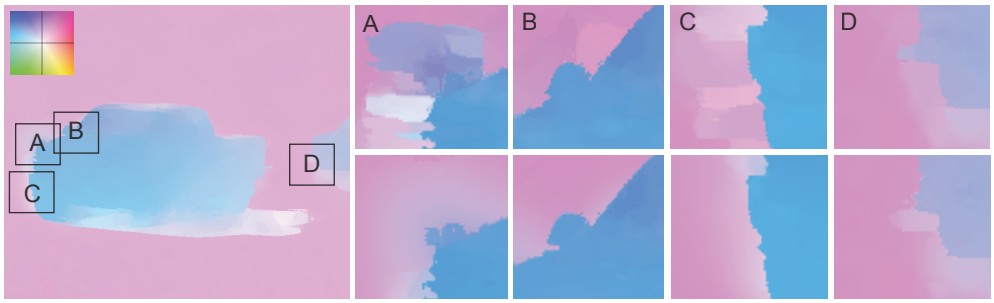


Figure 2.6: *Cross bilateral filtering extrapolates flow in Ω via motion and appearance priors, facilitating reliable occluder determination. Left: The extrapolated motion field $\hat{v}_t^{t+1}$. Boxes highlight occluded regions where notable change (often improvement) occurs. Right: For each box, v_t^{t+1} (top) and $\hat{v}_t^{t+1}$ (bottom) are shown.*

key to determining the occluder (Fig. 2.2), but cannot be proven "correct" as it hinges critically on the choice of prior. v_t^{t+1} is computed using publicly-available code [SRB10] ("classic-nl"), and occlusions are computed by thresholding the residual image. See [TKS15] for further details.

2.4.3 Foreground prior region

In practice, motion estimation makes mistakes near object boundaries (e.g. occluded regions). When computing κ_t, we first warp c_{t-1} to the current frame and then use morphological operations to erode the edges proportionally to the magnitude of the flow in that region. This ensures the prior does not leak outside of the object regions, but produces a poor estimate near the boundaries. To help recover the structure of these edges, we incorporate a set of local shape classifiers as in [BWS09] to better capture and predict the shape of the object boundary, the details of which are in [TKS15].

2.5 Optimization

The optimization problem (2.12) is convex but large enough that off-the-shelf methods cannot solve it without resorting to superpixels or other pre-processing to reduce its dimension. Here we present an efficient numerical primal-dual scheme based on [CP11] that allows us to solve it on the pixel grid.

The indicator function – not to be confused with characteristic function $\mathbb{1}$ used above – of a set A is defined by $I_A(x) = 0$ for $x \in A$ and $I_A(x) = \infty$ for $x \neq A$. For a function f, the *convex conjugate* is defined as $f^*(y) \doteq \sup_x y^T x - f(x)$. The *prox operator* of f is defined as

$$\mathbf{prox}_{\sigma f}(y) \doteq \arg\min_x \frac{1}{2\sigma}\|x - y\|^2 + f(x). \tag{2.15}$$

Since the optimization is performed on a finite pixel grid, the depth values c can be written as a vector $c \in \mathbb{R}^n_+$, with c_i indicating layer value at i-th pixel. We denote by $\mathcal{D}$ the gradient operator represented by a matrix of finite differences (specifically, a concatenation of horizonal and vertical finite difference $\mathcal{D} = (\mathcal{D}_x; \mathcal{D}_y)$).[3] Weights associated with the edges are denoted by the diagonal matrix W. A difference matrix for occlusion constraints is denoted by $\mathcal{D}_{occ}$, and as before we use λ to denote the cost of violating the constraint. As before, τ is used for regularization, and κ as a weighted indicator of foreground region. We can then write the objective in shorthand as

$$\min_c \|W\mathcal{D}c\|_1 + \tau^T c + \kappa^T \max(0, 1 - c) + \lambda^T \max(0, 1 - \mathcal{D}_{occ}c) + I_{\{c\geq 0\}}(c). \tag{2.16}$$

Let $G(c) = \tau^T c + \kappa^T \max(0, 1-c) + I_{\{c\geq 0\}}(c)$. Let also $z_1 = \mathcal{D}c$, $z_2 = \mathcal{D}_{occ}c$, introduce functions $F_1(z_1) = \|Wz_1\|_1$, $F_2(z_2) = \lambda^T \max(0, 1 - z_2)$, and the dual variables y_1, y_2. The augmented Lagrangian follows as

$$\min_{z_1,z_2,c} \max_{y_1,y_2} F_1(z_1) + F_2(z_2) + G(c) + y_1^T(\mathcal{D}c - z_1) + y_2^T(\mathcal{D}_{occ}c - z_2), \tag{2.17}$$

or, equivalently, using the convex conjugates, as

$$\min_c \max_{y_1,y_2} G(c) - F_1^*(y_1) - F_2^*(y_2) + y_1^T\mathcal{D}c + y_2^T\mathcal{D}_{occ}c.$$

This saddle-point problem is addressed in [CP11], so we can apply their primal-dual algorithm

[3] Note that we use what is called the "anisotropic" total variation, i.e. $\sum_{ij}(|c_{i,j+1} - c_{i,j}| + |c_{i+1,j} - c_{i,j}|)$, rather than the "isotropic" one $\sum_{ij}\sqrt{(c_{i,j+1} - c_{i,j})^2 + (c_{i+1,j} - c_{i,j})^2}$. The former yields undesirable "metrification" artifacts. On the hand, the isotropic variant leads to a second order cone problem, rather than a linear program. It is straightforward to change our solver to use the isotropic penalty, however, if one does so, the solutions cease to be integer-valued.

Algorithm 1 Layer Solver

Initialize: Pick $\sigma_y, \sigma_c > 0$, $\sigma_y \sigma_c \leq \frac{1}{8}$, and $\theta \in [0, 1]$. Arbitrarily initialize feasible y_1^0, y_2^0, c^0. Set $\bar{x}^0 = c^0$.

Perform iterates for $k = 0, 1, 2, \ldots$**:**

$$
\begin{aligned}
y_1^{k+1} &= \mathbf{prox}_{\sigma_y F_1^*}\left(y_1^k + \sigma_y \mathcal{D} \bar{x}^k\right) \\
y_2^{k+1} &= \mathbf{prox}_{\sigma_y F_2^*}\left(y_2^k + \sigma_y \mathcal{D}_{occ} \bar{x}^k\right) \\
c^{k+1} &= \mathbf{prox}_{\sigma_c G}\big(c^k - \sigma_c(\mathcal{D}^T y_1^{k+1} + \mathcal{D}_{occ}^T y_2^{k+1})\big) \\
\bar{x}^{k+1} &= c^{k+1} + \theta(c^{k+1} - c^k).
\end{aligned}
$$

shown in Alg. 1.

Alg. 1 depends on the ability to compute proximal operators for G, F_1^* and F_2^*. All three operators have simple closed form solutions that require few arithmetic operations:

$$\mathbf{prox}_{\sigma G}\,(y) = \max\left(0, \min\left(y - \sigma\tau + \sigma\kappa, \max\left(1, y - \sigma\tau\right)\right)\right) \tag{2.18}$$

$$\mathbf{prox}_{\sigma F_1^*}(y) = \text{sign}(y)\min\{\text{diag}(W), |y|\} \tag{2.19}$$

$$\mathbf{prox}_{\sigma F_2^*}(y) = \min\left(\max\left(y - \sigma 1, -\lambda\right), 0\right) \tag{2.20}$$

These expressions are derived below.

Proximal operator for $G(c) = \tau^T c + \kappa^T \max(0, 1 - c) + I_{\{c \geq 0\}}(c)$

First define $g_i(c_i) = \tau_i c_i + \kappa_i \max(0, 1 - c_i) + I_{\{c_i \geq 0\}}(c_i)$, so that $G(c) = \sum_{i=1}^n g_i(c_i)$. This is done in anticipation of the prox-operator for G being separable[4]:

$$\mathbf{prox}_{\sigma G}(y) = \arg\min_{c \geq 0} \frac{1}{2\sigma}\|c - y\|_2^2 + \tau^T c + \kappa^T \max(0, 1 - c) \tag{2.21}$$

$$= \arg\min_{c \geq 0} \sum_{i=1}^n \frac{1}{2\sigma}(c_i - y_i)^2 + \tau_i c_i + \kappa_i \max(0, 1 - c_i) \tag{2.22}$$

$$= \left(\mathbf{prox}_{\sigma g_1}(y_1), \ldots, \mathbf{prox}_{\sigma g_n}(y_n)\right) \tag{2.23}$$

[4]We use the following "separable sum" rule.
If $f(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = g(x_1) + h(x_2)$, then $\mathbf{prox}_f(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = \begin{pmatrix} \mathbf{prox}_g(x_1) \\ \mathbf{prox}_h(x_2) \end{pmatrix}$

We now will derive the expression for $\mathbf{prox}_{\sigma g_i}(y_i)$. Since the general form of the expression will be the same for each i, we drop the index for clarity. For convenience, let $f(c) = \frac{1}{2\sigma}(c-y)^2 + g(c)$ (as shown in Fig. 2.7, f is a parabola with a kink), i.e. $\mathbf{prox}_{\sigma g}(y) = \arg\min f(c)$. Since f is strictly convex and the minimizer is unique, we initially ignore the constraint $\{c \geq 0\}$, eventually projecting the minimizer back onto the feasible domain if necessary. The subdifferential of $f(c)$ is

$$\partial f(c) = \begin{cases} \frac{1}{\sigma}(c-y) + \tau & \text{if } c > 1 \\ \frac{1}{\sigma}(c-y) + \tau - \kappa & \text{if } 0 \leq c < 1 \\ \frac{1}{\sigma}(c-y) + \tau - \kappa[0,1] & \text{if } c = 1 \end{cases} . \tag{2.24}$$

The optimality condition $0 \in \partial f(c)$ is satisfied when

$$c^* = \begin{cases} y - \sigma\tau & \text{if } y > 1 + \sigma\tau \\ 1 & \text{if } y \in [1 + \sigma(\tau - \kappa), 1 + \sigma\tau] \\ \max\big(0, y - \sigma(\tau - \kappa)\big) & \text{if } y < 1 + \sigma(\tau - \kappa) \end{cases} . \tag{2.25}$$

These three cases are shown in Fig. 2.7,

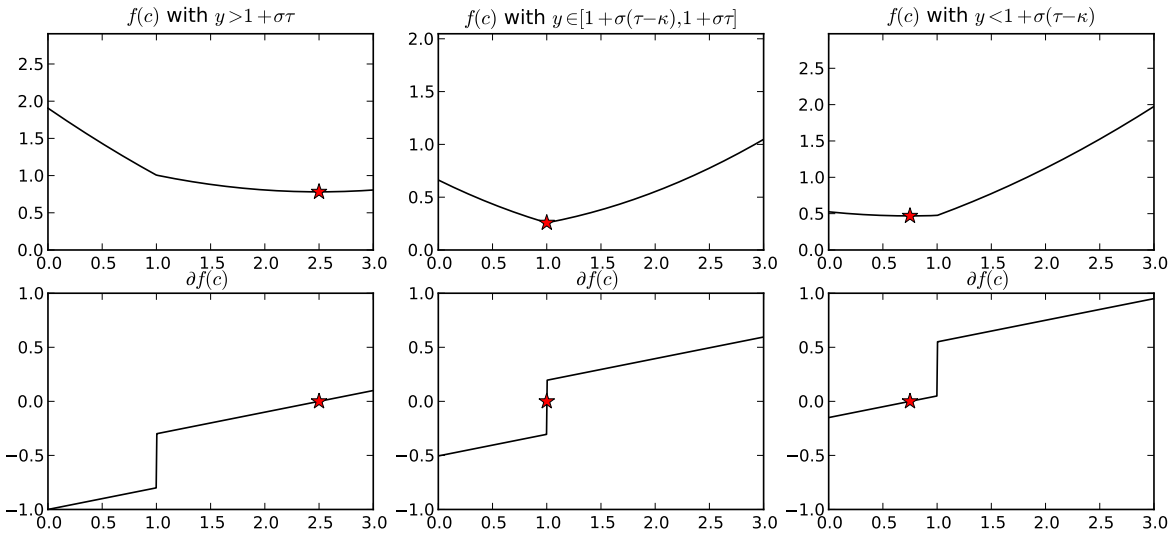


Figure 2.7: Top: $f(c)$ *shown for the three cases described in* (2.25). Bottom: *subdifferentials of corresponding* $f(c)$. *Shown in red are the pairs* $(c^*, f(c^*))$.

which can be rewritten as

$$\mathbf{prox}_{\sigma g_i}(y_i) = \max\Big(0, \min\big(y_i - \sigma(\tau_i - \kappa_i), \max\big(1, y_i - \sigma\tau_i\big)\big)\Big). \tag{2.26}$$

Using (2.23) and interpreting $\max$/$\min$ componentwise, the prox-operator for G is written as

$$\mathbf{prox}_{\sigma G}(y) = \max\Big(0, \min\Big(y - \sigma(\tau - \kappa), \max\big(1, y - \sigma\tau\big)\Big)\Big). \tag{2.27}$$

Proximal operator for $F_1^*(y)$, with $F_1(y) = \|Wy\|_1$

We note that this operator appears in all problems that involve TV-regularization, that our derivation below is by no means novel, and that it is included only for completeness. Since the function is a (weighted) norm, we expect its convex conjugate (i.e. F_1^*) to be a (weighted) indicator of a dual norm ball. Moreover, we expect the proximal operator of the indicator function to be the projection onto the feasible set [BV04]. The steps below verify it.

$$F_1^*(z) = \sup_y y^T z - F_1(y) \tag{2.28}$$

$$= \sum_{i=1}^{n} \max_{y_i} \underbrace{y_i z_i - w_i |y_i|}_{g_i(y_i)} \tag{2.29}$$

The maximum of each term $g_i(y_i)$ can be determined as

$$\max_{y_i} g_i(y_i) = w_i\Big(\max_{y_i} y_i \frac{z_i}{w_i} - |y_i|\Big) \tag{2.30}$$

$$= \begin{cases} 0 & \text{if } |\frac{z_i}{w_i}| \leq 1 \\ +\infty & \text{else} \end{cases}, \tag{2.31}$$

so $F_1^*(z)$ is the indicator of the weighed ℓ_∞ ball:

$$F_1^*(z) = \begin{cases} 0 & \text{if } \|W^{-1}z\|_\infty \leq 1 \\ +\infty & \text{else} \end{cases}. \tag{2.32}$$

Using $\text{diag}(W)$ to write the main diagonal as a vector (i.e. $\text{diag}(W) = (W_{11}, \ldots, W_{nn})$), the prox operator for F_1^* can be written as

$$\begin{aligned} \mathbf{prox}_{\sigma F_1^*}(y) &= \arg\min_z \frac{1}{2\sigma}\|z-y\|_2^2 + F_1^*(z) && (2.33)\\ &= \text{sign}(y)\min\{\text{diag}(W), |y|\}. && (2.34)\end{aligned}$$

Proximal operator for $F_2^*(y)$, with $F_2(y) = \lambda^T \max(0, 1-y)$

The convex conjugate is

$$\begin{aligned} F_2^*(z) &= \max_y z^T y - F_2(y) && (2.35)\\ &= \sum_{i=1}^{n} \max_{y_i} \underbrace{y_i z_i - \lambda_i \max(0, 1-y_i)}_{g_i(y_i)}. && (2.36)\end{aligned}$$

Notice that when $z_i > 0$, $\max_{y_i} g_i(y_i) \to \infty$ (achieved with $y_i \to \infty$) and similarly $\max_{y_i} g_i(y_i) \to \infty$ when $z_i < -\lambda_i$ (achieved with $y_i \to -\infty$). These cases are shown in Fig. 2.8. So, $F_2^*(z) = \infty$ for $z \notin [-\lambda, 0]$. For $z \in [-\lambda, 0]$ we compute the subdifferential of $g_i(y_i)$ (both shown in Fig. 2.8):

$$\partial g_i(y) = \begin{cases} z_i + \lambda & y_i < 1 \\ z_i + \lambda\big[0,1\big] & y_i = 1 \\ z_i & y_i > 1 \end{cases}. \tag{2.37}$$

The optimality condition $0 \in \partial g_i(y_i^*)$ suggests that when $z_i \in (-\lambda_i, 0)$, $y_i^* = 1$. In other words, for that interval, we have $\max_{y_i} g_i(y_i) = z_i$. On the interval boundaries y^* is not unique (when $z_i = -\lambda$, $y^* \in (-\infty, 1]$, and when $z_i = 0$, $y^* \in [1, \infty)$), but $\max_{y_i} g_i(y_i) = z_i$ still holds. So the convex conjugate can be written as follows:

$$F_2^*(z) = \begin{cases} 1^T z & \text{if } z \in [-\lambda, 0] \\ +\infty & \text{else} \end{cases}. \tag{2.38}$$

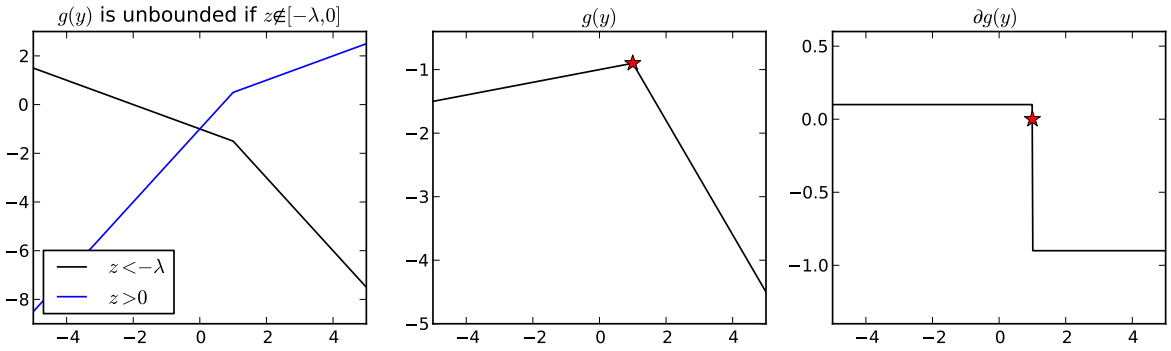


Figure 2.8: Left: *$g(y)$ plotted for $z \notin [-\lambda, 0]$. As noted in text, these functions are unbounded.* Middle: *$g(y)$ with $z \in (-\lambda, 0)$. This is a piecewise linear function with a unique maximizer (if $z \in \{-\lambda, 0\}$, the function is bounded, but the maximizer is not unique).* Right: *subdifferential of $g(y)$ on the left.*

To solve for $\mathbf{prox}_{\sigma F_2^*}(z)$, we first find $\operatorname{argmin}_x \frac{1}{2\sigma}\|z - x\|_2^2 + 1^T x$ (ignoring the constraint $x \in [-\lambda, 0]$), and then project it onto the feasible set:

$$\mathbf{prox}_{\sigma F_2^*}(z) = \arg\min_x \frac{1}{2\sigma}\|z - x\|_2^2 + F_2^*(x) \tag{2.39}$$

$$= \min\big(\max\big(z - \sigma 1, -\lambda\big), 0\big). \tag{2.40}$$

2.6 Experiments

Our method produces a segmentation of the video into depth layers. Unfortunately, no benchmark dataset is available to evaluate it directly. However, our method can be modified to produce binary and multi-label segmentations; leveraging this, we evaluate the algorithm on two datasets: MoSeg [OMB14] (designed for video *object* segmentation with no consideration for depth ordering), on which we focus, as well as BVSD [GNC13] (designed for video segmentation).

Evaluation methodology. We follow the process described in [OMB14]. The dataset contains 59 sequences, ranging from 19 to 800 frames. Each has pixel-wise ground truth annotation for a sparse subset of frames (3–41). As in [OMB14], we report *precision*, *recall*, *F-measure*, and the number of extracted objects (regions with F-measure ≥ 0.75). For multi-label segmentation tasks, we treat each connected component of the depth layers as a unique "object". We also evaluate on foreground/background (FG/BG) video object segmentation, which come directly from depth

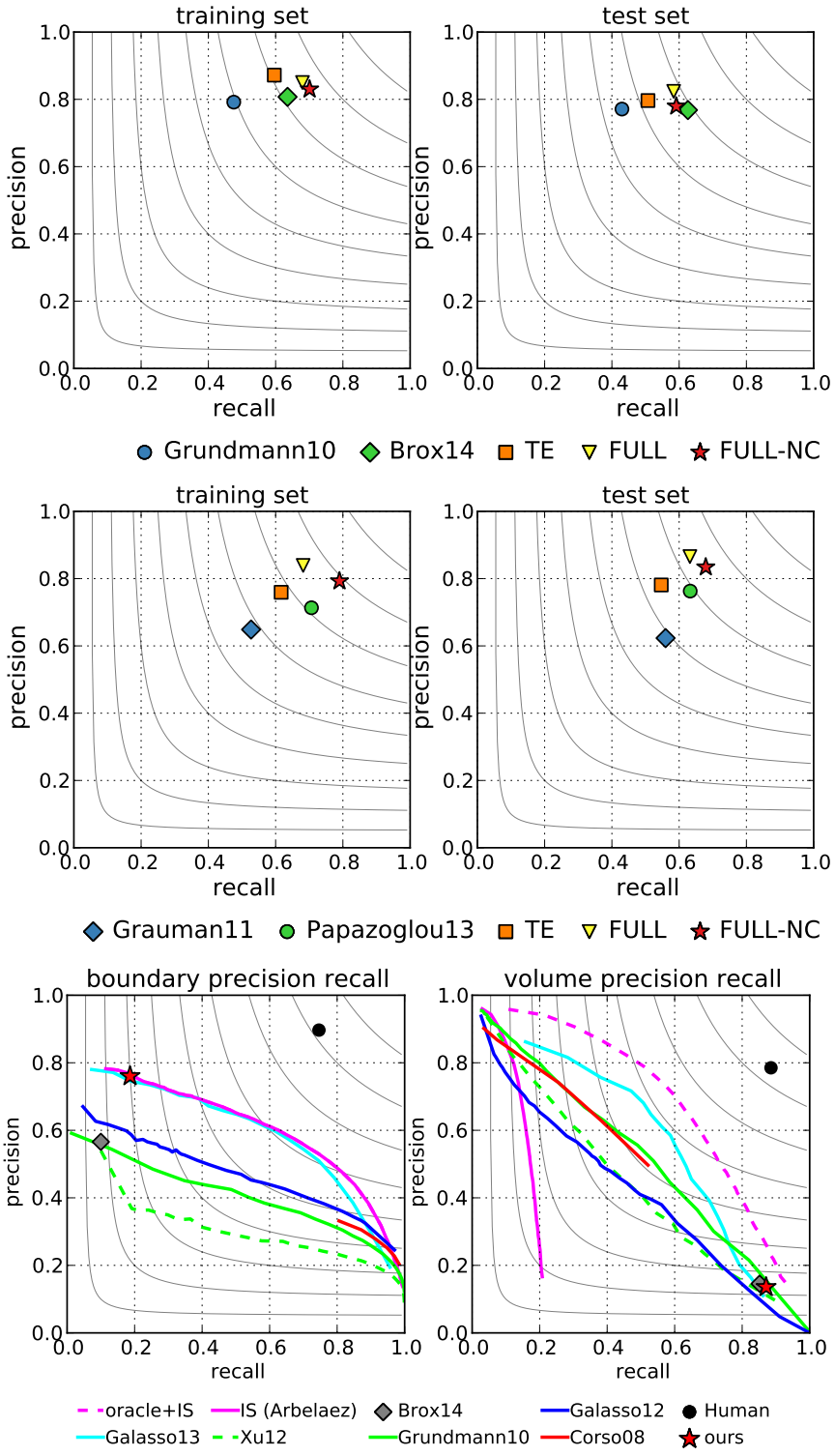


Figure 2.9: *Quantitative results. From top to bottom: MoSeg multi-label segmentation, MoSeg FG/BG segmentation. BVSD multi-label segmentation.*

layers as $FG \doteq \{x : c(x) \geq 1\}$, $BG \doteq \{x : c(x) = 0\}$. *Precision*, *recall*, and *F-measure* are reported on the ground truth annotations converted to binary masks. Note we cannot evaluate "number of extracted objects" in the FG/BG scenario.

The methods that we compare against ([GKH10, LKG11, PF13, OMB14]) are non-causal and "batch", whereas our method is causal. Since we do not know the future, we do not detect objects until they undergo sufficient motion, which sometimes causes us to miss objects in the beginning of video sequences. To fairly compare against non-causal methods, we also perform a non-causal evaluation (reported as "NC")—we run our algorithm forward in time to accumulate all priors, and then backward in time. The latter half is used for evaluation.

Effects of system components. In Sec. 2.3 we described individual components of our model and showed examples where they improved results (see Fig. 2.4,2.5). Here we quantify this improvement. We evaluate [AS12] ("BASIC"), their temporal extension ("TE"), foreground-background prior (Sec. 2.3.1, "FG"), and the full model ("FULL"). In addition, we evaluated the full model without flow extrapolation (Sec. 2.4.2) to understand its effects ("NOFE"). These results are reported in Table 2.1. "BASIC" does not use long-term temporal information. "TE" integrates weights using previous segmentations, increasing the cost of making a cut away from object boundaries. "FG" discourages previously segmented regions from falling into background. "FULL" is a combination of all components.

The "BASIC" method does not use temporal information, so on the multi-label benchmark, whenever objects disappear (as they often do, due to insufficient motion) and re-appear, they are assigned a *new* object label[5]. Long-term integration helps avoid missed detections and propagates object labels throughout the sequence. Performance on the FG/BG evaluation suggests that objects are often not detected at all.

Precision decreases for the "FULL" system due to an increased number of "false positives"—often we detect more objects than labeled in the annotation (see Fig. 2.13). "NC" provides a small performance boost by allowing us to label objects before they move.

[5]Our "FULL" scheme misses objects due to insignificant motion less frequently. But if an object completely disappears and reappears later, it is still assigned a new label, since our model of objects does not have any notion of appearance. An extension is possible (see e.g. [HE05] who are nonetheless restricted to static backgrounds), but is beyond our scope here.

Multi-label segmentation

	Training set (29 sequences)				**Test set** (30 sequences)			
	P	R	F	$N/65$	P	R	F	$N/69$
BASIC	84.90	53.10	65.34	10	78.80	44.49	56.87	4
TE	**87.20**	59.60	70.81	17	79.64	50.73	61.98	7
FG	86.98	60.99	71.71	18	79.04	52.08	62.79	10
NOFE	86.67	58.06	69.54	14	80.71	50.64	62.24	8
FULL	85.00	67.99	75.55	21	**82.37**	58.37	**68.32**	17
FULL-NC	83.00	**70.10**	**76.01**	**23**	77.94	59.14	67.25	15
[GKH10]	79.17	47.55	59.42	4	77.11	42.99	55.20	5
[OMB14]	81.50	63.23	71.21	16	74.91	**60.14**	66.72	**20**

Binary segmentation

	Training set (29 sequences)				**Test set** (30 sequences)			
	P	R	F	-	P	R	F	-
BASIC	**89.99**	40.86	56.21	-	**93.21**	33.69	49.49	-
TE	75.94	61.64	68.05	-	78.11	54.68	64.33	-
FG	75.93	63.07	68.91	-	76.97	56.16	64.94	-
NOFE	68.92	66.09	67.48	-	74.27	53.99	62.52	-
FULL	83.92	68.19	75.24	-	86.54	63.20	73.05	-
FULL-NC	79.26	**78.99**	**79.12**	-	83.41	**67.91**	**74.87**	-
[LKG11]	64.86	52.70	58.15	-	62.32	55.97	58.97	-
[PF13]	71.34	70.66	71.00	-	76.29	63.29	69.18	-

Table 2.1: *Comparison of our approach (rows 4–5) to baselines using individual components (rows 1–3) and state-of-the-art (rows 6–7) on the MoSeg dataset.* **R**$\doteq$*recall,* **P**$\doteq$*precision,* **F**$\doteq$*F-measure,* **N**$\doteq$ *number of extracted objects.*

Video object segmentation. In Table 2.1 and Fig. 2.9 we report results of the comparison with multi-label dense motion segmentation [OMB14], video over-segmentation [GKH10], as well as binary (i.e. FG/BG) video object segmentation methods [LKG11, PF13]. On multi-label segmentation, we outperform [GKH10],[AS12], and [OMB14] in F-measure. The improvement from the latter is not great; however, note that unlike theirs, our method is causal and has a small memory footprint. We are not the best in terms of "number of extracted objects". As mentioned before, unless the object undergoes sufficient motion, it will not be detected. On FG/BG segmentation, we outperform [LKG11],[PF13], and [AS12].

Video segmentation. BVSD [SBM11a, GNC13] contains 40 training and 60 testing sequences, each up to 121 frames. Pixel-wise ground truth annotation is provided for a subset of frames. Video sequences are in HD; we resize images to 540×960. While we report results for a variety of algorithms [CSD08, AMF11, GCS12, XXC12] (with data from [GNC13]), our primary point of

comparison is [OMB14]. Performance is benchmarked using "boundary precision-recall" (BPR) and "volume precision-recall" (VPR) metrics. BPR is commonly used in image segmentation, while VPR quantifies the spatiotemporal overlap between machine-generated and ground-truth segmentations (see [GNC13] for details).

Video *object* segmentation algorithms are expected to be in high-precision regime in the BPR metric, and in high-recall regime in the VPR metric, which indeed both we and [OMB14] satisfy (see Fig. 2.9). We obtain $(P, R, F) = (0.760, 0.186, 0.299)$ and $(0.136, 0.870, 0.234)$ on BPR and VPR respectively, while they obtain $(0.566, 0.100, 0.170)$ and $(0.146, 0.852, 0.249)$. Sample results are in Fig. 2.12. Note that ground truth is often fine-grained – with objects spanned by multiple regions. Thus, on this benchmark, object segmentation methods will not obtain the best F-measure.

Timing. Given optical flow (which all video segmentation require as input), our algorithm takes 30s on a VGA image, on a standard desktop; most time is spent solving (2.12), but a GPU implementation can reduce this number.

Optimization. In this section we compare the timing performance of our first-order solver to several other standard solvers. First-order methods, like the one we use, are quick at finding a point close to the solution, but may be slow at finding the exact solution (often unnecessary in computer vision tasks); to reach high accuracy, one may need to perform a large number of iterations. On the other hand, each iteration is cheap – only linear in the number of variables (see Alg. 1 and (2.18)). Interior-point methods, on the other hand, reach high accuracy with a small number of iterations, but each iteration is very expensive (requiring a solution to a large linear system of equations). We compare to SDPT3[TTT99], SeDuMi[Stu98] (both accessible via [GB14]), and also to primal simplex, primal-dual simplex, and the interior-point method in MOSEK [ApS15]. We tested the methods on several simple problems (shown in Fig. 2.10, with varying dimensionality (defined by the size of the image, rather than the number of unknowns in the problem). These timing results are shown in Fig. 2.11. Our method was ran until per-pixel ℓ_1 error (that is, $\frac{1}{n}\|c^{opt} - c\|_1$) between the current and true solution reached a tolerance level[6]. We tested tolerances 10^{-2} and

[6]In all other experiments where the "ground truth" solution was not known, we ran the algorithm for a predefined number of iterations, possibly stopping when $\|c^{k+1} - c^k\|$ decreased below a tolerance level.

10^{-3}. When $\frac{1}{n}\|c^{opt} - c\|_1 < 10^{-2}$, we observed that the ℓ_∞ error was always within acceptable range (usually, $\|c^{opt} - c\|_\infty \in [10^{-2}, 10^{-1}]$, and since we desire an integer-valued solution, the critical threshold is 0.5). Our approach with 10^{-2} tolerance performs faster than the other solvers. When the tolerance is decreased to 10^{-3}, the simplex method appears faster, while the interior point approaches remain slower. While we did not use a GPU-based parallelization, we did use SSE instrinsics – these reduced cost-per-iteration by 3-4 times.

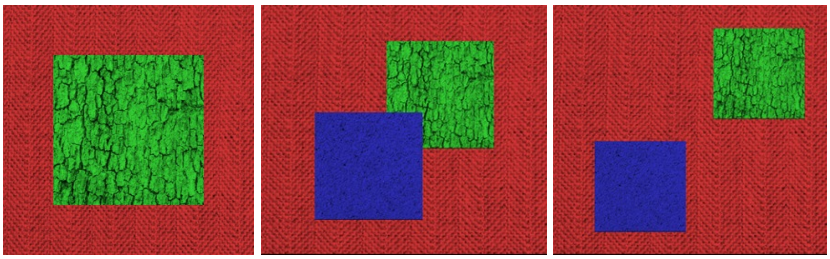


Figure 2.10: *Sample synthetic problems used to evaluate optimization method timing.*

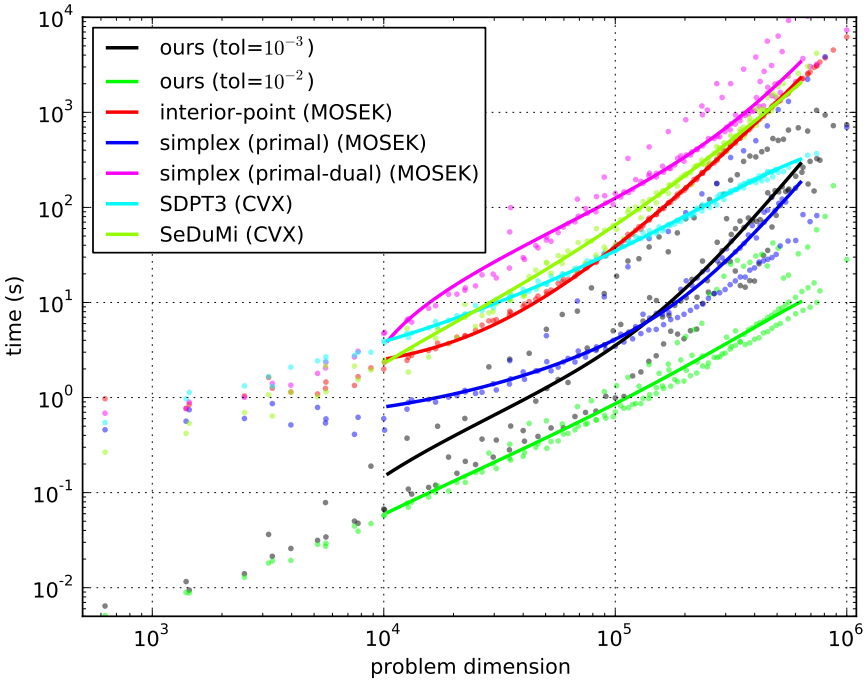


Figure 2.11: Performance of different optimization algorithms. See text for details.

2.7 Discussion

Occlusion relations inform the partition of the image domain into segments, but proper inference of such relations requires knowledge of the segments in turn. Rather than tackling an intractable chicken-and-egg problem, we use priors informed by Gestalt principles to arrive at a convex optimization scheme that can be efficiently solved with primal-dual methods. To compare with existing benchmarks, we converted our layers into "objects" and into "foreground/background". The evaluation highlights strengths and limitations of our method, with some of the latter due to the particular characteristics of the benchmarks. While our scheme still relies on decent optical flow and occlusion detection to bootstrap the layer segmentation, it is less prone to cascading failure than previous methods, as it better exploits priors on motion, appearance, and layer consistency.

Our objects are represented by integer labels, without any semantic annotations, which are necessary in a vast range of applications. However, it is possible to combine our convex problem with a similar convex multilabel segmentation problem [PCC09] used for semantic segmentation. This was done in [TAR13], who simultaneously obtain depth order and a semantic segmentation, by jointly optimizing for both. While semantic segmentation is important for many tasks, it is also very computationally demanding, and there are practical advantages in decreasing its computational cost. Thus, the next chapter addresses semantic segmentation from an *information-acquisition*, rather than from a *perception* viewpoint.

Figure 2.12: *Sample results on BVSD.* Left to right: *original image, ground truth, [GKH10],[OMB14], ours (object maps). Row 1: as reflected by BPR, our method produces accurate boundaries (see Fig. 2.9). Both actors are correctly segmented—the arm occluding the animal's body is a distinct depth layer. Row 2: "failure case"—complex motion and inaccurate flow can result in inaccurate segmentations. Row 3: failure case—object is not detected throughout the sequence due to lack of occlusions.*

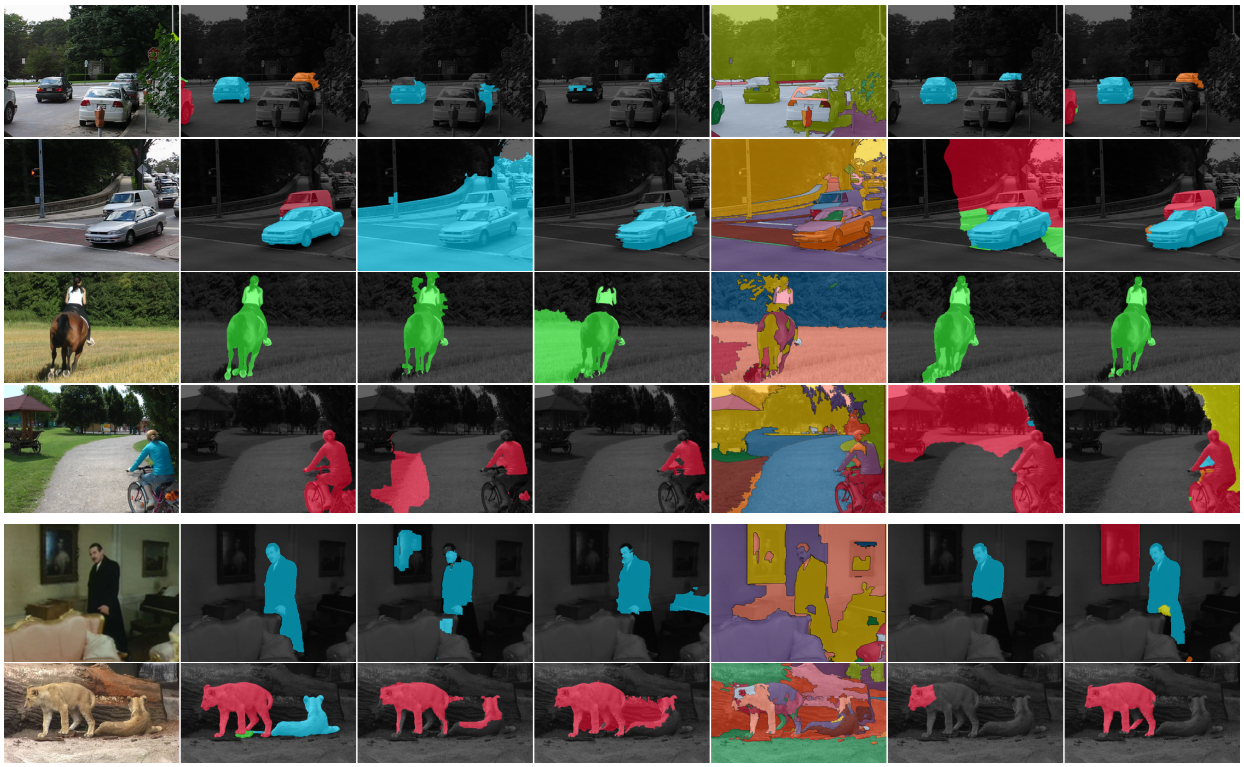


Figure 2.13: *Sample results on MoSeg.* Left to right: *original image, ground truth, [LKG11], [PF13], [GKH10], [OMB14], ours (object maps). Top two rows: occlusion cues allow us to obtain even the barely visible cars (row 1 - orange, row 2 - red, row 2 - green). Row 3: use of both motion and appearance cues allows us to generate an accurate object boundary. Row 4: occlusion cues yield three depth layers (bicyclist, tree, background) (see also Fig. 2.1). Notice that the tree (and some cars in rows 1–2) is not annotated, so our scheme is penalized despite providing the correct answer. [LKG11, PF13] suffer from trailing and only produce binary segmentation. [GKH10] suffers from oversegmentation. [OMB14] performs comparably to our method; The last two rows show failure cases. Row 5: the painting is recognized as an "object" due to false occlusion detection; the hand is assigned to a separate layer. Row 6: the lioness is missed due to insufficient motion and lack of occlusions.*

CHAPTER 3

Active Frame, Location, and Detector Selection for Automated and Manual Video Annotation

We describe an information-driven active selection approach to determine which detectors *to deploy at* which location *in* which frame *of a video to minimize semantic class label uncertainty at every pixel, with the smallest computational cost that ensures a given uncertainty bound. We show minimal performance reduction compared to a "paragon" algorithm running all detectors at all locations in all frames, at a small fraction of the computational cost. Our method can handle uncertainty in the labeling mechanism, so it can handle both "oracles" (manual annotation) or noisy detectors (automated annotation).*

3.1 Introduction

Semantic video segmentation refers to the annotation of each pixel of each frame in a video with a class label. If we are given a *data collection mechanism,* either as an "oracle" or a detector for each known object class, we could perform semantic video segmentation in a brute-force (and simplistic) way by labeling each pixel in each frame. Such a "baseline" algorithm is clearly inefficient as it fails to exploit spatio-temporal regularities in the video signal. Moreover, capturing and exploiting these regularities is computationally inexpensive and can be done using a variety of low-level vision techniques. On the other hand, detecting and localizing objects in the scene requires high-level semantic procedures that have far greater computational cost (in the manual annotation scenario, semantic procedures are replaced with an expensive human annotator). In other words, the complexity of annotating a video sequence is dominated by the cost of high-level procedures, *e.g.* submitting images to a battery of detectors. The annotation cost decreases if fewer

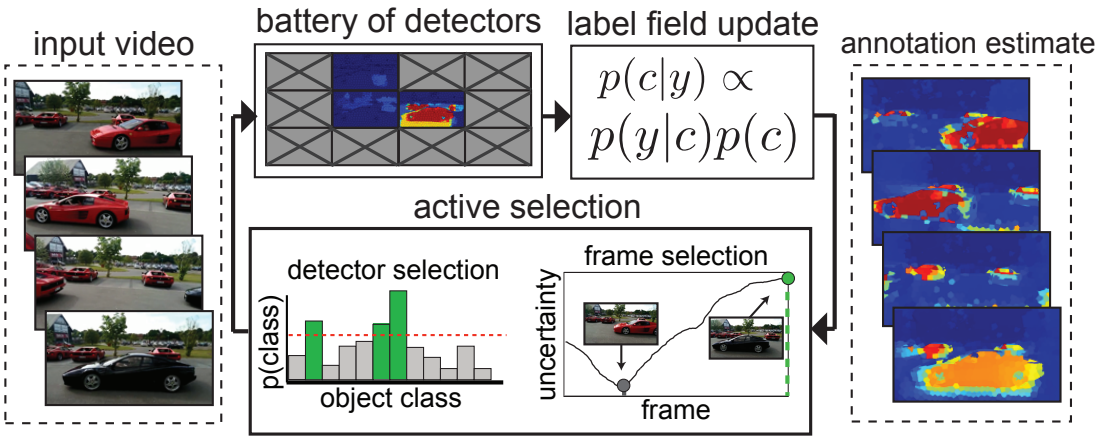


Figure 3.1: *Given an input video, our approach iteratively improves the annotation estimate by submitting "informative" frames to a "relevant" subset of object detectors. At each iteration, we select the most informative frame (and possibly region within it) based on uncertainty of current annotations. We then select a subset of relevant object detectors, based on estimate of classes that are present in the video. Responses of these detectors are used to update the posterior of the label field, which is then used to perform selection at the next iteration. Project page:* `http://vision.ucla.edu/activeselection/`

such procedures are performed.

We describe a method to reduce the complexity of a labeling scheme, using either an oracle or a battery of detectors, by exploiting temporal consistency and actively selecting *which* data to gather (which detector), *when* (which frame) and *where* (location in an image).

It is important to stress that our method aims to *reduce complexity*, but in principle *can do no better than the baseline*, since it is using only a subset of the data. To avoid confusion, we call the performance upper bound *paragon*, rather than baseline. If the data collection mechanism is reliable (*e.g.* an oracle), we show minimal performance reduction at a fraction of the cost.

Our approach is framed as *uncertainty reduction* with respect to the choice of frame, detector, and location. As a result, we can work with uncertain data collection mechanisms, unlike many label propagation schemes that assume an oracle [VG12]. As output, we provide a class-label probability distribution per pixel, which can be used to estimate the most likely class, and also provides the labeling uncertainty.

Our method hinges on the *causal* decision of what *future* data to gather, when, and where, based on inference from past data, so as to reduce labeling uncertainty (or "information gain" [Lin56]).

The method is formulated as a stochastic subset selection problem, finding an optimal solution to which is intractable in general. However, the problem enjoys the *submodular* property [NWF78], so a greedy heuristic attains a constant factor approximation of the optimum [KG05], a fact that we exploit in our method. A brief overview of our framework is shown in Fig. 3.1.

3.1.1 Related work and contributions

We are motivated by the aim to perform semantic video segmentation using as few resources (frames and detectors) as possible, while guaranteeing an upper bound on residual uncertainty. We do so by sequentially choosing the best measurements, which relates to *active learning*. Searching for the best region within the image relates to *location selection*. Detector selection is performed by leveraging object co-occurrences in the video; thus, work on *contextual information* is relevant.

Active learning aims to minimize complexity by selecting the data that would provide the largest "value" relative to the task at hand. It has been used in image segmentation [VBF12] and user-guided object detection [YGL12]; tradeoffs between cost and informativeness were studied in [VG09], and a procedure efficiently computing the objective was described in [MCK14]. The active learning framework has been used for keyframe selection in an interactive video annotation task in [VR11]. In a work closest to ours, [VG12] addresses frame selection for label propagation in video. However, their method relies on oracle labeling and moreover cannot be easily extended to location selection.

Location selection has been studied to reduce the number of evaluations of a sliding-window detector. Previously this was done by generating a set of diverse image partitions likely to cover an object [ADF10]. In [SUG11], it was shown that using such segmentation-driven approach on large image databases maintains performance, while providing computational savings. In [AHT12] a class-dependent sequential decision ("where to look next") approach exploits "context" learned in training, but is limited to finding a single object per image. Recently, a sequential decision strategy that requires user interaction, but does not have this limitation was described in [CSM14]. Our approach is not limited to a single object, is class-independent, and is based on direct uncertainty-minimization framework.

Contextual information has been used to prune false positives in *single images* [CLT10] by using co-occurrence statistics and learning dependencies among object categories. Similarly, [YFU12] use a conditional random field to infer "category presence" using co-occurrence statistics in single images. Our work is related to [KBF12], who exploit co-occurrences to sequentially choose which *detectors* to run to improve speed of object recognition in single images. On the other hand, we tackle video, which allows us to obtain significant computational savings by not running "unnecessary" detectors. In this chapter, we focus only on labeling "objects" in video, as found by bounding box detectors. Extending the approach to also use region-based detectors, as in [TL13], is possible, and could allow for using geometric context [HK08].

Our *first contribution* is a frame selection method for video annotation, which naturally allows for uncertainty in the measurements, and thus is applicable to both a battery of standard object detectors (yielding automated annotation), as well as an error-free oracle (yielding manual annotation). Region selection within an image is then naturally added to the framework – this is our *second contribution*. Our *third contribution* is the extension of the framework to enable not just the frame and location selection, but also the selection of detectors based on video shot context.

3.2 Formulation

In Sec. 3.2.1 we give an overview of our probability model, introduce object detectors, and describe how their outputs are used to infer the underlying object class labels. Sec. 3.2.2 introduces the information gathering framework, and proposes an online strategy to select the most informative frames on which to run detectors. This strategy is extended to selecting a region within an image in Sec. 3.2.3. Sec. 3.2.4 describes a method for selecting the best subset of detectors by inferring and exploiting context in the video.

Let $I(t) : D \to \mathbb{Z}_+$ be the image defined on a domain $D \subset \mathbb{R}^2$, and let $\{I(t)\}_{t=1}^F$ be F frames of a video. The set of temporally consistent regions (*e.g.* supervoxels or other oversegmentation of the video) $\{S_i\}_{i=1}^N$ forms a partition of the video domain $D^F = \cup_{i=1}^N S_i$ with $S_i \cap S_j = \varnothing$ (see Fig. 3.2). We assume that these regions respect object boundaries, so that it is possible to associate (temporally-consistent) object class labels to them. For the i-th region, we denote such label by

Figure 3.2: *A pair of frames with temporally consistent regions ($\{S_i\}$, [CWI13]). The highlighted regions are present in both frames.*

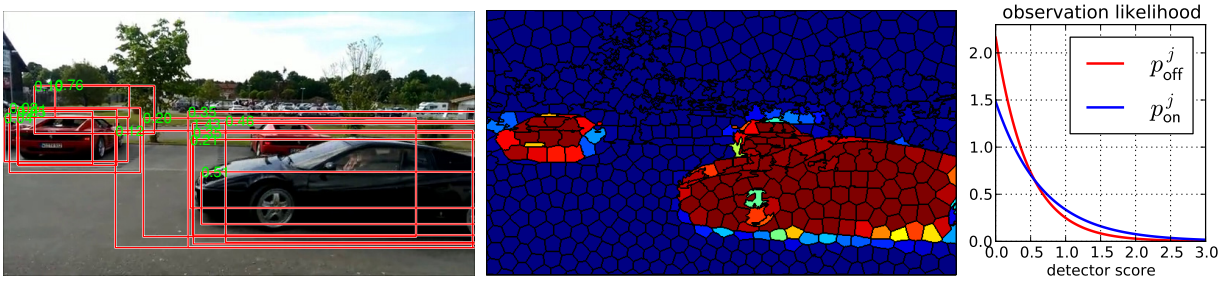


Figure 3.3: *A pseudo-measurement provided by the car detector.* Left: *a set of bounding boxes given by a DPM detector [FGM10].* Middle: *segmentation result using Grabcut [RKB04]. Color indicates detector score. This is taken as the measurement in our framework.* Right: *likelihood* (3.1) *for a "car" class.*

$c_i \in \{0, \dots, L\}$.

A bank of object detectors represents a set of "test functions" that, when executed at time t, provide a measurement $y(t) : D \to \mathbb{R}^L_+$. We are interested in labels assigned to regions, so we will write $y_i(t) \in \mathbb{R}^L_+$ to denote responses of a detector bank supported on subset of S_i in t-th frame. Its j-th component $y^j_i(t) \in \mathbb{R}_+$ is the "detection score" for object class $j \in \{1, \dots, L\}$[1] (see Fig. 3.3) that provides uncertain evidence for the underlying labels.

3.2.1 Probability model

To measure the "value" of future measurements y for the task of inferring labels $\{c_i\}_{i=1}^N$, we need to quantify their uncertainty before actually measuring them, which requires a probabilistic detector response model. We assume object labels to be spatially independent: $p(c_1, \dots, c_N) = \prod_{i=1}^N p(c_i)$, with a uniform prior $p(c_i) = \frac{1}{L+1}$. This assumption is introduced for simplicity, and can be lifted

[1] We have $L + 1$ object classes and L detectors, since there is no standard "background detector".

by using the Markov random field (MRF) model.

Detector response model. When a bank of detectors is deployed on t-th frame, we obtain evidence for the object labels of regions supported in that frame. For i-th region, this evidence is modeled by a likelihood $p(y(t)|c_i) = p(y_i(t)|c_i)$ (responses whose domain does not intersect the i-th region are not informative). Moreover, we assume that individual detector responses are conditionally independent given the label: $p(y_i(t)|c_i) = \prod_{j=1}^{L} p(y_i^j(t)|c_i)$. To learn these distributions, we use VOC 2010 database. Namely, for each detector, we learn the "true positive" $p_{j,\mathsf{on}}(y^j) \doteq p\big(y^j|c=j\big)$ and the "false positive" $p_{j,\mathsf{off}}(y^j) \doteq p\big(y^j|c\neq j\big)$ distributions, which we model as exponentials:

$$\begin{cases} p_{j,\mathsf{on}}(y^j) & = \lambda_{j,\mathsf{on}} \exp(-y^j \lambda_{j,\mathsf{on}}) \\ p_{j,\mathsf{off}}(y^j) & = \lambda_{j,\mathsf{off}} \exp(-y^j \lambda_{j,\mathsf{off}}) \end{cases} \tag{3.1}$$

As shown in Fig. 3.3, $p_{j,\mathsf{off}}$ decays faster than $p_{j,\mathsf{on}}$ – false positive responses usually have smaller scores than true positive responses. Using these distributions, for a background class ($c_i = 0$), we have $p(y_i(t)|c_i = 0) = \prod_{j=1}^{L} p_{j,\mathsf{off}}(y_i^j(t))$, while for an object class ($k \geq 1$), we have $p(y_i(t)|c_i = k) = p_{k,\mathsf{on}}(y_i^k(t)) \prod_{\substack{j=1 \\ j\neq k}}^{L} p_{j,\mathsf{off}}(y_i^j(t))$.

Oracle response model. An error-free oracle can be viewed as an ideal detector that returns "1" if an object of a particular class is present at region i and "0" otherwise. In this case, the (binary) measurements $y_i^j(t) \in \{0,1\}$ are deterministic functions of the underlying labels, and can be written using Kronecker deltas as: $y_i^j(t) = \delta(j - c_i)$, with the likelihood:

$$p(y_i^j(t)|c_i) = \begin{cases} 1-\epsilon & \text{if } y_i^j = 1 \\ \epsilon & \text{if } y_i^j = 0 \end{cases}. \tag{3.2}$$

The regularization by ϵ (a small number) is needed to avoid singularities due to violations of modeling assumptions (either oracle errors, or failure of labels' temporal consistency). Because our goal is automated annotation, we will refer to detector responses throughout the chapter; however, the methods can be directly applied to oracle annotation as well.

Label field update. Given detector responses in frame t, the label probability in region i is updated as

$$p\big(c_i\big|y(t)\big) \propto p\big(y_i(t)\big|c_i\big)p\big(c_i\big). \tag{3.3}$$

This can be extended to a recursive update: if $\mathcal{Y}^k = \big\{y(t_1), \ldots, y(t_k)\big\}$ is the history of detector responses taken at frames $\{t_j\}_{j=1}^k$, then

$$p\big(c_i\big|\mathcal{Y}^k\big) \propto p\big(y_i(t_k)\big|c_i\big)p\big(c_i\big|\mathcal{Y}^{k-1}\big), \tag{3.4}$$

which is standard in recursive Bayesian filtering [Jaz70]. When $k = 1$ (the first update) it simply restates (3.3) (Bayes' rule).

3.2.2 Information gathering; active frame selection

Having described the update of label probabilities after measuring detector responses, we return to the question of selecting the best subset of frames where to run them. Our *goal* is to maximize the "Value of Information", (VoI [HHM93, How66]), or *uncertainty reduction*, with respect to the *choice* of frames to submit to the oracle, or to test against a battery of detectors. Thus, given frames $\{I(t)\}_{t=1}^F$, regions $\{S_i\}_{i=1}^N$, and a budget of K frames, we minimize uncertainty on labels $\{c_i\}_{i=1}^N$ with respect to a selection of K measurements (yet) to be taken

$$t_1^*, \ldots, t_K^* = \underset{\mathcal{T}:|\mathcal{T}|\leq K}{\operatorname{argmin}} H\big(c_1, \ldots, c_N\big|y(t_1), \ldots, y(t_K)\big). \tag{3.5}$$

We measure uncertainty by (Shannon) entropy [CT91].

The problem (3.5) is in general intractable. For a very special case of noise-free measurements, a dynamic programming (DP) solution exists [KG09, Alg.1], as was also shown in [VG12]. When measurements are noisy, this solution is not guaranteed to be optimal. However, due to conditional assumptions made in Sec. 3.2.1, the problem is *submodular*, so a greedy decision policy yields a constant factor approximation to the optimum [KG05] (see also Sec. 3.5 for proof). Thus we settle

for

$$s^* = \underset{s}{\operatorname{argmin}}\, H\big(c_1, \ldots, c_N \big| y(s), \mathcal{Y}^k\big), \tag{3.6}$$

where $\mathcal{Y}^k = \{y(t_1), \ldots, y(t_k)\}$ is the set of *already observed* detector responses, and s is the frame index for the next measurement *yet to be taken*. This policy begins with $\mathcal{T} = \varnothing$, at each stage chooses the frame s^* that provides the greatest uncertainty reduction, updates the set of chosen frames $\mathcal{T} := \mathcal{T} \cup \{s^*\}$, and repeats. Notice that this policy has two advantages over the DP solution: first, it is *online* and thus does not require a pre-defined budget (K). Second, it is less susceptible to modeling errors because it uses *observed* responses in making the next decision.

Using the properties of conditional entropy we can write $H(c_1, \ldots, c_N | y(s), \mathcal{Y}^k) = H(c_1, \ldots, c_N | \mathcal{Y}^k) - \mathbb{I}(y(s); c_1, \ldots, c_N | \mathcal{Y}^k)$. Since the first term (uncertainty of c_i's *before* the next selection) is independent of $y(s)$, (3.6) is equivalent to maximizing the second term, which is the mutual information (MI) between the *next* measurement and the labels: $\mathbb{I}(y(s); c_1, \ldots, c_N | \mathcal{Y}^k)$.

Due to the spatial independence of labels, we have that $\mathbb{I}(y(s); c_1, \ldots, c_N | \mathcal{Y}^k) = \sum_{i=1}^{N} \mathbb{I}(y(s); c_i | \mathcal{Y}^k)$. Thus, we can rewrite (3.6) as

$$s^* = \arg\max_{s} \sum_{i=1}^{N} \mathbb{I}(y(s); c_i | \mathcal{Y}^k) \tag{3.7}$$

Moreover, if c_i is not present in frame s, then $y(s)$ does not provide evidence for it, and $\mathbb{I}(y(s); c_i | \mathcal{Y}^k) = 0$. On the other hand, if c_i is present, $y(s)$ is informative – proportionally to uncertainty in c_i. Thus, the criterion prefers frames that have the largest number of uncertain regions. As in the twenty-question game, we wish to label the data that is *most uncertain* given prior measurements. Taking measurements on frames that have little uncertainty provides little information gain.

There are no closed form expression for mutual information for the densities that we consider here. Hence, we need to either approximate it by Monte Carlo sampling, or to find efficiently computable proxies. Because we are interested in a maximization problem, the natural proxy of interest is the lower bound on $\mathbb{I}(y(s); c_i | \mathcal{Y}^k)$. However, it is also very common to use upper bounds (most often using the second-moment Gaussian approximation). Upper bounds are acceptable because we are ultimately interested in the maximizing *point*, rather than the maximizing *value*.

Thus, the tightness of the bound is irrelevant, and it is only required that the maxima are preserved. We use an upper bound:

$$\mathbb{I}(y(s); c_i|\mathcal{Y}^k) \leq \sum_{m=0,n=0}^{L,L} w_m w_n \sum_{j=1}^{L} \frac{\eta_{jn}}{\eta_{jm}} - L \tag{3.8}$$

where $w_m \doteq p(c_i = m|\mathcal{Y}^k)$, and $\eta_{jm} = \lambda_{j,\mathsf{on}}$ if $j = m$ and $\eta_{jm} = \lambda_{j,\mathsf{off}}$ otherwise. We prove this result in Sec. 3.5 and empirically show that it preserves the local maxima of the Monte Carlo approximation.

As an alternative to (3.7) we can try to select frames with high classification error. Let $\mathcal{E}(s,i) = \mathbf{alive}(S_i, s)\big(1 - \max_\ell p(c_i = \ell|\mathcal{Y}^k)\big)$ where $\mathbf{alive}(S_i, s) = 1$ if region S_i is supported on frame s and is 0 else, and the second term is simply the classification error. We then use a criterion

$$s^* = \arg\max_s \sum_{i=1}^{N} \mathcal{E}(s,i). \tag{3.9}$$

Note that unlike MI, this criterion does not make predictions about the next measurement $y(s)$, and is therefore very simple to compute. Yet, as we show in Sec. 5.3, it performs competitively in practice.

3.2.3 Active location selection

It is straightforward to augment the frame selection criterion (3.7) with the selection of the most informative region R ($R \subseteq D$) within the image (to either run detectors on, or to query an oracle). In this case, at each stage we maximize $\sum_{i=1}^{N} \mathbf{1}_{\{S_i \subset R\}} \mathbb{I}(c_i; y(s)|\mathcal{Y}^k)$ over a pair (s, R), where the indicator $\mathbf{1}_{\{S_i \subset R\}}$ discards all the labels c_i that are outside R (the region that is submitted to detectors). However, because the mutual information is nonnegative, the best region chosen by this strategy will always contain the entire image ($R^* = D$), and a proper subset of the image domain will never be chosen. Thus, it is necessary to associate a *cost* to R. We choose a cost that

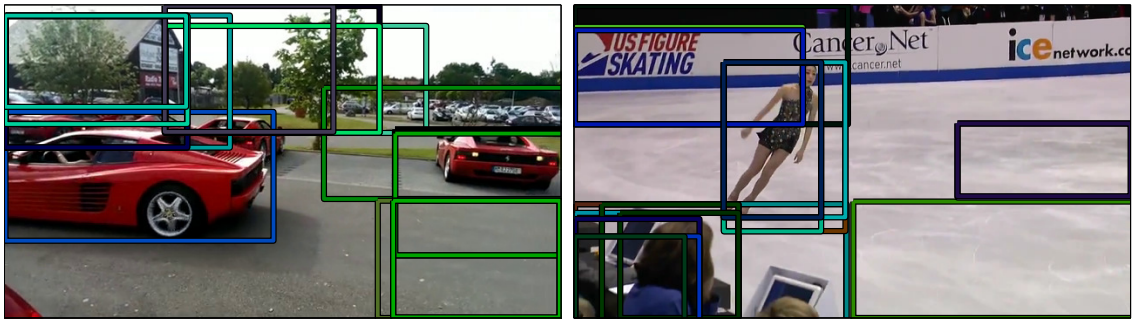


Figure 3.4: *Candidate regions used for selecting the most informative location* (3.10)

is proportional to the region size, and trade off the two as:

$$s^*, R^* = \arg\max_{s,R} \sum_{i=1}^{N} \mathbf{1}_{\{S_i \subset R\}} \mathbb{I}(c_i; y(s)|\mathcal{Y}^k) - \gamma|R| \tag{3.10}$$

with γ – a weighing term. A more sophisticated approach could estimate and use the computational effort in running a detector bank as a function of $|R|$ (which needs not be linear). In practice, we maximize the criterion over a finite, diverse set of candidate regions (shown in Fig. 3.4), which presumably cover objects in the image.

3.2.4 Context and active detector selection

In situations where L (the number of available detectors) is large, but the number of classes present in the scene is small, it is not computationally efficient to run the entire battery of detectors. To address this issue, we extend the framework to include not just a selection of subsets of frames and regions to be labeled, but also a selection of the subset of detectors to be deployed, by exploiting context in the video.

Probability model. To describe the *context* of the video sequence, we introduce a random variable $o = (o_1, \ldots, o_L) \in \{0, 1\}^L$ that is global within a video shot, where o_j represents presence or absence of k-th object category in the shot. Detector responses provide soft evidence as to whether object categories are present in the shot. Our belief in objects being present is summarized by the distribution $p(o|\mathcal{Y}^k)$ – the posterior of the context variable, given evidence from the detectors.

To infer this distribution, we must first specify the likelihood $p(y|o)$. We assume that the

distribution can be factorized as $p(y|o) = \prod_{j=1}^{L} p(y^j|o_j)$, where each term is a model for a detector response given that an object j is present (or absent). To be invariant to response location, we use the maximum detector response score within an image $z^j(t) = \max_i y_i^j(t)$ as the observation associated with j-th category presence, and specify the model as:

$$\begin{cases} p(y^j(t)|o_j = 0) & = p_{j,\mathsf{off}}\big(z^j(t)\big) \\ p(y^j(t)|o_j = 1) & = \pi p_{j,\mathsf{on}}\big(z^j(t)\big) + (1-\pi) p_{j,\mathsf{off}}\big(z^j(t)\big). \end{cases} \tag{3.11}$$

where $p_{j,\mathsf{off}}$, $p_{j,\mathsf{on}}$ are the distributions used in (3.1). The density $p(y^j(t)|o_j = 1)$ is a mixture, and can account for the possibility of an object *not* being present in frame t despite being present in the video shot. The mixture parameter π is related to the fraction of time the object is expected to be visible in the shot.

Detector selection. The marginal distributions $p(o_j|\mathcal{Y}^k)$ describe the probability that j-th class is present in the video, given the observation history. If computation is limited, when $p(o_j|\mathcal{Y}^k)$ is small, we should avoid running j-th detector. This can be phrased in terms of a threshold α on marginal probabilities, yielding a two-stage procedure:

$$\begin{cases} J & = \{j : p(o_j|\mathcal{Y}^k) > \alpha\} \\ s^* & = \arg\max_s \sum_{i=1}^{N} \mathbb{I}(\{y^j(s)\}_{j\in J}; c_i|\mathcal{Y}^k) \end{cases} \tag{3.12}$$

where $\{y^j(s)\}_{j\in J}$ is a set of responses for detectors indexed by J. This procedure is performed at each stage of our sequential decision problem: once the set J of object categories is chosen, we select the most informative frame s^* to run these detectors on, acquire new evidence, update the posteriors, and repeat.

Of course, the frame selection step in the detector selection procedure (3.12) can be extended to allow for region selection. Due to space constraints, we do not explicitly write out the equation; however, it is no different from the extension of the original frame selection criterion (3.7) to *frame and region* selection (3.10).

Figure 3.5: *Sample frames from a subset of video sequences used for testing our algorithm, from BVSD ([SBM11b],top) and Flickr (ours, bottom).*

3.3 Experiments

To evaluate our approach, we use public benchmark datasets including human-assisted motion[LFA08], MOSEG[BM10], Visor[VC10], and BVSD[SBM11b], as well several videos from Flickr (Fig. 3.5). We are interested in classification error, so we manually created pixelwise ground truth by labeling these sequences.

Implementation Details. We use [CWI13] to compute video-superpixels (temporally consistent regions), with 500-700 superpixels in an image. Our detectors contain models for 20 classes, pre-trained on VOC 2010 [EVW], based on DPM [FGM10], and refined with GrabCut [RKB04], as shown in Fig. 3.3. We offset detector scores to make them nonnegative and convert them into likelihoods for use in our model (3.1), with likelihood distribution learned from the VOC 2010. To approximate $p(o|\mathcal{Y}^k)$, we use a fully connected MRF, with node and edge potentials learned using the UGM toolbox [Sch]. The co-occurrence statistics are derived from VOC. Throughout all experiments we used $\alpha = 0.005$ (detector selection threshold (3.12)), $\pi = 0.5$ (mixture weight in (3.11)). As written in (3.7) and (3.10), the selection criteria weigh terms corresponding to different regions equally. In practice we weighted the terms according to region size; this choice improved performance.

Videos in our database vary in duration (from 19 to 300 frames), so we report classification accuracy as a function of percentage of sequence labeled; this can be either a percentage of *frames* submitted to detectors, or a percentage of *pixels* in a video submitted to detectors.

Frame selection. Our first experiment compares our information-seeking approach (with criteria

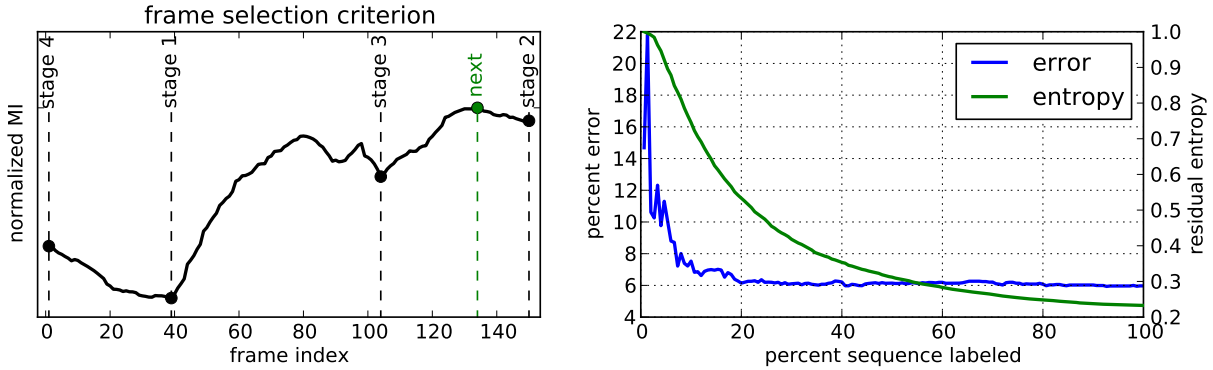


Figure 3.6: Left: *frame selection criterion for "*Ferrari*" sequence (marked with "*" in Fig. 3.5), for the first few selection stages. The selected frames are not uniformly distributed.* Right: *classification error and normalized residual entropy ($\sum_{i=1}^{N} H(c_i|\mathcal{Y}^k)$) as a function of number of frames selected for labeling. Note that after 20% sequence is labeled, error reaches an asymptote.*

given by (3.8) and (3.9)) with the naive methods (uniform and random selection). We also compare with the DP approach of [VG12]: we use their selection criterion, but propagate labels using our model, to make the comparison with the other methods fair. As can be seen in Table 3.1(top), we consistenly outperform other methods when the error-free "oracle" is used. The improvement over [VG12] is due to our objective being closely related to the labeling error and birth/death of temporal regions, whereas their selection involves a combination of optical flow, image intensities, and backward-forward optical flow consistency (a proxy for occlusions).

When the "oracle" is replaced by a battery of detectors, the DP approach is not optimal. Moreover, due to erroneous detections (false positives or misses), *any* of the methods is susceptible to failure, even if they choose "informative" frames. We observe this in Table 3.1 (bottom): although our methods perform best on average, at 10% labeling, "uniform" attains a lower classification error. When the detector performance is poor, any sampling scheme yields equally bad or worse performance.

Location selection. Our second experiment compares our region and frame selection scheme against the random selection. The approach in [VG12] cannot be easily extended to region selection, so we do not compare against it. Our candidate regions vary in size and location, and uniformly selecting representatives out of this set is rather problematic; therefore we do not test against this approach.

	Oracle labeling					Detector labeling				
%	$\mathcal{E}$	MI	[VG12]	uniform	random	$\mathcal{E}$	MI	[VG12]	uniform	random
1	**4.666**	**4.666**	5.112	6.079	5.30 $\pm$ 0.29	**11.224**	11.277	12.623	11.948	12.62 $\pm$ 1.0
5	**2.696**	2.700	3.130	3.270	3.56 $\pm$ 0.19	**9.069**	9.129	10.446	9.363	9.36 $\pm$ 0.43
10	**2.264**	2.274	2.434	2.478	2.91 $\pm$ 0.14	8.097	8.393	8.455	**7.764**	8.64 $\pm$ 0.44
15	**2.088**	2.092	2.110	2.194	2.59 $\pm$ 0.10	7.862	**7.674**	8.389	7.844	8.68 $\pm$ 0.37
20	**2.005**	2.007	2.023	2.072	2.46 $\pm$ 0.06	7.611	**7.582**	8.041	7.589	8.50 $\pm$ 0.26
30	1.935	**1.930**	1.961	1.979	2.36 $\pm$ 0.07	7.449	**7.377**	8.105	7.890	8.22 $\pm$ 0.25

Table 3.1: *Average classification error of different frame selection strategies with oracle (left) and detector (right) labeling; 1%-30% frames. Our proposed approaches are "$\mathcal{E}$" and "MI".*

We compute candidate regions using [SUG11], which typically consist of bounding boxes that entirely contain objects of interest (see Fig. 3.4). Typically, per image, we have 10-20 regions that occupy 10%-80% of the image. Results with oracle and detector labeling are shown in Table 3.2, as a function of percent pixels used to obtain a labeling (proportional to the sum of selected regions' areas). Perhaps unsurprisingly, the "random" selection performs poorly.

	Oracle labeling			Detector labeling		
%	$\mathcal{E}$ (our)	MI (our)	random	$\mathcal{E}$ (our)	MI (our)	random
1	6.503	**6.189**	9.177 $\pm$ 0.502	16.878	**16.861**	17.113 $\pm$ 1.225
5	3.434	**3.125**	4.662 $\pm$ 0.374	**15.266**	15.923	16.061 $\pm$ 0.458
10	2.715	**2.456**	3.273 $\pm$ 0.155	14.588	**13.830**	15.192 $\pm$ 0.716
20	2.392	**2.187**	2.600 $\pm$ 0.067	12.987	**12.372**	13.935 $\pm$ 0.304
30	2.289	**2.144**	2.329 $\pm$ 0.061	11.970	**11.819**	12.963 $\pm$ 0.182

Table 3.2: *Average classification percent error of different region+frame selection strategies with oracle (top) and detector (bottom) labeling; using 1%-30% pixels.*

Detector selection. This experiment demonstrates the possibility of reducing computation effort without suffering a performance penalty, by reducing the number of detectors deployed at each stage. In these experiments we perform frame selection using the MI criterion (3.8) (although $\mathcal{E}$ can be used as well). We do not perform region selection. The typical behavior of the detector selection, as shown in Fig. 3.7, is to run fewer detectors as more and more frames are selected. Often, in the limit, only the detectors for the classes that are present are fired. Thus, the cost of measurement decreases (and computational savings increase) with number of labeled frames.

One may wonder how much is gained from using co-occurrence information. To investigate this, we performed a set of experiments under the independence assumption $p(o|\mathcal{Y}^k) =$

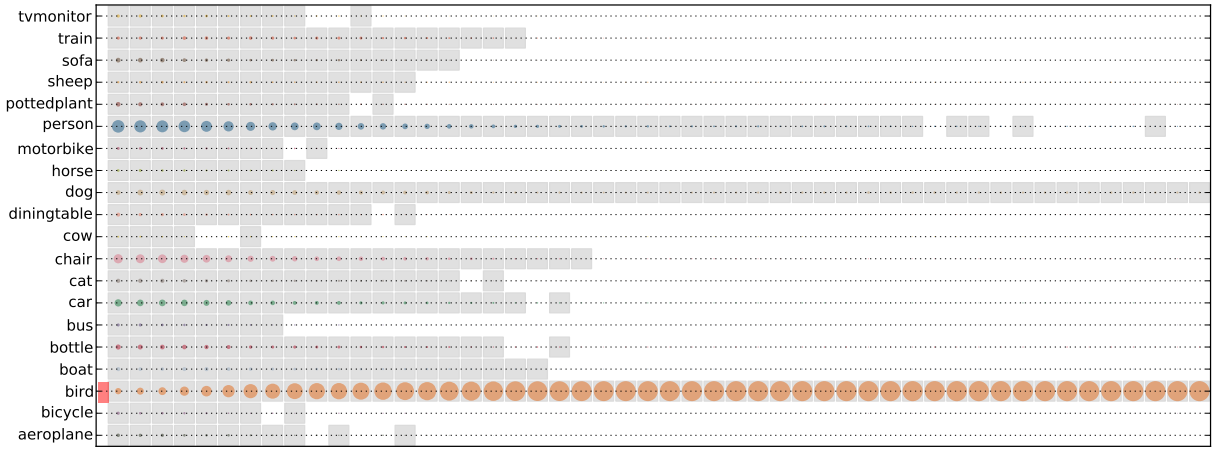


Figure 3.7: *Detector selection on "*Planet Earth*" sequence (marked with "#" in Fig. 3.5): For each frame selected for labeling (abscissa), deployed detectors are shown as gray boxes. Colored circles represent $p(o_j|\mathcal{Y}^k)$ – the belief of a particular class being present, with the area proportional to the value of the posterior, and ground truth is indicated on the left by a red mark (bird). The "dog" class is fired the longest because it co-occurs with "bird" in the training set.*

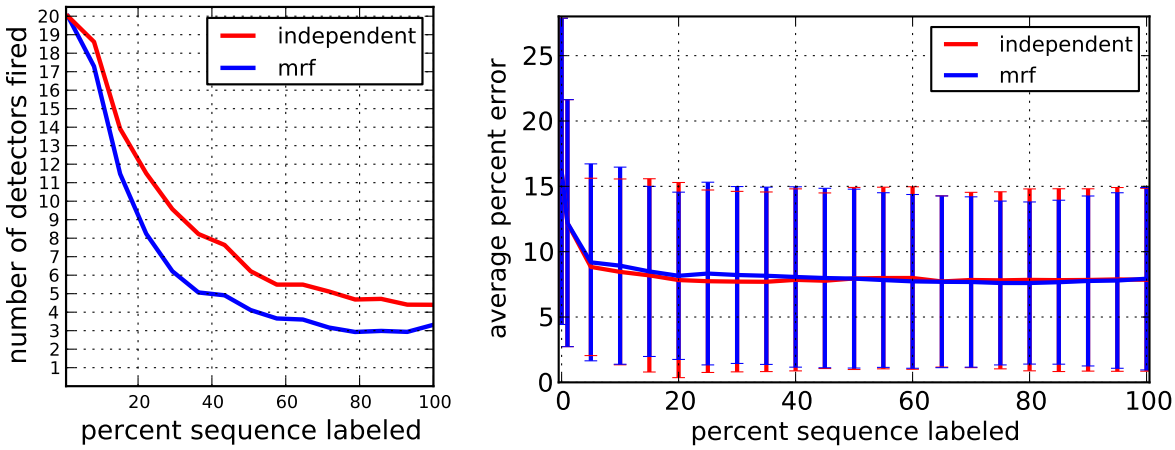


Figure 3.8: Left: *As more frames get labeled, fewer detectors are fired. We show the average number of detectors fired, over all sequences in our datasets, for "MRF" and "independent" approximations. MRF approximation of the joint distribution makes it possible to quickly stop firing contextually irrelevant detectors.* Right: *classification error as function of labeled frames. The slightly larger error in "MRF" is the price paid for reduced number of used detectors.*

$\prod_{j=1}^{L} p(o_j|\mathcal{Y}^k)$. The average computation savings and the average classification errors are shown in Fig. 3.8. Using an MRF, we get a substantial decrease in the number of detectors that are fired at each stage. This is because co-occurrence information allows us to quickly suppress probabilities for contextually atypical situations.

Baseline and "paragon" annotation. We first illustrate the gain from using temporal infor-

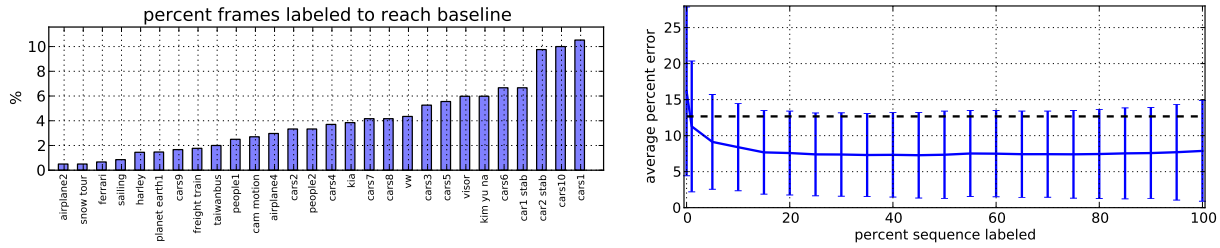
quantity	method	cost
$y(t)$	[FGM10]+[RKB04]	$60s + 180s$
$y(t)\mathbf{1}_R$	[FGM10]+[RKB04]	$\frac{\lvert R\rvert}{\lvert D\rvert}(60s + 180s)$
$\{y^j(t)\}_{j\in J}$	[FGM10]+[RKB04]	$\frac{\lvert J\rvert}{L}(60s + 180s)$
$\{S_i\}_{i=1}^N$	[ARS12]+[CWI13]	$F(10s + 5s)$
candidate regions	[SUG11]	$F(0.5s)$
frame selection	(3.7) + (3.4)	$0.02s$
frame+region selection	(3.10) + (3.4)	$0.045s$
detector selection	(3.12)	$5s$

Table 3.3: *A summary of quantities that are computed in our framework, associated procedures, and costs, measured in terms of time.*

mation. We compare our approach with the one that does not use temporal consistent regions. Specifically, our "probabilistic baseline" (PB) is the maximum likelihood estimation using the model (3.1), applied to the detected and subsequently segmented regions. It considers detector responses from *all* frames, but treats each frame independently. Our framework outperforms this approach; in fact, we perform better after labeling only a small fraction of the sequence. Fig. 3.9a shows the percentage of frames needed to reach the performance of this baseline: we perform better after labeling only 10% of the sequence. Fig. 3.10 shows several examples of annotation using PB and our approach: temporal regularities allow us to suppress a large number of false positive detections.

We also compare against the "paragon" approach, which uses *all* frames, *all* detectors, and temporal consistency. But by running detectors on only a fraction of the frames, we do not perform significantly worse. As shown in Fig. 3.9b, classification error has a "diminishing returns" property: as more frames are labeled, the improvement is decreasing. This suggests that using *all* frames is unnecessary, and if one has computational constraints, "early stopping" can be beneficial.

Computation savings. To produce a PB annotation, one runs detectors and segmentation on every frame. DPM [FGM10] takes 60s/frame (for 20 classes) and Grabcut[RKB04] takes 180s/frame (we segment every bounding box using an unoptimized MATLAB implementation); the sum of the two is the "cost" of observing $y(t)$. To leverage temporal consistency, we use [CWI13] (5s/frame) and optical flow [ARS12] (10s/frame). The costs of our *frame selection* framework are negligible: computation of frame selection utility (3.7) (or (3.9)) takes 15 ms/stage on all frames, inference



(a) (b)

Figure 3.9: *(a) Percentage and number of frames to be labeled by detectors to match PB performance. On average across all sequences we only need to label* 4.012% *of the frames. Baseline obtains* 12.669% *average error using detectors on all frames. (b) Blue: average error over entire dataset as a function of labeled frames. As more frames are labeled, the improvement in error decreases (on average), suggesting that if computational budget is limited, it is unnecessary to use all frames. Black dashed line is PB (which uses all frames independently).*

(3.4) takes 5 ms/stage, both measured on the longest sequence. Region selection requires the candidate regions [SUG11], which cost 0.5s/frame, but computation of location selection utility (3.10) remains negligible (40 ms/stage). Our *detector selection* framework requires estimating "presence" marginals $p(o_j|\mathcal{Y}^k)$ at a cost of 5s per stage of the algorithm. These "costs" are in Table 3.3.

We can estimate the PB cost as $F(240s)$, where F is the number of frames in the sequence. The "paragon" requires temporally consistent regions $\{S_i\}_{i=1}^N$, and thus costs $F(270s)$. The frame selection framework requires negligible computation per stage and reduces computation cost to $F(15s) + K(240s)$, where K is the budget of frames to be labeled (according to Fig. 3.9b, $K \approx 0.2F$ is sufficient). The region selection framework decreases the observation cost linearly in region size; an admittedly coarse assumption. The cost of a measurement supported on region R, denoted $y(t)\mathbf{1}_R$, is then reduced from $240s$ to $|R|/|D|(240s)$. The detector selection framework decreases the observation cost to $|J|/L(240s)$, but incurs an additional $5s$ per stage (due to context inference). As a specific example, for a "Ferrari" sequence with $F = 150$, PB costs 600 min. The "paragon" costs 638 min. Our framework with frame selection and 20% labeling needs only $\sim$158 min. Frame and region selection costs the same amount. Using frame and detector selection framework, we use 54% detectors in the first 20% frames, reducing the cost to just $\sim$85 min.

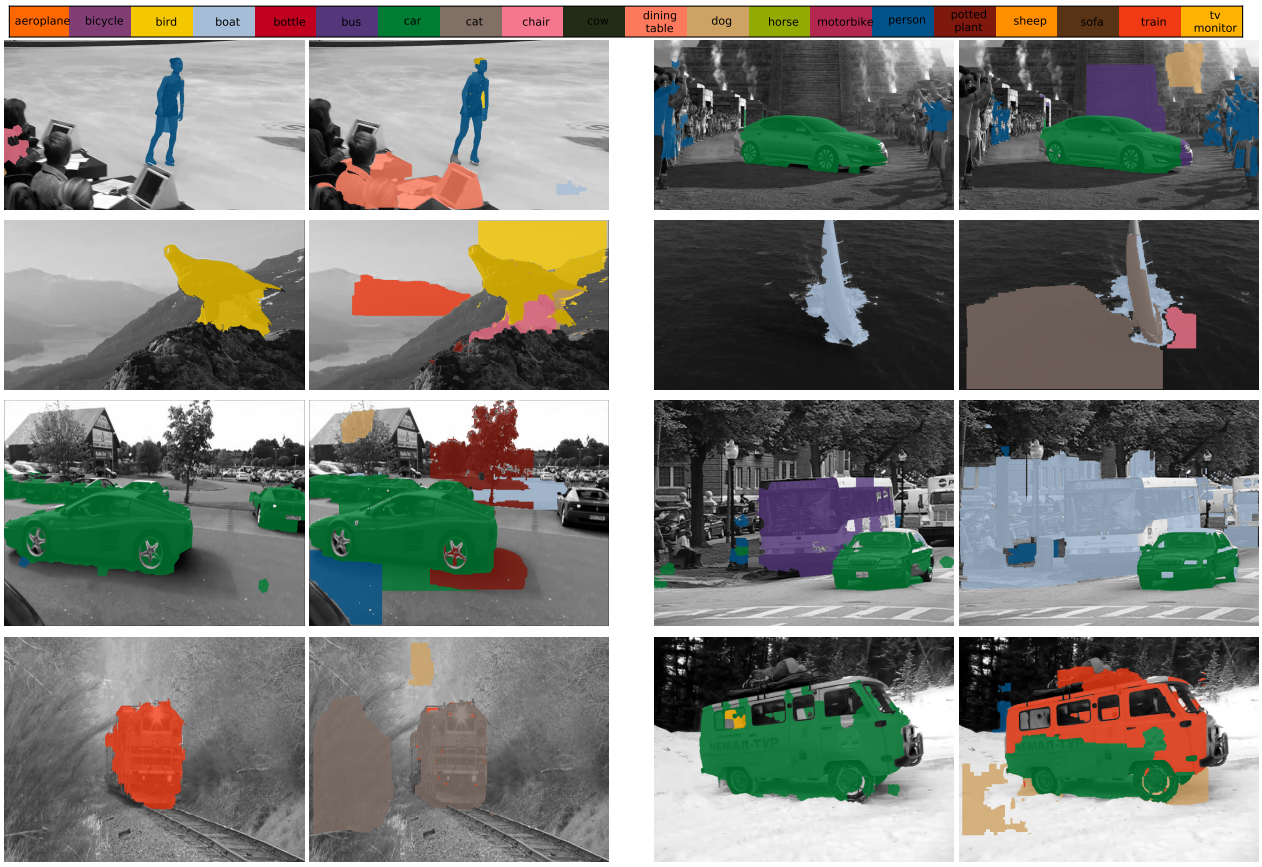


Figure 3.10: *Comparison with "PB". Our method is using 20% frames and temporal consistency, PB is using all frames independently. Temporal regularities allow us to remove many false positives present in PB. Color legend is shown on top. Sequences, from left to right, top to bottom: "Kim Yu Na", "Kia", "Planet Earth", "Sailing", "Ferrari", "Cars10", "Freight train", "Snow tour".*

3.4 Discussion

We have presented an uncertainty-based active selection approach to determine *which locations* of *which frames* in a video shot to run *which detector* on to arrive at a labeling of each pixel at the smallest computational cost that ensures a bound on residual uncertainty ($\sum_i H(c_i|\mathcal{Y}^k)$). We proposed two information-seeking criteria, MI and $\mathcal{E}$, and demonstrated that they outperm other selection schemes.

Unlike existing label propagation schemes that assume an oracle, we can handle uncertainty in the measurements, by leveraging an explicit probabilistic detector response model, a prior on classes learned from the PASCAL VOC dataset, and a hidden context variable global to each

video shot. Our method is causal, respects the spatio-temporal regularities in the video, and falls within the class of submodular optimization problems that enjoy desirable bounds in performance of greedy inference relative to the (intractable) optimum.

We compare the performance of our scheme on various baselines, including "paragons" running all detectors at all locations in all frames. In the presence of reliable detectors (an oracle, in the limit), a manyfold reduction of computational cost is possible with negligible performance drop.

An information-gathering approach that we described in this chapter has natural applications in the field of "active perception"[Baj88], where the *control* is not a choice of which data to *process*, but a choice of which data to *acquire* – by physically moving the visual sensor to a new location. The next chapter continues the theme of "information acquisition" in that situation.

3.5 Proofs

To approximate the frame selection criterion (3.7), we used an upper bound (3.8):

$$\mathbb{I}(y(s); c_i|\mathcal{Y}^k) \leq \sum_{m=0,n=0}^{L,L} w_m w_n \sum_{j=1}^{L} \frac{\eta_{jn}}{\eta_{jm}} - L \tag{3.13}$$

$$w_m \doteq p(c_i = m|\mathcal{Y}^k)$$

$$\eta_{jm} \doteq \begin{cases} \lambda_{j,\mathsf{on}} \text{ if } j = m \\ \lambda_{j,\mathsf{off}} \text{ if } j \neq m \end{cases} .$$

We now prove this claim: Rem.1 proves an upper bound on differential entropy and Rem.2 relates this bound to mutual information.

Remark 1. *Let* $p(y) = \sum_{i=0}^{L} w_i p(y|c = i)$ *where* $y \in \mathbb{R}^L$, $w_i \doteq p(c = i)$, $p(y|c = i) = \prod_{j=1}^{L} p(y^j|c = i)$ *and* $p(y^j|c = i) = \eta_{ji} \exp(-\eta_{ji} y^j)$. *Then, the differential entropy is bounded above by:*

$$H(y) \leq -\sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} + \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \frac{\eta_{jl}}{\eta_{ji}} \tag{3.14}$$

Proof.

$$H(y) = -\int \sum_{i=0}^{L} w_i p(y|c=i) \log\Big(\sum_{l=0}^{L} w_l p(y|c=l)\Big) dy \tag{3.15}$$

$$\leq -\int \sum_{i=0}^{L} w_i p(y|c=i) \sum_{l=0}^{L} w_l \log\Big(p(y|c=l)\Big) dy \tag{3.16}$$

$$= -\sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \int p(y|c=i) \log\Big(p(y|c=l)\Big) dy \tag{3.17}$$

$$= -\sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \int p(y|c=i) \Big(\sum_{j=1}^{L} \log \eta_{jl} - y^j \eta_{jl}\Big) dy \tag{3.18}$$

$$= -\sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} + \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \eta_{jl} \int y^j p(y|c=i) dy \tag{3.19}$$

$$= -\sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} + \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \frac{\eta_{jl}}{\eta_{ji}} \tag{3.20}$$

where (3.16) follows by Jensen's inequality, and the last line follows because:

$$(i) \quad \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} = \big(\sum_{i=0}^{L} w_i\big)\big(\sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl}\big) \tag{3.21}$$

$$(ii) \quad \sum_{i=0}^{L} w_i = 1 \tag{3.22}$$

$$(iii) \quad \int y^j p(y|c=i) dy = \mathbb{E}[y^j|c=i] = \eta_{ji}^{-1} \tag{3.23}$$

□

Remark 2. *When assumptions of Remark 1 hold,* $\mathbb{I}(y;c)$ *is upper bounded by*

$$\mathbb{I}(y;c) \leq \sum_{i=0,l=0}^{L,L} w_i w_l \sum_{j=1}^{L} \frac{\eta_{jl}}{\eta_{ji}} - L \tag{3.24}$$

Proof. First notice that $H(y|c)$ can be written as

$$H(y|c) = \sum_{l=0}^{L} w_i H(y|c=l) \tag{3.25}$$

$$= \sum_{l=0}^{L} w_l \sum_{j=1}^{L} H(y^j|c=l) \tag{3.26}$$

$$= \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \left(1 - \log \eta_{jl}\right) \tag{3.27}$$

$$= L - \sum_{i=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} \tag{3.28}$$

Then, using the result of Remark 1, we have that

$$\mathbb{I}(y;c) = H(y) - H(y|c) \tag{3.29}$$

$$\leq -\sum_{l=0}^{L} w_l \sum_{j=1}^{L} \log \eta_{jl} + \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \frac{\eta_{jl}}{\eta_{ji}} - H(y|c) \tag{3.30}$$

$$= \sum_{i=0}^{L} w_i \sum_{l=0}^{L} w_l \sum_{j=1}^{L} \frac{\eta_{jl}}{\eta_{ji}} - L \tag{3.31}$$

□

This quantity is related to our problem by choosing $w_m = p(c_i = m|\mathcal{Y}^k)$ and η_{ji} as

$$\eta_{jm} = \begin{cases} \lambda_{j,\mathsf{on}} \text{ if } j = m \\ \lambda_{j,\mathsf{off}} \text{ if } j \neq m \end{cases} \tag{3.32}$$

The comparison of this upper bound and Monte Carlo estimate of mutual information is shown in Fig. 3.11, using "Kim Yu Na" and "Visor" sequences as examples. Note that our upper bound does not grossly change the global maximum.

Remark 3. *Mutual information is submodular if observations are conditionally independent given the state. [KG05, Cor. 3]*

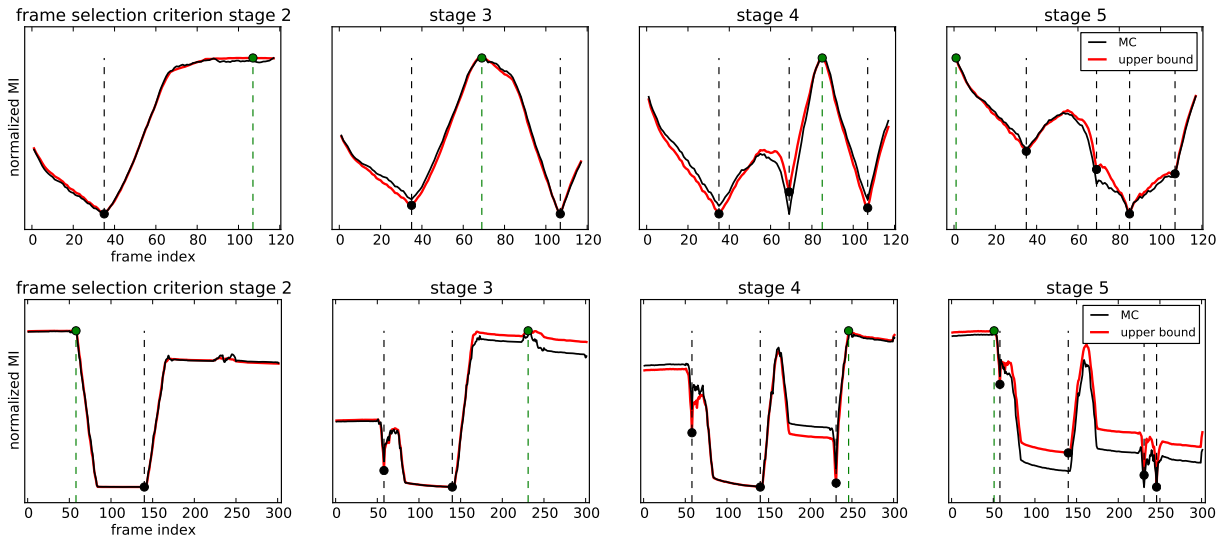


Figure 3.11: *Comparison of the upper bound (red) and the Monte Carlo (black) approximations of mutual information. Top:* "Kim Yu Na", *Bottom:* "Visor". *Both quantities are normalized, since we are interested in locations of global maxima, rather than absolute amplitudes. These plots illustrate that absolute maxima of the upper bound approximation and the Monte Carlo estimate are in rough agreement; thus, the upper bound is an acceptable proxy.*

A submodular function F satisfies

$$F(A \cup Z) - F(A) \geq F(B \cup Z) - F(B), \quad \textit{for } Z \textit{ a singleton and } A \subset B \tag{3.33}$$

If A is an index set, $y_A = (y_{a_1}, y_{a_2}, ..., y_{a_{|A|}})$, and $p(y_A|x) = \prod_{j \in A} p(y_j|x)$, then $F(A) \doteq \mathbb{I}(y_A; x)$ is a submodular function.

Proof. We need to show that (3.33) is satisfied. Note that $\mathbb{I}(y_{A \cup Z}; x) - \mathbb{I}(y_A; x) = \mathbb{I}(y_Z; x|y_A)$. Hence, we only need to show:

$$\mathbb{I}(y_Z; c|y_A) \geq \mathbb{I}(y_Z; x|y_B) \tag{3.34}$$

This is proved in [Lin56, Thm. 3], which is paraphrased here for completeness:

$$\mathbb{I}(y_Z; x|y_B) = H(y_Z|y_B) - H(y_Z|y_B, x) \tag{3.35}$$

$$= H(y_Z|y_B) - H(y_Z|y_A, x) \tag{3.36}$$

$$\leq H(y_Z|y_A) - H(y_Z|y_A, x) \tag{3.37}$$

$$= \mathbb{I}(y_Z; x|y_A) \tag{3.38}$$

where (3.36) follows from the conditional independence assumption: if $p(y_Z|y_B, x) = p(y_Z|x) = p(y_Z|y_A, x)$, then also $H(y_Z|y_B, x) = H(y_Z|x) = H(y_Z|y_A, x)$. (3.37) follows from "conditioning reduces entropy" property ($H(y_Z|y_B) \leq H(y_Z|y_A)$ since $A \subset B$). □

CHAPTER 4

Active Control for Visual Search and Control-Recognition Tradeoffs

We focus on the problem of "visual search" of an unknown object in an otherwise known and static scene. We propose a simple model that captures the phenomenology of image formation, including scaling and occlusions. An optimal control strategy in this setting is intractable, and we propose a suboptimal, yet efficient heuristic, and empirically verify its performance. Finally, we describe the tradeoff between the performance in a visual recognition problem and the control authority that the agent can exercise on the sensing process.

4.1 Introduction

We are interested in visual recognition of objects and scenes embedded in physical space. Rather than using datasets consisting of collections of isolated snapshots, however, we wish to actively control the sensing process during the recognition process. This is because, in the presence of nuisance factors involving occlusion and scale changes, learning requires mobility [Soa11]. Visual learning is thus a process of discovery, literally *uncovering* occluded portions of an object or scene, and viewing it from close enough that all structural details are revealed.

Within this scenario, exploration and search can be framed as optimal control and optimal stopping time problems. These relate to active vision (next-best-view generation), active learning, robotic motion planning, sequential decision in the setting of partially-observable Markov decision processes (POMDP) and a number of related fields. As often in this class of problems, inference algorithms are essentially intractable, so we wish to design surrogate tasks and prove performance bounds to ensure desirable properties of the surrogate solution.

In this chapter we consider the problem of detecting and estimating discrete parameters of an unknown object in a known environment. To this purpose we describe the simplest model that includes scaling and occlusion nuisances, a two dimensional "cartoon flatland," and a test suite to perform simulation experiments. We derive an explicit probability model to compute the posterior density given photometric measurements. We discuss and test algorithms for visual search based on the maximization of the conditional entropy of future measurements and the proxies of this quantity. These algorithms can be used to locate an unknown object in unknown position of a known environment, or to perform change detection in an otherwise known map, for the purpose of updating it. We provide experimental validation of the algorithms, including *regret* and expected exploration length.

We also describe tradeoffs between the performance in visual decision problems and the control authority that the agent can exercise on the sensing process. We propose a measure of control authority and empirically relate it to the expected risk and its proxy (conditional entropy of the posterior density). We then show that a "passive" agent can provide no guarantees on performance beyond what is afforded by the priors, and that an "omnipotent" agent, capable of infinite control authority, can achieve arbitrarily good performance (asymptotically). In between these limiting cases, the tradeoff is characterized empirically.

4.1.1 Related work

Active search and recognition of objects in the scene has been one of the mainstays of Active Perception in the eighties [Baj88, Bal91], and has recently resurged (see [AT09] and references therein). The problem can be formulated as a POMDP [RGT05], solving which requires developing approximate, near-optimal policies. Active recognition using next-best-view generation and object appearance is discussed in [KPP00] where authors use PCA to embed object images in a linear, low dimensional space. The scheme does not incorporate occlusions or scale changes. More recently, information driven sensor control for object recognition was used in [ES10, MB10, HMS11], who deal with visual and sonar sensors, but take features (*e.g.* SIFT, SURF) to be the observed data. A utility function that accounts for occlusions, viewing angle, and

distance to the object is proposed in [JSC11] who aim to actively learn object classifiers during the training stage. Exploration and learning of 3D object surface models by robotic manipulation is discussed in [KCF11]. The case of object localization (and tracking if object is moving) is discussed in [BGF04]; information-theoretic approach for solving this problem using a sensor network is described in [HT10]. Both authors used realistic, nonlinear sensor models, which however are different from photometric sensors and are not affected by the same nuisances.

With regards to models, our work is different in several aspects: instead of choosing the next best view on a sphere centered at the object as in [KPP00], we model a cluttered environment where the object of interest occupies a negligible volume and is therefore fully occluded when viewed from most locations. Second, we wish to operate in a continuous environment, rather than in a world that is discretized at the outset. Third, given the significance of quantization-scale and occlusions in a visual recognition task, we model the sensing process such that it accounts for both.

4.2 Formulation

Let $\mathbf{y} \in \mathcal{Y}$ denote data (measurements) and $\mathbf{x} \in \mathcal{X}$ a hidden class variable from a finite alphabet that we are interested in inferring. Recall that the expected Bayes risk with Hamming loss can be written as

$$\mathcal{E}^*(y) = 1 - \max_x p(x|y) \tag{4.1}$$

and minimized by Bayes' decision rule ($x^* = \arg\max_x p(x|y)$), which chooses the class label with maximum a posteriori probability. We are interested in controlling the data acquisition process so as to make this risk as small as possible. We use the problem of visual search (finding a not previously seen object in a scene) as a motivation. It is related to active learning and experimental design. In order to enforce temporal continuity, we model the search agent ("explorer") as a dynamical system of the form:

$$\begin{cases} \xi_{t+1} = & \xi_t \\ g_{t+1} = & f(g_t, u_t) \\ y_t = & h(g_t, \xi) + n_t \end{cases} \tag{4.2}$$

where g_t denotes the pose state at time t, u_t denotes the control, and ξ denotes the *scene* that describes the search environment – a collection of objects (simply-connected surfaces supporting a radiance function) of which the target $\mathbf{x}$ is one instance. Constraints on the controller enter through f; photometric nuisances, quantization and occlusions enter through the measurement map h. Additive and unmodeled phenomena that affect observed data are incorporated into n_t, the "noise" term.

4.2.1 Signal models

We represent both the shape and the appearance of the scene as described in [Soa11]. We denote an instance of this model, the scene, by $\xi = (\beta_1, \ldots, \beta_C)$, where $\beta_i = (\rho_i, S_i)$ is an object described by appearance ρ_i and shape S_i. While this representation allows us to reason about occlusions and finite sensor resolution, it is somewhat challenging to represent (we prefer to avoid dealing with shape spaces). We simplify the representation by projecting it into $\mathbb{R}^2$ and parameterizing both ρ_i and S_i.

In this case, the model can be called "cartoon flatland", where a bounded subset of $\mathbb{R}^2$ is populated by self-luminous line segments, corresponding to C clutter objects, each denoted by β_k. The number of objects in the scene C is the *clutter density* parameter that can possibly grow to be infinite in the limit. Each object is described by its center (c_k), length (l_k), binary orientation (o_k), and radiance function supported on the segment ρ_k. This is the "texture" or "appearance" of the object, which in the simplest case can be assumed to be a constant function:

$$\beta_k = (c_k, l_k, o_k, \rho_k) \in [0,1]^3 \times \{0,1\} \times [\mathbb{R}^2 \to \mathbb{R}^+] \tag{4.3}$$

An agent can move continuously throughout the search domain. We take the state $g_t \in \mathbb{R}^2$ to be its current position, $u_t \in \mathbb{R}^2$ the currently exerted move, and assume trivial dynamics: $g_{t+1} = g_t + u_t$. More complex agents where $g_t \in SE(3)$ can be incorporated without conceptual difficulties.

The measurement model is that of an omnidirectional m-pixel camera, with each entry of

$y_t \in \mathbb{R}^m$ in (4.2) given by:

$$y_t(i) = \int_{(i-\frac{1}{2})\frac{2\pi}{m}}^{(i+\frac{1}{2})\frac{2\pi}{m}} \int_0^\infty \rho_{\ell(\theta,g_t)}(z) d\theta d\tau + n_t(i), \text{ with } z = (\tau\cos(\theta), \tau\sin(\theta)) \tag{4.4}$$

where $\frac{2\pi}{m}$ is the angle subtended by each pixel. The integrand is a collection of radiance functions which are supported on objects (line segments). Because of *occlusions*, only the closest objects that intersect the pre-image contribute to the image. The index of the object (clutter or object of interest) that contributes to the image is denoted by $\ell(\theta, g_t)$ and is defined as:

$$\ell(\theta, g_t) = \arg\min_k \left\{ \gamma_k \middle| \exists (s_k, \gamma_k) \in [-\frac{l_k}{2}, \frac{l_k}{2}] \times \mathbb{R}_+ \text{ s.t. } c_k + \begin{pmatrix} o_k \\ 1 - o_k \end{pmatrix} s_k = g + \hat{g}(\theta)\gamma_k \right\} \tag{4.5}$$

Above, g and $\hat{g}(\theta) = (\cos(\theta), \sin(\theta))$ are current position and direction, respectively. c_k, l_k, and o_k are k-th segment center, length, and orientation. Condition $c_k + s_k = g + \hat{g}(\theta)\gamma_k$ encodes intersection of ray $g + \hat{g}(\theta)$ with a point on a segment k. The segment closest to viewer, *i.e.* one that is visible, has the smallest γ_k. Integration over $\frac{2\pi}{m}$ in (4.4) accounts for quantization, and the layer model (4.5) describes occlusions. While the measurement model is non-trivial (in particular, it is not differentiable), it is the simplest that captures the nuisance phenomenology. All unmodeled phenomena are lumped in the additive term n_t, which we assume to be zero-mean Gaussian "noise" with covariance $\sigma^2 I$.

In order to design control sequences to minimize risk, we need to evaluate the uncertainty of future measurements, those we have not yet measured, which are a function of the control action to be taken. To that end, we write the probability model for computing the posterior and the predictive density. We first describe the general case of *visual exploration* where the entire environment is unknown. We begin with noninformative prior for objects $k = 1, \ldots, C$

$$p(\beta_k) = p(c_k)p(l_k)p(o_k)p(\rho_k) = U[0, N_c]^2 \times Exp(\lambda) \times Ber(1/2) \times U[0, N_\rho] \tag{4.6}$$

where U,Exp and Ber denote uniform, exponential, and Bernoulli distributions parameterized by

N_c, λ, and N_ρ. Then $p(\xi) = p(\beta_1, ..., \beta_C)$. We can write the posterior as:

$$p(\xi|y^t, g^t) \propto \prod_{\tau=1}^{t} p(y_\tau|g_\tau, \xi)p(\xi) = \prod_{\tau=1}^{t} \mathcal{N}(y_\tau - h(g_\tau, \xi); \sigma^2 I)p(\xi) \tag{4.7}$$

Above, $\mathcal{N}(z, \Sigma)$ denotes the value of a zero-mean Gaussian density with covariance Σ at z. The density can be decomposed as a product of likelihoods since knowledge of environment (ξ) and location (g_t) is sufficient to predict measurement y_t up to Gaussian noise. The predictive distribution (distribution of the next measurement conditioned on the past) is computed by marginalization:

$$\begin{aligned} p(y_{t+1}|y^t, g^t, g_{t+1}) &= \int p(y_{t+1}|\xi, y^t, g_{t+1})p(\xi|y^t, g^t, g_{t+1})d\xi & (4.8) \\ &= \int \mathcal{N}(y_{t+1} - h(g_{t+1}, \xi), \sigma^2 I)p(\xi|y^t, g^t)d\xi & (4.9) \end{aligned}$$

The marginalization above is essentially intractable. In this chapter we focus on *visual search* of a particular object in an otherwise known environment, so marginalization is only performed with respect to a single object in the environment, x, whose parameters are discrete, but otherwise analogous to (4.6):

$$p(x) = U\{0, ..., N_c - 1\}^2 \times Exp(\lambda) \times Ber(1/2) \times U\{0, \ldots, N_\rho - 1\} \tag{4.10}$$

We denote by x_i, $i = 1, ..., |\mathcal{X}|$ object with parameters (c_i, l_i, o_i, ρ_i) and write $\xi_i = (x_i, \beta_1, \ldots, \beta_C)$ to denote the scene with *known* clutter objects $\beta_1, ..., \beta_C$ augmented by an *unknown* object x_i. In this case, we have:

$$p(x|y^t, g^t) \propto \prod_{\tau=1}^{t} \mathcal{N}(y_\tau - h(g_\tau, \xi); \sigma^2 I)p(x) \tag{4.11}$$

$$p(y_{t+1}|y^t, g^t, g_{t+1}) = \sum_{i=1}^{|\mathcal{X}|} \mathcal{N}(y_{t+1} - h(g_{t+1}, \xi_i), \sigma^2 I)p(x_i|y^t, g^t) \tag{4.12}$$

4.2.2 Control policy

Given g_t and ξ, in order to minimize average risk (4.1) with respect to a sequence of control actions we formulate a finite k-step horizon optimal control problem:

$$u_t^{*t+k-1} = \arg\min_{u_t^{t+k-1}} \int p(y_{t+1}^{t+k}|y^t, u_t^{t+k-1}) \underbrace{\left(1 - \max_i p(x_i|y^t, y_{t+1}^{t+k}, u_t^{t+k-1})\right)}_{\mathcal{E}^*(y^{t+k})} dy_{t+1}^{t+k} \tag{4.13}$$

which is unfortunately intractable. As is standard, we can settle for the greedy $k = 1$ case:

$$u_t^* = \arg\min_{u_t} \int p(y_{t+1}|y^t, u_t) \underbrace{\left(1 - \max_i p(x_i|y^t, y_{t+1}, u_t)\right)}_{\mathcal{E}^*(y^{t+1})} dy_{t+1} \tag{4.14}$$

but it is still often impractical. We relax the problem further by choosing to minimize the upper bound on Bayesian risk, of which a convenient one is the conditional entropy (see [HR70], which shows $\mathcal{E}^*(y) \leq \frac{1}{2}H(x|y)$): This implies that control action can be chosen by entropy minimization:

$$u_t^* = \arg\min_{u_t} H(x|y^t, y_{t+1}, u_t) \tag{4.15}$$

Using chain rules of entropy, we can rewrite minimization of $H(x|y^t, y_{t+1}, u_t)$ as maximization of conditional entropy of next measurement:

$$\begin{aligned} u_t^* &= \arg\min_{u_t} H(x|y^t, y_{t+1}, u_t) && (4.16)\\ &= \arg\min_{u_t} H(x|y^t) - \mathbb{I}(y_{t+1}; x|y^t, u_t) && (4.17)\\ &= \arg\max_{u_t} H(y_{t+1}|y^t, u_t) - H(y_{t+1}|y^t, u_t, x) && (4.18)\\ &= \arg\max_{u_t} H(y_{t+1}|y^t, u_t) && (4.19) \end{aligned}$$

because $H(y_{t+1}|y^t, u_t, x) = H(n_t)$ is due to Gaussian noise, since $y_{t+1} = h(g_{t+1}; \xi) + n_{t+1}$ and both g_{t+1} and ξ are known (the only unknown object in the scene is x, and it is conditioned on). $H(y_{t+1}|y^t, u_t)$ is the entropy of a Gaussian mixture distribution which can be easily approximated by Monte Carlo, and for which both lower [HO07] and upper bounds [HBH08] are known.

Since the controller has energy limitations, *i.e.* is unable to traverse the environment in one step, optimization is taken over a small ball in $\mathbb{R}^2$ centered at current location g_t. In practice, the set of controls needs to be discretized and entropy computed for each action. However, rather than myopically choosing the next control, we instead choose the next target position, in a "bandit" approach [VTS12, VSK02]: maximization in (4.19) is taken with respect to all locations in the world, rather than the set of controls (locations reachable in one step), and the agent takes the minimum energy path toward the most informative location. Since this location is typically not reachable in a single step, one can adopt a "stubborn" strategy that follows the planned path to the target location before choosing next action, and an "indecisive" – that replans as soon as additional information becomes available as a consequence of motion. We demonstrate the characteristics of conditional entropy as a criterion for planning in Fig. 4.1.

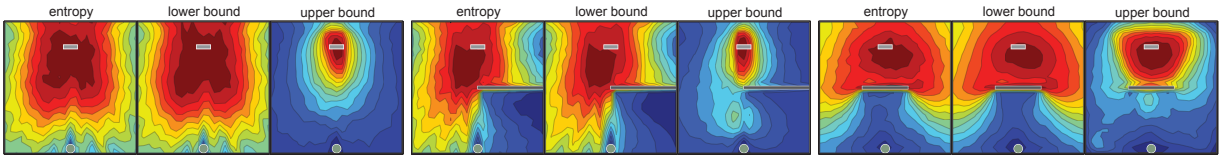


Figure 4.1: *"Value of measurement" described by conditional entropy $H(y_{t+1}|y^t, g)$ as a function of location g. We focus on three special cases, and for each show entropy, its lower bound (see [HO07]), and upper bound (based on Gaussian approximation, see [CT91]). In all cases, the agent is at the bottom of the environment, and a small unknown object is at the top. The agent has made one measurement (y_1) and must now determine the best location to visit. The left three panels demonstrate a case of scaling: object is seen, but due to noise and quantization its parameters are uncertain. Agent gains information if putative object location (top) is approached. Middle three panels demonstrate partial occlusion: a part of the object has been seen, and there is now a region (bottom right corner) that is uninformative – measurements taken there are predictable. Full occlusion is shown in the right three panels. The object has not been seen (due to occluder in the middle of the environment) and the best action is to visit new area. Notice that lower and upper bounds are maximized at the same point as actual entropy. This was a common occurrence in many experiments that we did. Because we are interested in the maximizing point, rather than the maximizing value, even if the bounds are loose, using them for navigation can lead to reasonable results.*

4.3 The role of control in active recognition

It is clear from equations (4.11), and (4.12) that the history of agent's positions g^t plays a key role in the process of acquiring new information on the object of interest x for the purpose of

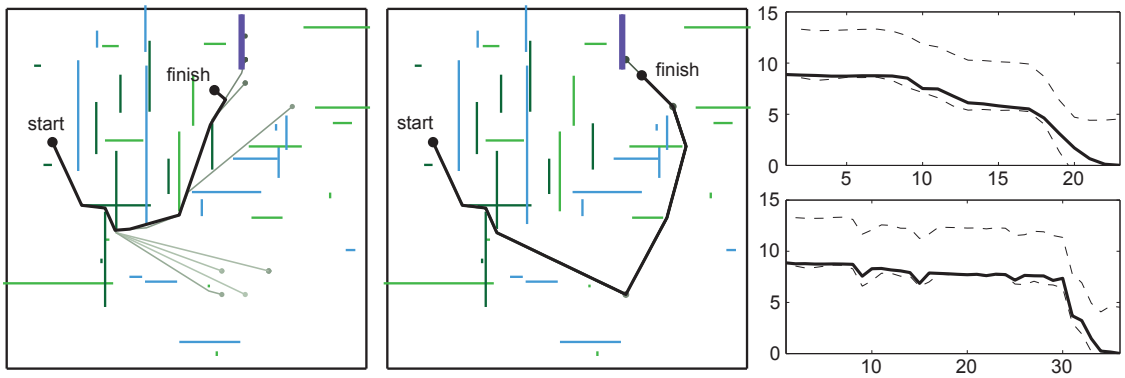


Figure 4.2: *A typical run of "indecisive" (left) and "stubborn" (right) strategies. Objects are colored according to their radiance and the unknown object is shown as a thick line. Traveled path is shown in black. The thinner lines are the planned paths that were not traversed to the end because of replanning. Stubborn explorer traverses each planned segment to its end. Right: Residual entropy $H(\mathbf{x}|y^t)$ shown over time for the two strategies (top: "indecisive", bottom: "stubborn"). Lower and upper bounds on $H(\mathbf{x}|y^t, y_{t+1})$ can be computed prior to measuring y_{t+1} using upper and lower bounds on $H(y_{t+1}|y^t)$. Sharp decrease occurs when object becomes visible.*

recognition. This is encoded by the conditional density (4.11). In the context of the identification of the model (4.2), one would say that data y^t (a function of the scene and the history of positions) must be sufficiently informative [Pro08] on x, meaning that y^t contains enough information to estimate x; this can be measured e.g. through the Fisher information matrix if x is deterministic but unknown, or by the posterior $p(x|y^t)$ in a probabilistic setting. This depends upon whether u^t is sufficiently exciting, a "richness" condition that has been extensively used in the identification and adaptive control literature [Bit84, Lju97], which guarantees that the state trajectory g^t explores the space of interest. If this condition is not satisfied, there are limitations on the performance that can be attained during the exploration process. There are two extreme cases which set an upper and lower bounds on recognition error:

Passive recognition: there is no active control, and instead a collection of vantage points g^t is given a-priori. Under this scenario it is easy to prove that, averaging over the possible scenes and initial agent locations, the probability of error approaches chance (i.e. that given by the prior distribution) as clutter density and/or the environment volume increase.

Full control on g^t: if the control action can take the "omnipotent agent" anywhere, and infinite time is available to collect measurements, then the conditional entropy $H(x|y^t)$ decreases asymptotically to zero thus providing arbitrarily good error rate in the limit.

In general, there is a tradeoff between the ability to gather new information through suitable control actions, which we name "control authority", and the recognition rate. In the sequel we shall propose a measure for the "control authority" over the sensing process; later in the chapter we will consider conditional entropy as a proxy (upper bound) on probability of error and evaluate empirically how control authority affects the conditional entropy decrease.

4.3.1 Control authority

Unlike the passive case, in the controlled scenario time plays an important role. This happens in two ways. One is related to the ability to visit previously unexplored regions and therefore is related to the reachable space under input and time constraints, the other is the effect of noise which needs to be averaged. If objects in the scene move, this can be done only at an expense in energy, and achieving asymptotic performance may not be possible under control limitations. This considerably more complex scenario is beyond our scope in this work. We focus on the simplest case of static environment.

Control authority depends on (i) the controller u, as measured for instance by a norm[1] $\|u\| : \mathcal{U}[0,T] \to \mathbb{R}$ and (ii) on the geometry of the state space, the input-to-state map and on the environment. We propose to measure control authority in the following manner: associate to each pair of locations in the state space (g_o, g_f) and a given time horizon T the cost $\|u\|$ required to move from g_o at time $t = 0$ to g_f at time $t = T$ along a minimum cost path i.e.

$$J_\xi(g_o, g_f, T) \doteq \inf_{u \,:\, g_u(0)=g_o, g_u(T)=g_f \;\xi} \|u\| \tag{4.20}$$

where $g_u(t)$ is the state vector at time t under control u. If g_f is not reachable from g_o in time T we set $J_\xi(g_o, g_f, T) = \infty$. This will depend on the dynamical properties of the agent $\dot{g} = f(g, u)$ (or $g_{t+1} = f(g_t, u_t)$ for discrete time) as well as on the scene ξ where the agent has to navigate through while avoiding obstacles.

The control authority ($\mathcal{C}A$) can be measured via the volume of the reachable space for fixed

[1] This could be, for instance, total energy, (average) power, maximum amplitude and so on. We can assume that the control is such that $\|u\| \leq 1$

control cost, and will be a function of the initial configuration g_0 and of the scene ξ, i.e.

$$\mathcal{C}A(k, g_o, \xi) \doteq Vol\{g_f : J_\xi(g_0, g_f, k) \leq 1\} \tag{4.21}$$

If instead one is interested in *average* performance (e.g. w.r.t. the possible scene distributions with fixed clutter density), a reasonable measure is the average of smallest volume (as g_0 varies) of the reachable space with a unit cost input

$$\mathcal{C}A(k) \doteq E_\xi\big[\inf_{g_o} \mathcal{C}A(k, g_o, \xi)\big] \tag{4.22}$$

If planning on an indefinitely long time horizon is allowed, then one would minimize $J(g_o, g_f, T)$ over time T:

$$J(g_o, g_f) \doteq \inf_{T \geq 0} J(g_o, g_f, T) \tag{4.23}$$

with

$$\mathcal{C}A_\infty \doteq \inf_{g_o} \left(Vol\{g_f : J(g_o, g_f) \leq 1\}\right) \tag{4.24}$$

The figures $\mathcal{C}A(k, g_o, \xi)$ in (4.21), $\mathcal{C}A(k)$ and $\mathcal{C}A_\infty$ in (4.24) are proxies of the exploration ability which, in turn, is related to the ability to gather new information on the task at hand. The data acquisition process can be regarded as an experiment design problem [Pro08] where the choice of the control signal guides the experiment. Control authority, as defined above, measures how much freedom one has on the sampling procedure; the larger the $\mathcal{C}A$, the more freedom the designer has. Hence, having fixed (say) the number of snapshots of the scene one may consider the time interval over which these snapshots can be taken, the designer is trying to maximize the information the data contains on the task (making a decision on class label); this information is of course a non-decreasing function of $\mathcal{C}A$. More control authority corresponds to more freedom in the choice of which samples one is taking (from which location and at which scale). Therefore the risk, considered against $\mathcal{C}A(k)$ in (4.22), $\mathcal{C}A(k, g_o, \xi)$ in (4.21) or $\mathcal{C}A_\infty$ in (4.24) will follow a surface that depends on the clutter: For any given clutter (or clutter density), the risk will be a monotonically non-increasing function of control authority $\mathcal{C}A(k)$. This is illustrated in Fig. 4.4.

4.4 Experiments

We evaluate "indecisive" and "stubborn" control strategies described above, and also consider several different uncertainty measures. Section 4.2.2 provided arguments for $H(y_{t+1}|y^t, g)$ (a "max-ent" approach) which is a proxy for minimization of Bayesian risk. Another option is to maximize covariance of $p(y_{t+1}|y^t, g)$ ("max-var"), for example due to reduced computational cost. Alternatively, if we do not wish to hypothesize future measurements and compute $p(y_{t+1}|y^t, g)$, we may search by approaching the mode of the posterior distribution $p(x|y^t)$ ("max-posterior"). To test average performance of these strategies, we consider search in 100 environment instances, each containing 40 known clutter objects and one unknown object. Clutter objects are sampled from the continuous prior distribution (4.6) and unknown object is chosen from the prior (4.10) discretized to $|\mathcal{X}| \approx 9000$. Agent's sensor has $m = 30$ pixels, with additive noise σ set to half of the difference between object colors. Conditional entropy of the next measurement, $H(y_{t+1}|y^t, g_{t+1})$, is calculated over the entire map, on a 16x16 grid. Search is terminated once residual entropy falls below a threshold value: $H(x|y^t) < 0.001$. We are interested in average search time (expressed in terms of number of steps) and average *regret*, which we define as the excess fraction of the minimum energy path to the center of the unknown object (c_0) that the explorer takes:

$$\textbf{regret} \doteq \frac{c_u(x_o, x_c) - J(x_o, x_c)}{J(x_o, x_c)}. \tag{4.25}$$

Because it is not always necessary to reach the object to recognize it (viewing it closely from multiple viewpoints may be sufficient), this quantity is an approximation to minimum search effort. We show an example of a typical problem instance in Fig. 4.2. Statistics of strategies' performance are shown in Fig. 4.3. Minimum energy path and random walk strategy play roles of lower and upper bounds. For each of the three uncertainty measures, "indecisive" outperformed "stubborn" in terms of both average path length and average regret, as also shown in Table 4.1. Notice however that for specific problem instances "indecisive" can be much worse than "stubborn" – the curves for the two strategy types cross. Generally, "max-ent" strategy seems to perform best, followed by "max-var", and "max-posterior". "Random-walk" strategy was unable to find the object unless it was visible initially or became visible by chance.

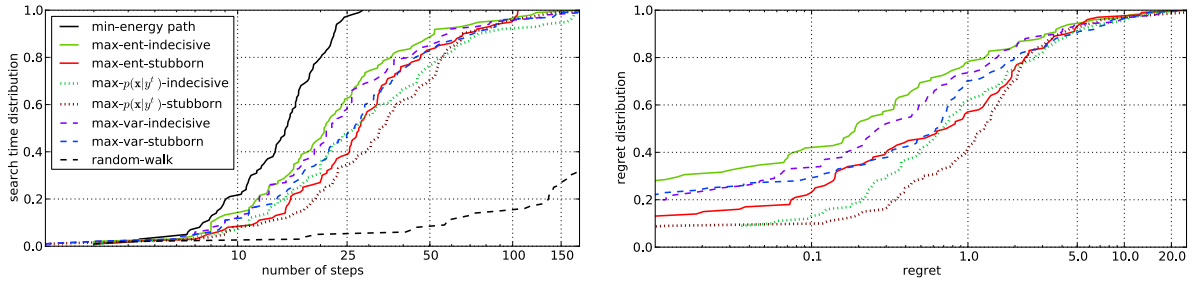


Figure 4.3: *Search time statistics for a 100 world test suite. Left: cumulative distribution of distance until detection traveled by the max-entropy, max-posterior, max-variance explorers, and random walker. Right: cumulative distribution of* regret *for the explorers.*

	Average search duration			Average regret		
	max-ent	max-var	max-$p(\mathbf{x}\|y^t)$	max-ent	max-var	max-$p(\mathbf{x}\|y^t)$
indecisive	28.42	32.70	41.00	1.27	1.44	1.96
stubborn	34.26	36.17	41.49	1.71	1.78	2.19

Table 4.1: *Search time statistics for different strategies.*

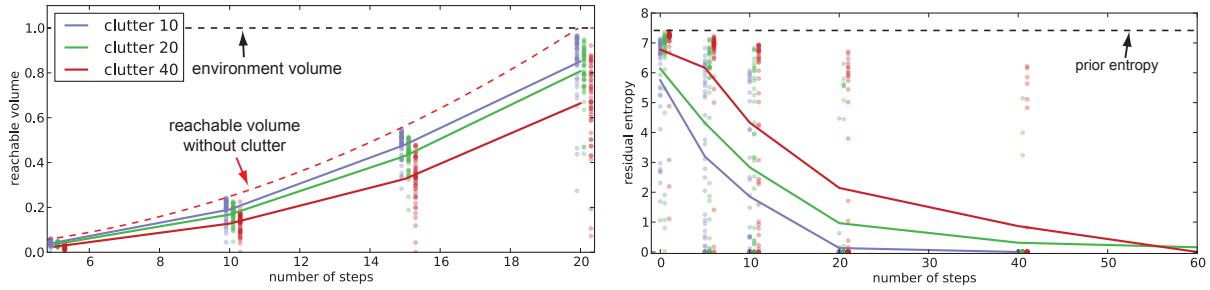


Figure 4.4: *Left: Control authority. The red dashed curve corresponds to reachable volume in the absence of clutter. The black dashed line is the normalized maximum reachable volume in the environment. Right: Residual entropy $H(x|y^t)$, as a function of control authority and clutter density. Black dashed line indicates $H(x)$, entropy prior to taking any measurements. Lines correspond to residual entropy for a given control authority averaged over the test suite; markers – to residual entropy on a specific problem instance. For certain scenes, agent is unable to significantly reduce entropy because the object never becomes unoccluded (once object is seen, there is a sharp drop in residual entropy, as shown in Fig. 4.2).*

We next empirically evaluated explorer's exploration ability under finite control authority. Reachable volume was computed by Monte Carlo sampling, following (4.21)-(4.22) for several clutter density values. For each clutter density, we generated 40 scene instances and tested "indecisive" max-entropy strategy with respect to control authority. Here $|\mathcal{X}| \approx 2000$, and other parameters remained as in previous experiment. Fig. 4.4 empirically verifies discussion in Section 4.3.

4.5 Conclusion

We have described a simple model that captures the phenomenology of nuisances in visual decision problems, that include uncertainty due to occlusion, scaling, and other "noise" processes. We have described greedy algorithms based on maximizing entropy of the prediction density. We have then related the amount of "control authority" the agent can exercise during the data acquisition process with the performance in the visual coverage task. The extreme cases show that if one is given a passively gathered dataset, performance cannot be guaranteed beyond what is afforded by the prior. In the limit of infinite control authority, arbitrarily good decision performance can be attained. In between, we have empirically characterized the tradeoff between decision performance and control authority.

We have constrained ourselves to the simple case of static scenes, with a single mobile agent. If the scene is populated by other agents, and there is a basic requirement of collision avoidance, the situation is best formulated by a game. A game-theoretic formulation can be avoided by assuming that other agents are oblivious to us – that they do not attempt to perform collision avoidance. However, even in that case, the problem is more difficult than what we discussed here, since unlike (4.2), now $\xi_{t+1} \neq \xi_t$. The fact that the scene is dynamic forces us to predict how it will evolve. If the dynamics of our agent were laggy, this prediction would need to be long term: we would need to estimate $p(\xi_{t+K}|\xi_t)$ for some $K > 1$. The next chapter avoids active control, and focuses on describing a rich model for generating such predictions.

CHAPTER 5

Intent-Aware Long-Term Prediction of Pedestrian Motion

We present a method to predict long-term motion of pedestrians, modeling them as jump-Markov processes with their goal a hidden variable. Assuming approximately rational behavior, and incorporating environmental constraints and biases, including time-varying ones imposed by traffic lights, we model intent as a policy in a Markov decision framework. We infer pedestrian state using a Rao-Blackwellized filter, and intent by planning according to a stochastic policy, reflecting individual preferences in aiming at the same goal. We demonstrate our method on a driving dataset and report typical performance.

5.1 Introduction

Safe operation of autonomous systems co-mingling with humans could benefit from some level of understanding of human behavior by the robot. One of the simplest requirements of safe operation is collision avoidance: The robot must avoid regions of space likely to be concurrently occupied by a human. This requires predicting human motion beyond the short time-horizon afforded by generic phenomenological models. Instead, future trajectories of humans depend on their goals, or intentions, which are not manifest to the viewer. We posit that knowledge of a person's goals, in conjunction with contextual knowledge can help prediction beyond a few seconds into the future. Since in reality we do not know a person's goals, we can infer it along a prediction of human behavior or marginalize it as a nuisance variable. In other words, while short-term prediction is sufficiently informed by recent past history, longer time-scale prediction depends on intents or goals. In this chapter we explore the problem of long-term prediction of pedestrian behavior in the autonomous (or assisted) driving scenario.

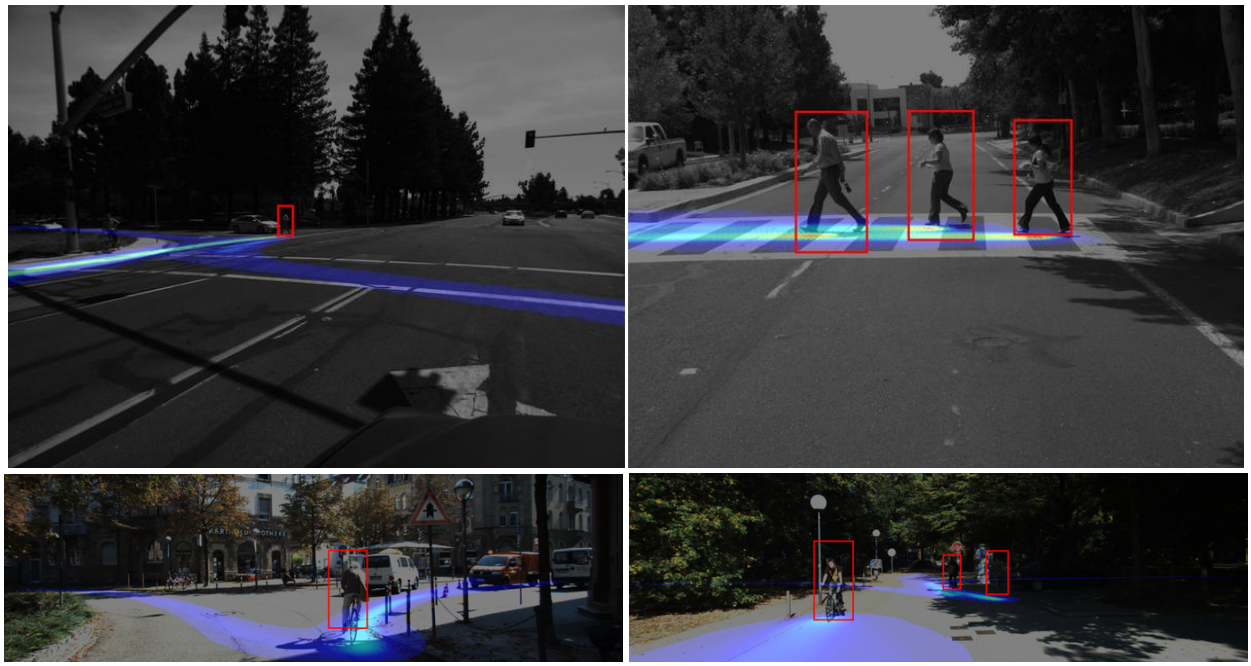


Figure 5.1: *Sample long-term predictions of traffic participants' motion generated by our model. Notice that the predictions are multi-modal and obey constraints of the environment.*

5.1.1 Related work and contributions

Common approaches to prediction of time series assume they are samples from a process generated by a linear model (*e.g.* ARMA) driven by a white, zero-mean, Gaussian input. Traditional tools from filtering and system identification can then be used to infer the state and model parameters [Jaz70]. When the order of the model is restricted, it yields predictions that are accurate at short time-scales, but insufficient beyond a few steps, as the processes it represents are stationary. It is also difficult to embed context, such as partial knowledge of the environment, into these models while preserving their structure. Extension to switching models (*e.g.* PWARX) have been used in constrained scenarios [SG13, KSF14, BS09] but again their predictive ability has only been demonstrated on short horizons. A nonparametric approach based on Gaussian processes proposed in [ESR09] is accurate on longer time-scales, but requires significant computational efforts. Similarly, models of "social" interaction [PES09, YBO11, CPS13] are complex, yet only enable short-term prediction. Thus, we focus on each pedestrian individually, discarding their interactions.

A common approach to improve accuracy of long-term prediction has been to postulate "goals"

or "destinations", and to assume that agents navigate toward them by approximately following stochastic shortest paths [FT02, BST09, ZRG09, BCH12, KZB12, XTZ13]. This framework allows one to represent rich dynamics, and to naturally incorporate environment constraints – that the agent cannot navigate through "obstacles" [FT02, ZRG09, XTZ13], and that regions may be more or less preferred depending on their semantic category [KZB12]. During training, this framework requires one to identify the shortest path policy used by the agents – *i.e.* to solve an inverse reinforcement learning problem [NR00, ZMB08, RA07]. At test time, one uses the agent's recent measured motion to estimate the goal, whose knowledge then completely reveals future trajectories. We show that these models of agents' behavior can be interpreted as switching nonlinear dynamical systems, with the latent goal variable governing the switches, and the policy describing the nonlinear motion dynamics.

In this chapter, we provide an interpretation of models in [BST09, ZRG09, KZB12] as jump-Markov processes. The large body of work on this topic provides guidance on how estimation/prediction can be approached, while systematically accounting for the uncertainty in dynamics or measurements. We show how the abovementioned discrete-space models can be applied to a problem in continuous space by adding speed as an additional latent variable, and extend the model to handle dynamic environments and additional measurements.

Sec. 5.2 describes the framework: Sec. 5.2.1 is dedicated to estimation and prediction, Sec. 5.2.2 focuses on the underlying optimal control problems, Sec. 5.2.3 explains the extension to dynamic environments. Sample performance of our framework is shown in Fig. 5.1, and we further demonstrate the performance in Sec. 5.3. Sec. 5.4 is reserved for discussion.

5.2 Formalization

The state of the pedestrian is described by her position, speed, and orientation in the global coordinate frame $(x, s, \theta) \in \mathbb{R}^2 \times \mathbb{R}_+ \times \mathbb{S}$, and the (unknown) goal $\mathbf{g} \subset \mathbb{R}^2$ (possibly just a point) from a finite set $\mathbf{G}$ which is known a-priori.

We define *intent* to be a function π mapping goals to *actions*: given $\mathbf{g}$ and a current state $\mathbf{x} = (x, s, \theta)$, an intent π yields future physical state trajectories from the current time t to a

future time when the goal is achieved. Because different individuals having the same goal will act differently, the map π is stochastic. In the language of Markov decision processes (MDPs), an intent π is called a *policy*, a sample from it, integrated over time, is called a *plan*, which at each instant of time corresponds to an *action*. For simplicity, we assume that the stochastic component of the intent is represented by a process w_π, given which the policy π has a known functional form. So, we write $\pi = \pi(\mathbf{x}, \mathbf{g}, w_\pi)$ to indicate that, given a sample w_π and a goal $\mathbf{g}$, the policy uniquely determines an action.

We assume that the goal $\mathbf{g}$ is slowly time-varying; specifically that at each time instant it switches to another, uniformly chosen goal with a small probability. If we exploit finiteness of $\mathbf{G}$ to interpret $\mathbf{g}$ as an *index* in $\{1, \dots, |\mathbf{G}|\}$ (naturally mapping to a corresponding *goal region*), then we can summarize these dynamics by a distribution $p(\mathbf{g}_{t+1}|\mathbf{g}_t)$, or by the relation $\mathbf{g}_{t+1} = \mathbf{g}_t \oplus w_{g,t}$ where w_g is an integer-valued random process, and the operation is addition modulo $|\mathbf{G}|$. This allows us to write the model as a discrete-time jump-Markov process:

$$\begin{cases} \mathbf{x}_{t+1} = f(\mathbf{x}_t, \pi(\mathbf{x}_t, \mathbf{g}_t, w_{\pi,t})) \\ \mathbf{g}_{t+1} = \mathbf{g}_t \oplus w_{g,t} \end{cases} \tag{5.1}$$

where the low-level pedestrian dynamics are abstracted into f. This is a model of a stochastic hybrid system with continuous states $\mathbf{x}$ and discrete state $\mathbf{g}$, and feedback π. As a further simplification, we assume that π is only affected by the positional component x of $\mathbf{x}$. This is equivalent to saying that the goal only affects the *direction* of heading, whereas the speed at which the pedestrian walks is unknown, but otherwise unaffected by the goal. This is clearly a simplification (one may walk at different speeds depending on where s/he is heading), but sufficient for our purpose. We

therefore summarize the model as:

$$\begin{cases} x_{t+1} & = x_t + s_t \begin{bmatrix} \cos(\theta_t) \\ \sin(\theta_t) \end{bmatrix} \\ s_{t+1} & = s_t + w_{s,t} \\ \theta_{t+1} & = \pi(x_t, \mathbf{g}_t, w_{\pi,t}) \\ \mathbf{g}_{t+1} & = \mathbf{g}_t \oplus w_{g,t} \end{cases} \tag{5.2}$$

where $(w_{\pi,t}, w_{s,t}, w_{g,t}) \sim P_w$ are the state transition processes, with distribution P_w. The pose of our vehicle in the global coordinate frame is described by $(x_{\text{ego},t}, \theta_{\text{ego},t})$ and is assumed to be known. The observation $\mathbf{y}_t$ consists of the pedestrian position measured relative to the vehicle:

$$\mathbf{y}_t = \mathsf{R}(\theta_{\text{ego},t})^T (x_t - x_{\text{ego},t}) + n_t \tag{5.3}$$

where $\mathsf{R} : \mathbb{S}^1 \to SO(2)$ maps orientation to a rotation and $n_t \sim P_Q$ is the measurement noise.

If we knew the goal state $\mathbf{g}_t$, assuming rational behavior of the agent, we could infer her intent by computing the (future) state trajectory that minimizes expected time-to-goal or, equivalently in our case, path cost. Unfortunately, we do not know the goal state, which we must instead infer using past state trajectory, which is only known through the history of measurements up to time t, i.e. $\mathbf{y}_1, \ldots, \mathbf{y}_t$ or $\mathbf{y}^t$. The goal state could be inferred along with the physical state by estimating (filtering) the posterior $p(\mathbf{x}_t, \mathbf{g}_t | \mathbf{y}^t)$. A filter maintains an estimate of the filtering density by computing the prediction $p(\mathbf{x}_{t+1}, \mathbf{g}_{t+1} | \mathbf{x}_t, \mathbf{g}_t)$ using Chapman-Kolmogorov's equation and knowledge of $P_w, \mathbf{G}$, and updating the prediction once measurements $\mathbf{y}_{t+1}$ become available: $p(\mathbf{x}_{t+1}, \mathbf{g}_{t+1} | \mathbf{x}_t, \mathbf{g}_t, \mathbf{y}_{t+1})$ using Bayes' rule and knowledge of P_Q.

The filtering density can be approximated by a generic particle filter, which maintains a sample-based representation of the posterior. However, in our case we exploit the structure of the model and apply a lower-complexity Rao-Blackwellized filter, as described in Sec. 5.2.1. Once we have the filtering density, we can infer intent by solving the same stochastic optimal control problem that, presumably, a rational agent solves. To do that, we must know the reward function, which we

Algorithm 2 Rao-Blackwell particle filter

for $i = 1 \dots |\mathbf{G}|$ **do**
$p(\mathbf{x}_t|\mathbf{y}^{t-1}, \mathbf{g}_{t-1} = i) \leftarrow \mathbf{predict}\{p(\mathbf{x}_{t-1}|\mathbf{y}^{t-1}, \mathbf{g}_{t-1} = i)\}$
end for
Compute marginal likelihood:
$p(\mathbf{y}_t|\mathbf{y}^{t-1}, \mathbf{g}_t = i) = \int p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{g}_t = i)p(\mathbf{x}_t|\mathbf{y}^{t-1}, \mathbf{g}_t = i)d\mathbf{x}$
Predict $\mathbf{g}$:
$p(\mathbf{g}_t|\mathbf{y}^{t-1}) = \sum_{\mathbf{g}_{t-1}} p(\mathbf{g}_t|\mathbf{g}_{t-1})p(\mathbf{g}_{t-1}|\mathbf{y}^{t-1})$
Update posterior over $\mathbf{g}$:
$p(\mathbf{g}_t|\mathbf{y}^t) \propto p(\mathbf{y}_t|\mathbf{y}^{t-1}, \mathbf{g}_t)p(\mathbf{g}_t|\mathbf{y}^{t-1})$
for $i = 1 \dots |\mathbf{G}|$ **do**
$p(\mathbf{x}_t|\mathbf{y}^t, \mathbf{g}_t = i) \leftarrow \mathbf{update}\{\mathbf{y}_t, p(\mathbf{x}_t|\mathbf{y}^{t-1}, \mathbf{g}_t = i)\}$
end for

Algorithm 3 Sampling state trajectory $\hat{\mathbf{x}}_t^{t+\tau}, \hat{\mathbf{g}}_t^{t+\tau}$

Generate $\hat{\mathbf{x}}_t, \hat{\mathbf{g}}_t \sim p(\mathbf{x}_t|\mathbf{y}^t, \mathbf{g}_t)p(\mathbf{g}_t|\mathbf{y}^t)$
for $k = 1 \dots \tau$ **do**
$\hat{\mathbf{g}}_{t+k} \sim p(\mathbf{g}_{t+k}|\hat{\mathbf{g}}_{t+k-1})$
$\hat{\mathbf{x}}_{t+k} \sim p(\mathbf{x}_{t+k}|\hat{\mathbf{x}}_{t+k-1}, \hat{\mathbf{g}}_{t+k-1})$
end for

obtain by leveraging the contextual information in the form of a semantic map, as described in Sec. 5.2.2. In Sec. 5.2.3 we extend the model in (5.2) to account for pedestrians changing their behavior with respect to changes in the environment. In an assisted driving scenario, an important example of this involves pedestrian behavior being influenced by traffic rules, including time-varying ones enforced by traffic lights. In Sec. 5.2.4 we show that the inference is improved via additional measurements of the pedestrian's orientation. Performance of our proposed model is compared to multiple baselines in Sec. 5.3.

5.2.1 Filtering and prediction

To infer the state posterior we use a Rao-Blackwellized particle filter (RBPF) [DDM00]. In our case, the the posterior is represented by a discrete distribution over $\mathbf{G}$, $p(\mathbf{g}_t|\mathbf{y}^t)$, and by a set of $|\mathbf{G}|$ filters (Kalman or particle filters) that approximate each distribution $p(\mathbf{x}_t|\mathbf{g}_t = i, \mathbf{y}^t)$. In Alg. 2 we list the RBPF filtering steps. The "predict" and "update" steps are the standard steps performed by either Kalman of particle filters.

Once we have the posterior $p(\mathbf{g}_t, \mathbf{x}_t|\mathbf{y}^t)$, we can perform prediction. We are interested in the state τ steps into the future, i.e. $p(\mathbf{x}_{t+\tau}, \mathbf{g}_{t+\tau}|\mathbf{y}^t)$ (or just the position, *i.e.* $p(x_{t+\tau}|\mathbf{y}^t)$). This is a multi-modal distribution and the integration needed to compute it is intractable. However, state trajectories $\hat{\mathbf{x}}_t^{t+\tau}, \hat{\mathbf{g}}_t^{t+\tau}$ can readily be sampled as described in Alg. 3, and can be used to approximate $p(x_{t+\tau}|\mathbf{y}^t)$. As a byproduct it allows us to compute an "occupancy map", which gives a probability that a region $r \subset \mathbb{R}^2$ (a "cell" in a discretization of $\mathbb{R}^2$) is occupied between t and $t+\tau$:

$$P_{occ}(r) \doteq \mathbb{P}\big(\cup_{k=1}^{\tau} \{x_{t+k} \in r\}\big) \tag{5.4}$$

This quantity enables visualizing future trajectories by discounting time.

5.2.2 Goals and environment context

In this section we describe how to model the *intent* of an agent as a solution to a planning problem. As noted above, the "rational" navigation is abstracted into π, which decomposes into independent functions $\{\pi_{\mathbf{g}}\}_{\mathbf{g}\in\mathbf{G}}$. Each is defined by $\pi_{\mathbf{g}} : \mathbb{R}^2 \to \mathbb{S}$, with $\mathbb{S}$ being the action space. This function specifies the optimal "move direction" for reaching the goal $\mathbf{g}$ quickly, while satisfying environment constraints and pedestrian's contextual preferences. Specifically, it is a solution to the optimal control problem, which can be formulated as a Markov decision process (MDP) [Put94], as follows:

$$\pi_{\mathbf{g}}^* = \arg\max_{\pi} \sum_{t=0}^{\infty} \gamma^t R_{\mathbf{g}}(x_t, \pi(x_t)) \tag{5.5}$$

$$\text{subject to } x_{t+1} = x_t + \pi(x_t) \tag{5.6}$$

where the objective is the sum of γ-discounted rewards $R_{\mathbf{g}}$ for being at location x and applying action $\pi(x)$. In our case, the transition dynamics given by (5.6) are deterministic; in the general MDPs they can be stochastic, as we will also explore in Sec. 5.2.3. This sum in the objective is also denoted by $V^{\pi_{\mathbf{g}}}(x_0)$ – the *value* of starting at x_0 and applying policy $\pi_{\mathbf{g}}$ thereafter. Using the

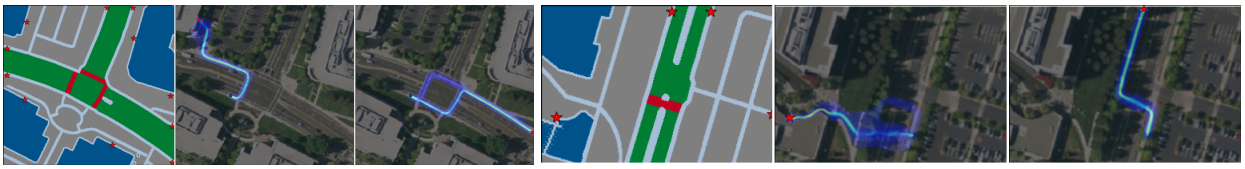


Figure 5.2: *Semantic map, goals, and shortest paths toward them in two regions. The semantic caterories are:* ≐*"building",* ≐*"grass",* ≐*"road",* ≐*"sidewalk",* ≐*"crosswalk". Pedestrian preferences are taken to be:* $\geq \geq \geq \geq$ *. The goals are marked with red "⋆". Stochastic shortest paths toward sample goals are shown in the same two areas, overlaid on the satellite image. Pedestrian is marked with "∘" and the destination – with "⋆".*

value function, it becomes possible to rewrite the optimization problem recursively as

$$\begin{cases} V^{\pi_{\mathbf{g}}}(x) & = \max_u Q^{\pi_{\mathbf{g}}}(x,u) \\ Q^{\pi_{\mathbf{g}}}(x,u) & = R_{\mathbf{g}}(x,u) + \gamma V^{\pi}(x + \pi_{\mathbf{g}}(x)). \end{cases} \tag{5.7}$$

Within this formalism, the optimal policy attains the maximum of Q at every x, that is, $\pi^*_{\mathbf{g}}(x) = \arg\max_u Q^{\pi^*_{\mathbf{g}}}(x,u)$. To fully describe the MDP, we need to specify $R_{\mathbf{g}}$, and to do that we leverage availability of the *semantic map*. Each point in $\mathbb{R}^2$ is classified in one of several categories (namely, "building", "sidewalk", "crosswalk", "road", "grass"), and the reward $R_{\mathbf{g}}$ is taken to be a function of the environment category. Specifically, $R_{\mathbf{g}}(x,u) = \theta^T \phi(x)$ where θ is a vector of preferences and $\phi(x)$ is a vector specifying probabilities of different semantic categories at x. The reward is low in regions corresponding to buildings (since those are "obstacles"), high on sidewalks and crosswalks, and is lower on roads. Moreover, for $x \in \mathbf{g}$, $R_{\mathbf{g}}(x,u) = 0$ and elsewhere $R_{\mathbf{g}}(x,u) < 0$, which implies that the optimal policy $\pi^*_{\mathbf{g}}$, the solution to (5.5) describes a shortest path to $\mathbf{g}$. We show a region of the world with overlaid semantic categories and associated reward preferences in Fig. 5.2.

Commonly, one solves an MDP to obtain a deterministic optimal policy. However, for prediction, it is advantageous to use a stochastic, possibly suboptimal policy. This is because whenever shortest paths to a certain $\mathbf{g}$ are not unique, nonzero probability should be assigned to each. Moreover, an optimal policy assumes that pedestrians are completely rational and assigns zero probability to any deviations from the optimal trajectory. To allow small deviations from optimality, we

use a stochastic Boltzmann policy, also used in [BST09, ZRG09, KZB12]:

$$u \sim \pi_{\mathbf{g}}(x) \text{ w.p. } \propto \exp\left(\alpha(Q_g(x,u) - V_g(x))\right) \tag{5.8}$$

As $\alpha \to \infty$, the policy assigns nonzero probability only to the optimal actions: $u \sim \pi_{\mathbf{g}}(x)$ w.p. $\propto \mathbf{1}\{u \in \arg\max_{\tilde{u}} Q_{\mathbf{g}}(x,\tilde{u})\}$. As $\alpha \to 0$, the policy becomes completely random: $u \sim \pi_{\mathbf{g}}(x)$ w.p. $\frac{1}{|\mathcal{U}|}$. Fig. 5.2 shows several stochastic shortest paths for different goals; note that the shortest path is not necessarily unique.

5.2.3 Environment dynamics

Accurately predicting pedestrian behavior is particularly important near intersections. The framework outlined above inadequately describes this behavior, as it does not account for traffic signals, the simplest instance of environment dynamics. In this section, we describe a simple extension to the framework that accurately models pedestrians waiting for the light signal near intersections. To achieve this, we include the state of the traffic signal into the policy π.

A traffic signal can itself be represented by a dynamical system. The output is one of four values $\{0,1,2,3\}$ (or $\{$$\}$. The four states correspond to the most typical signal configuration with red/green in either direction (2 states) and red/yellow in either direction (2 additional states). Extensions to multi-way signals are possible but not addressed here. If $\mathcal{T}_0$,$\mathcal{T}_1$,$\mathcal{T}_2$,$\mathcal{T}_3$ are the durations that each output is observed, and $\sum_{i=0}^{3} \mathcal{T}_i = \mathcal{T}$ is the period, then the state dynamics is a timer: $c_{t+1} = \text{mod}(c_t + 1, \mathcal{T})$. This model is deterministic and neglects feedback, due for instance to loop sensors detecting proximity of vehicles, including the own-vehicle. Nevertheless, even for this simplified model, directly adding c_t into π would yield a dramatic increase in state-space dimension, since $\mathcal{T}$ is typically very large, and since the pedestrian must operate on the belief about c_t, as it is not directly observed. We instead propose a simplified, "compressed", and fully-observable model where $c_t \in \{0,1,2,3\}$ and dynamics are stochastic:

$$p(c_{t+1} = j | c_t = i) = \begin{cases} 1 - \frac{1}{T_i} & \text{if } j = i \\ \frac{1}{T_i} & \text{if } j = \text{mod}(i+1, 4) \end{cases} \tag{5.9}$$

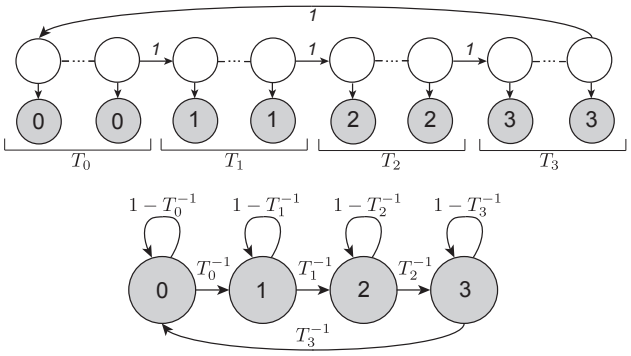


Figure 5.3: Left*: original representation of traffic light as a hidden Markov chain.* Right*: our compressed representation. Shaded nodes are observed, light nodes are hidden.*

The diagram in Fig. 5.3 illustrates the compression: effectively, the multinomial state transition distribution is replaced by a geometric one [DBP05]. To include c_t into the model of pedestrian behavior, we augment $\mathbf{x}_t$ with c_t (so that $\mathbf{x}_t = (x_t, s_t, \theta_t, c_t)$) and modify $\{\pi_{\mathbf{g}}\}_{\mathbf{g}\in\mathbf{G}}$ to depend on c_t in addition to x_t. Furthermore, we modify the reward $R_{\mathbf{g}}$ to be $R_{\mathbf{g}}(x, c, u) = \theta_c^T \phi(x)$; in other words we modify the vector of preferences to be dependent on the traffic light state. Specifically, the state of the signal affects the reward at the crosswalks: if the signal is "red", the cost of being in the region is high, and if the signal is "green", the cost is low. The net effect of these changes is that shortest paths on the modified MDPs involve the pedestrian waiting at intersections for the appropriate light signal, and avoiding moving in traffic. We assume that traffic light state is directly observed, i.e. $\mathbf{y}_t = (y_{1,t}, y_{2,t})$ with $y_{1,t} = \mathsf{R}(\theta_{\text{ego},t})^T(x_t - x_{\text{ego},t}) + n_t$ (as before), and $y_{2,t} = c_t$. In Sec. 5.3.4 we show an example of how this simple extension improves prediction accuracy near the intersections.

5.2.4 Pedestrian orientation

The model described above uses pedestrian *motion* to estimate the latent goal, and when the pedestrian is initially detected, the distribution over $\mathbf{g}$ is uniform. However, since a pedestrian does not typically walk backwards, we can leverage its orientation to infer which goals are more likely at the first detection time. We do this by incorporating an additional measurement of the pedestrian orientation. At each time we observe $\mathbf{y}_t = (y_{1,t}, y_{2,t})$ with $y_{1,t} = \mathsf{R}(\theta_{\text{ego},t})^T(x_t - x_{\text{ego},t}) + n_{1,t}$ and $y_{2,t} = \phi(\theta_t - \theta_{\text{ego},t} + n_{2,t})$, where $\phi(x) = \mathbf{floor}\left(\frac{N}{2\pi}x\right)$ models the output of a pose detector

quantized to N orientation values. The uncertainty of the pose detector response is accounted for by $n_{2,t}$. The additional measurement makes it possible to quickly reduce uncertainty over goals, yielding a more accurate prediction earlier. We show an example of this in Sec. 5.3.5. This improvement is especially important in situations when the pedestrian is tracked only for a small number of frames and an early prediction is required.

5.3 Experiments

We compare against the following several baselines:

$$\text{(RW) } x_{t+1} = x_t + n_t,\ n_t \sim \mathcal{N}(0, \sigma^2) \tag{5.10}$$

$$\text{(LM) } \begin{cases} x_{t+1} & = x_t + v_t \\ v_t & = v_t + n_t,\ n_t \sim \mathcal{N}(0, \sigma^2) \end{cases} \tag{5.11}$$

$$\text{(MM) } x_{t+1} = x_t + u_t,\ u_t \sim p(u|x_t) \tag{5.12}$$

$$\text{(DM) } x_{t+1} = x_t + \pi(x_t, \mathbf{g}, n_t) \tag{5.13}$$

"RW" is a simple random-walk. "LM" is a linear "constant-velocity" model commonly used for tracking, and used in comparisons in [SG13]. "MM" ("Markov model") learns a location-dependent probability distribution over actions, but does not reason about "goals", used in [KZB12]. "DM" ("discrete model") is described in [BST09, ZRG09, KZB12], and is similar to ours, except that it assumes a fixed goal and constant-length steps – it does not adapt to the observations of the pedestrian speed. In our experiments, step lengths for MM and DM were taken to be the average pedestrian speed in the training set.

In sections 5.3.2-5.3.3, we evaluate our "basic" model that does not use the orientation measurement. For this reason, θ can be omitted from the state in (5.2), and the model can be simplified

to:

$$\begin{cases} x_{t+1} & = x_t + s_t \begin{bmatrix} \cos(\theta_t) \\ \sin(\theta_t) \end{bmatrix}, \ \theta_t \doteq \pi(x_t, \mathbf{g}_t, w_{\pi,t}) \\ s_{t+1} & = s_t + w_{s,t} \\ \mathbf{g}_{t+1} & = \mathbf{g}_t \oplus w_{g,t} \end{cases} \tag{5.14}$$

To perform inference with a RBPF, in this case, we approximated $\{p(\mathbf{x}_t|\mathbf{y}^t, \mathbf{g}_t = i)\}_{i=1,\dots,|\mathbf{G}|}$ using a set of Kalman filters. This is possible if one assumes that $\nabla_x \pi(x_t, \mathbf{g}_t, w_{\pi,t}) = 0$, and that the constraint $s_t \in \mathbb{R}_+$ can be dealt with by projecting the variable onto the feasible set after each update. The orientation-aware model is evaluated in Sec. 5.3.5. There, the system dynamics are nonlinear, and to approximate $\{p(\mathbf{x}_t|\mathbf{y}^t, \mathbf{g}_t = i)\}_{i=1,\dots,|\mathbf{G}|}$ we used a set of particle filters.

Error metrics: The difficulty of prediction depends both on prediction horizon and on the period of observation (since some time is required to infer the pedestrian location and the goal). A natural measure that evaluates the model's ability is the expected ℓ_2 error between the model's predictions and the actual pedestrian locations (denoted $\{x_t^{obs}\}_{t=1}^T$), for a given prediction horizon τ and observed sequence length t:

$$\mathcal{E}(\tau, t) \doteq \mathbb{E}_{p(x_{t+\tau}|\mathbf{y}^t)}\big[\|x_{t+\tau} - x_{t+\tau}^{obs}\|\big]. \tag{5.15}$$

This quantity depends on two variables and makes it inconvenient to perform comparisons. A summary error for a given prediction horizon can be computed as the average error over all observation periods, as $\mathcal{E}(\tau) \doteq \frac{1}{T-\tau} \sum_{t=1}^{T-\tau} \mathcal{E}(\tau, t)$.

5.3.1 Dataset and implementation details

To verify the effectiveness of our model, we captured a dataset on a vehicle that is equipped with two cameras, LiDAR, and DGPS. The data streams are synchronized with best-effort which produces satisfactory data batches for the long-term prediction task at low vehicle speed. Data from LiDAR and DGPS makes it possible to accurately localize the pedestrians in world coordinates.

Figure 5.4: Left: *Sample frames from the front camera, with LiDAR returns (colored by depth) and the detected pedestrians.* Right: *Bird's-eye-view of the scene. The vehicle is in north-west corner and pedestrians, shown by circles, are in the center. LiDAR returns are shown in black. Note that the satellite image is outdated and does not contain the pedestrian crossing seen in the images.*

Sample data sequences acquired by the vehicle is shown in Fig. 5.4. For our experiments, we manually annotated 17 video sequences, ranging from 30 to 900 frames, and containing 67 pedestrian trajectories, used in "fully-observable" experiment. In "partially-observable" experiment, pedestrians in those videos were automatically detected and tracked; we used a total of 50 of the resulting trajectories (some were lost due to poor detections). In addition to this dataset, we evaluated our approach on a subset of videos from the KITTI benchmark[GLS13].

For learning model parameters, we used 80/20% train/test splits and four-fold cross-validation. Pedestrian detection was performed using a cascade classifier [ZYC06] using HoG [DT05] features. The location of the detections with respect to the vehicle is detemined by clustering the LiDAR measurements that are projected into the detection window. Once the relative position of the target object is known, its world-coordinates can be found by addition of the vehicle location given by the DGPS. In the experiment reported in Sec. 5.3.4, the traffic light behavior was manually annotated.

Goals, MDPs, and rewards: Destination points were placed near the map boundaries and near building entrances (due to lack of a large enough dataset to learn them); each map region contained less than 15 goals. We used standard value iteration to solve MDPs and produce policies π. Note that MDPs are typically formulated and solved in discrete spaces, but throughout the chapter we treated the MDP state-space (and action-space) as being continuous (i.e. $\mathbb{R}^2$ and $\mathbb{S}$). Since the

state space is fairly low-dimensional, we can solve the discretized MDP and perform interpolation thereafter: at test-time π is bilinearly interpolated from the state's nearest neighbors. We discretize $\mathbb{S}$ to 16-25 directions. To specify the reward in MDPs, formally, one should learn the preference vector θ in $R_{\mathbf{g}}$ using a large enough dataset. We found that the order relations between the cost of each category inferred via IRL [ZMB08] matches the obvious ones appointed manually.

Error computation: At very short observation periods, the state estimate (pedestrian location, velocity, goal) and the predictions are unreliable, and do not accurately reflect the model performance. For this reason, in computing the prediction errors for all models, we ignored all errors with observation period $t < 10$. The error during this "initialization stage" is shown in the error surfaces $\mathcal{E}(t, \tau)$. In all experiments we used 5000 samples to approximate the predictive $p(\mathbf{x}_{t+\tau}|y^t)$ and to calculate prediction errors, for $\tau = 1, \ldots, 350$ (about 23 seconds).

5.3.2 Prediction in the perfectly observed case

We first evaluated our framework on manually annotated "ground-truth" trajectories. In this case, the measurement noise $n_t = 0$, and observations fully disclose pedestrian locations. Quantitative performance results are summarized in Fig. 5.7, where we show prediction errors for up to 350 steps. As is well known, MM degenerates to random walk and is often unable to generate meaningful predictions, similarly to RW. LM accurately predicts short-term behavior but cannot predict inevitable direction changes. These baselines do not utilize environment semantics, which strongly affect behavior of traffic participants. DM generates meaningful predictions, but ones that are not temporally accurate, since the model does not estimate pedestrian speed. In particular, the predictions are erroneous when the pedestrian is stationary (e.g. waiting for the signal at the intersection) and the model predicts motion. Our model utilizes environment semantics, estimates the pedestrian speed, and performs best.

5.3.3 Prediction in the imperfectly observed case

We also tested the framework in the partially-observable setting (with $n_t \sim P_Q$), where observations are obtained from noisy detector responses. These results are qualitatively similar and are

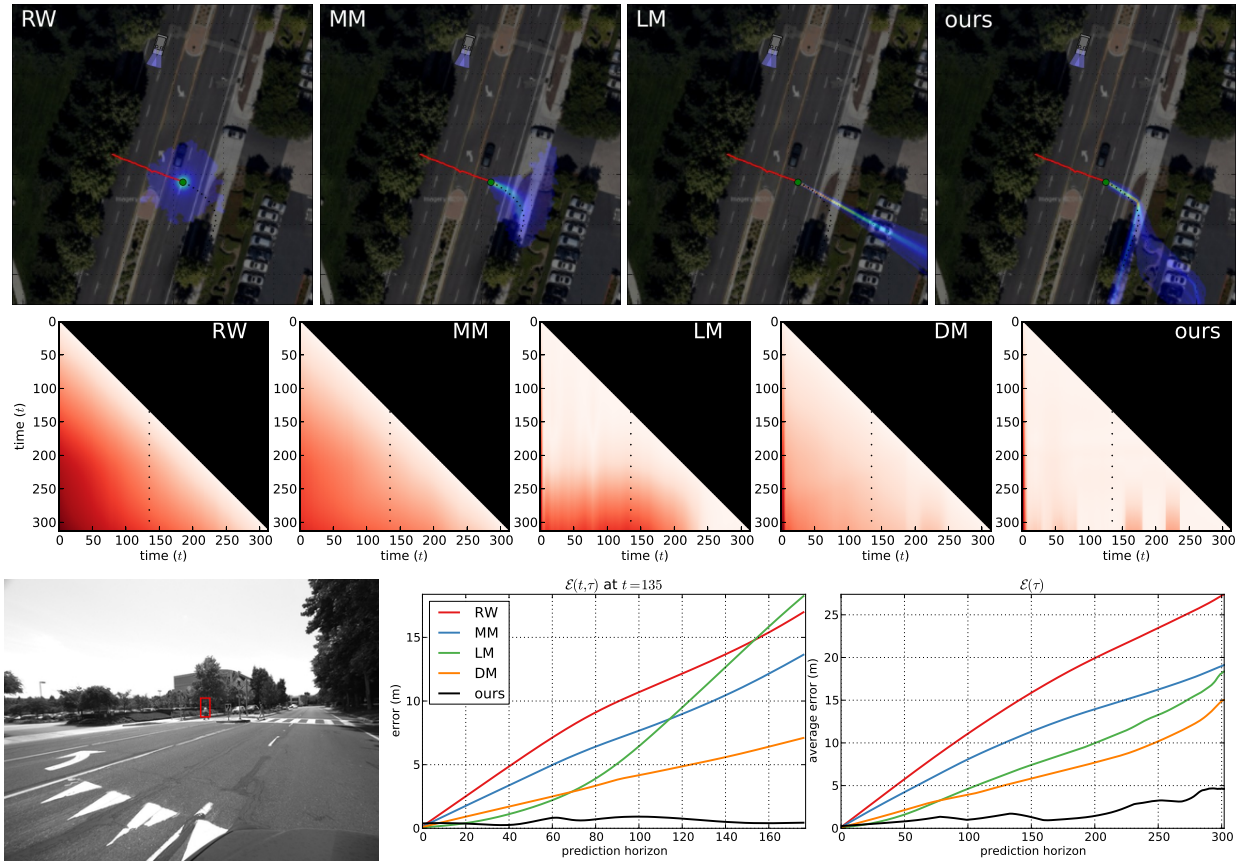


Figure 5.5: Top row: *occupancy maps predicted by the four methods ("DM" not shown, as it generates occupancy map identical to "ours").* Middle row: *quantitative prediction errors. Light colors correspond to small errors, and dark red – to large errors. Time is on the abscissa, and prediction horizon is on the ordinate. The cross-section of the error surface at* $t = 135$ *(indicated by a vertical dashed black line) is shown in the bottom panel.* Bottom row: *frame from the video sequence, and average prediction error* $\mathcal{E}(\tau)$*. Note that in this example, the satellite image used for visualization is outdated and does not contain the crosswalk seen in the sample frame. The crosswalk is present in our semantic map (see Fig. 5.2)*

shown in the same figure. Qualitative example of predictions generated by our model and by the baselines is shown in Fig. 5.5 and 5.6. In that figure DM is not shown, as it generates an occupancy map nearly identical to ours. Unfortunately, its error is high due to predicting motion at the incorrect speed.

5.3.4 Intent near intersections

We compared the basic framework with the traffic-aware extension described in Sec. 5.2.3. The extended framework is expected to reduce the probability that a pedestrian violates traffic rules

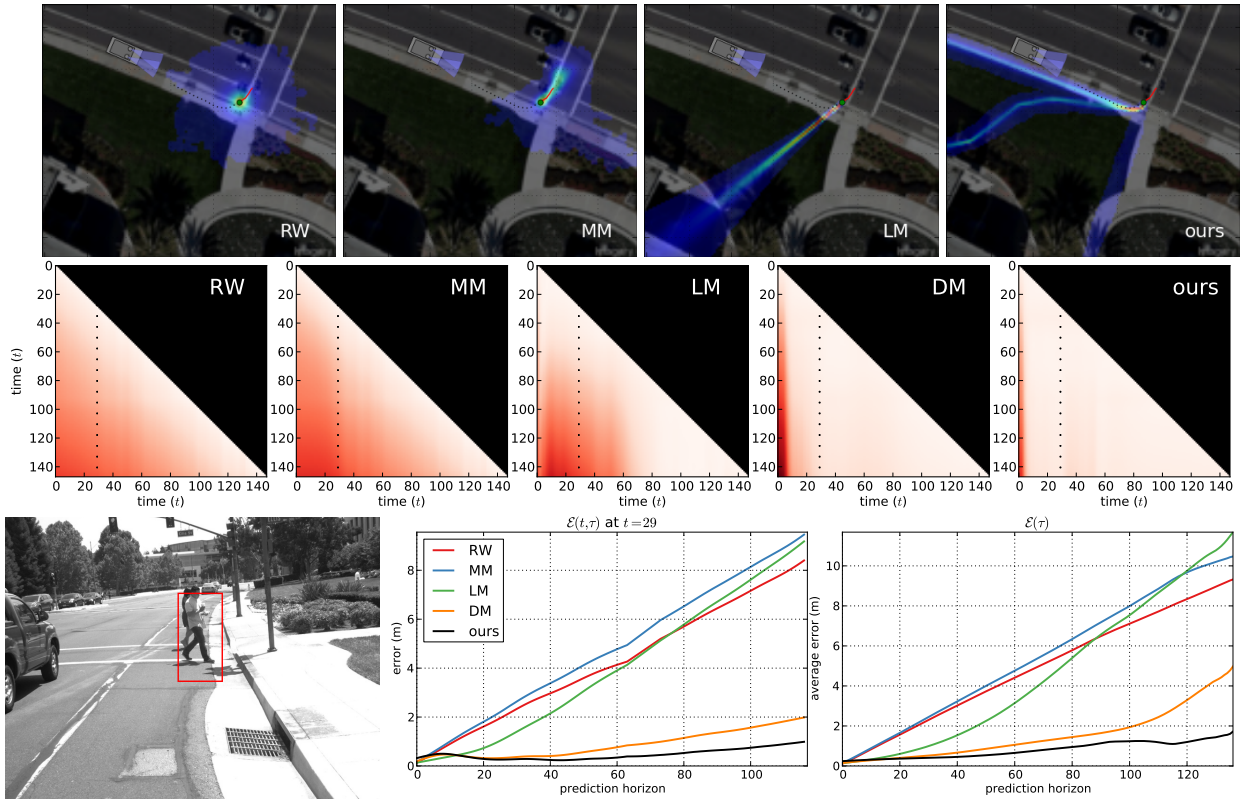


Figure 5.6: *Additional prediction example.* Top row*: predicted occupancy maps generated by the four methods. The past observed locations are shown in red and the future trajectory – in black. Our model correctly predicts that the pedestrian turns towards the sidewalk.* Middle row*: errors for the four methods. The cross-section of the error surface at* $t = 29$ *is shown in the bottom row.* Bottom row: *sample frame with the tracked pedestrian, prediction error at* $t = 29$*, and the average prediction error. In this example, the pedestrian is moving at the speed close to the fixed step size of DM, so the model performs comparably to ours.*

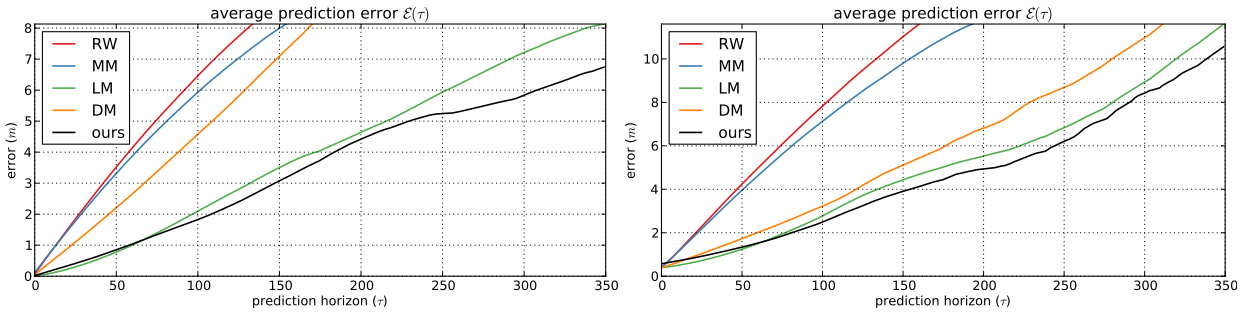


Figure 5.7: *Summary prediction error* $\mathcal{E}(\tau)$ *for all methods, on the entire dataset.* Left: *prediction with ground truth trajectories (perfectly observed case* $n_t = 0$*).* Right: *prediction with noisy trajectories (imperfectly observed case* $n_t \sim P_Q$*)*

(e.g. the probability that she walks on the crosswalk on "red"). An appropriate measure that illustrates the difference between the two models is the probability that the pedestrian is on the

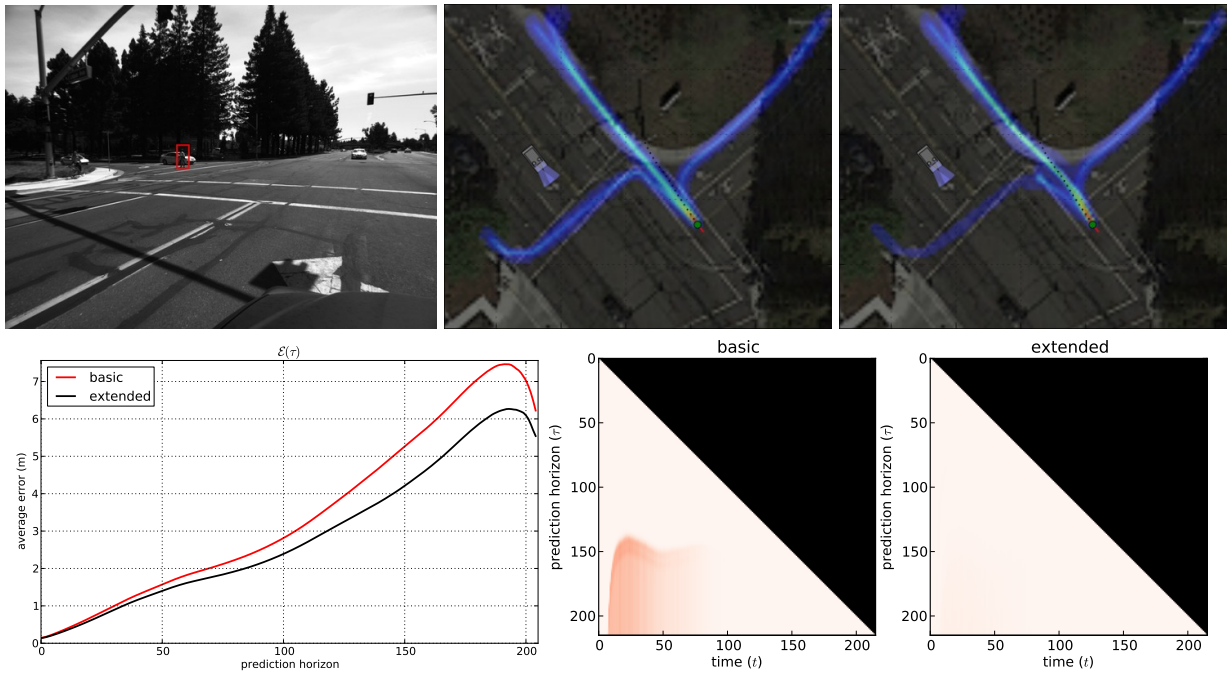


Figure 5.8: Top row*: Sample frame, and the trajectories predicted by the "basic" (left) and "extended" (right) models. Notice that the "extended model" assigns a much lower probability to pedestrian making a left turn – which would violate traffic rules.* Bottom row: *prediction error, and the probability of the pedestrian being on the road $P_{road}(\tau, t)$. Note that the model does not predict that the probability of left turn is zero: a nonzero probability is assigned because c_t may switch to "green", in which case a left turn is allowed.*

road at $t + \tau$, given the observations $\mathbf{y}^t$, i.e. $P_{road}(\tau, t) = \mathbb{E}_{p(x_{t+\tau}|\mathbf{y}^t)}[\mathbf{1}\{x_{t+\tau} \in \text{road}\}]$. In Fig. 5.8 we show that the extended model correctly predicts that pedestrian does not walk out on the road (doing so would violate traffic rules), and improves the overall prediction error.

5.3.5 Pedestrian orientation

To leverage orientation, we trained a linear classifier with HoG features that discretized the orientation θ to four values {"left","front","right","back"}, and learned a response likelihood $p(y_2|\theta)$. The improvement due to the additional measurement occurs primarily during the first few detection steps. In Fig. 5.9 we show an example where the orientation-aware model estimates goals faster than the "basic" model, and is able to reduce the prediction error as a result.

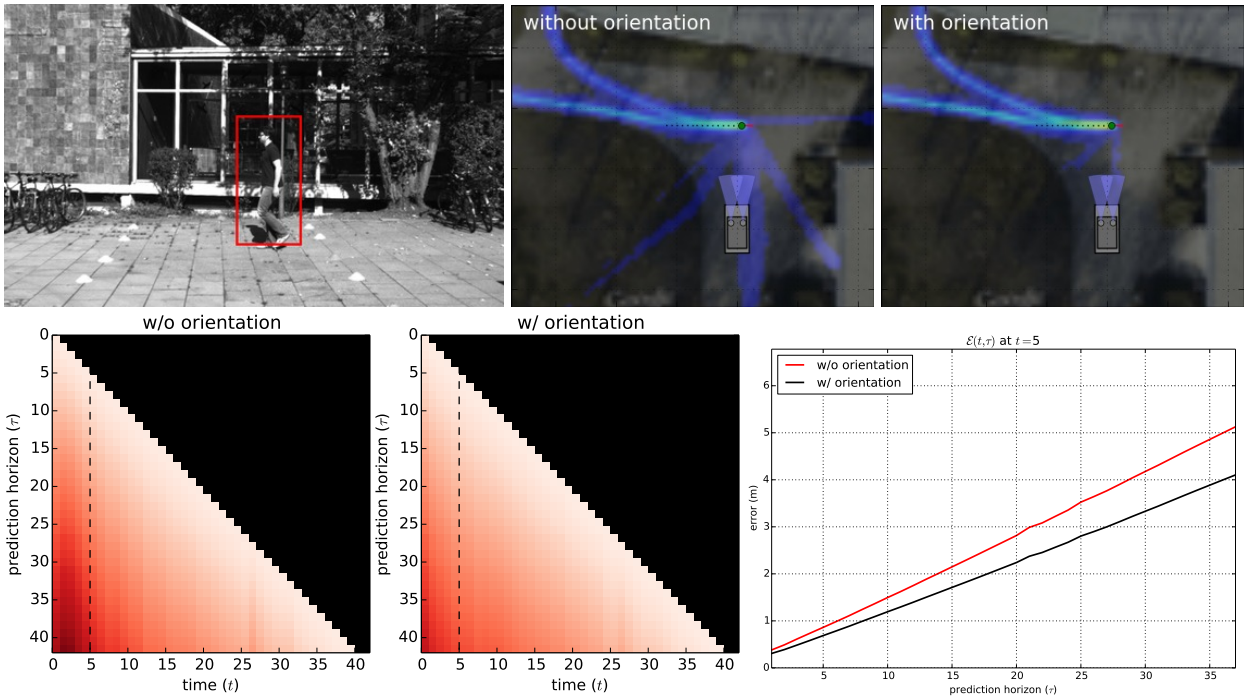


Figure 5.9: Top row*: Sample frame with a tracked pedestrian.* Middle row*: Predictions generated by the orientation-agnostic and orientation-aware models, at* $t = 5$. Bottom row: *Prediction errors* $\mathcal{E}(\tau, t)$*; the plot on the right is the cross-section at* $t = 5$ *(shown with dashed line).*

5.4 Discussion

Our method for long-term prediction of pedestrians has several limitations. We discard feedback and interactions, due to the influence of agents on each other and by the ego-vehicle, including feedback by the environment, *e.g.* adaptive traffic lights, and changes in the environment, *e.g.* parked cars. These could be incorporated at a significant increase in computational cost that would prevent real-time operation. We also assume rational agent behavior, necessary to even define a predictive strategy, which means that we cannot predict *rare* events such as a pedestrian suddenly jumping onto the road. This limitation is shared with the majority of existing methods.

CHAPTER 6

Discussion

This thesis focused on algorithms that enable visual systems to autonomously interact with the environment. We made no attempt to demonstrate an end-to-end version of such system, and completely avoided discussions about many of its system engineering aspects; instead the focus was on individual components. Speficically, we focused on topics in areas of perception (Chapter 2-3), information acquisition (Chapter 3-4), and prediction (Chapter 5).

We demonstrated a causal approach for object discovery and segmentation in video in Chapter 2. We found that it performs competitively to state-of-the-art *non-causal* approaches that require whole video for processing. This was made possible by relying on instantaneous occlusions as evidence for presence of objects. Occlusions provided local depth-ordering constraints, which were combined into a global depth order and object labels by solving a large scale convex problem.

Video segmentation was augmented with semantic category labels in Chapter 3. There our focus was on leveraging redundancy in video to improve efficiency; to perform semantic video segmentation "on a budget". We developed an "attention" mechanism – an online strategy that evaluates uncertainty in video annotation and selects which region of a frame to submit to a battery of detectors. This approach was formulated within the sequential decision making framework, and could be viewed as an application of information-gathering control in data-space.

In Chapter 4 we turned toward control in physical space. We argued that to reliably perform visual decision tasks, one needs the ability to resolve ambiguity due to occlusions and quantization-scale phenomena. We demonstrated that this is possible if one has mobility and control. As in the preceeding chapter, we used the sequential decision making framework to formulate an optimal control problem. Complexity of solving this problem turned out to be exponential in the horizon, rendering non-greedy approaches impractical; we showed that a simple, greedy heuristic achieves

an acceptable performance.

Control in dynamic environments is more involved. In Chapter 5 we considered environments populated by other mobile intelligent entities. There, the basic requirement of collision avoidance forces us to make predictions about how others will behave and to accordingly modify strategy. We brushed off the control problem, and focused on describing an approach for long-term prediction that uses "inverse optimal control" – recognition which of the finitely many possible optimal strategies an agent is executing.

REFERENCES

[ADF10] B. Alexe, T. Deselaers, and V. Ferrari. "What is an object?" In *CVPR*, 2010.

[AHT12] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. "Searching for objects driven by context." In *NIPS*, 2012.

[AMF11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. "Contour Detection and Hierarchical Image Segmentation." *TPAMI*, **33**(5), 2011.

[ApS15] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB.*, 2015.

[ARS12] A. Ayvaci, M. Raptis, and S. Soatto. "Sparse Occlusion Detection with Optical Flow." *IJCV*, **97**(3), 2012.

[AS12] A. Ayvaci and S. Soatto. "Detachable Object Detection: Segmentation and Depth Ordering from Short-Baseline Video." *PAMI*, **34**(10), 2012.

[AT09] A. Andreopoulos and J. K. Tsotsos. "A theory of active object localization." In *ICCV*, 2009.

[AYR14] M. R. Amer, S. Yousefi, R. Raich, and S. Todorovic. "Monocular Extraction of 2.1D Sketch Using Constrained Convex Optimization." *IJCV*, 2014.

[Baj88] R. Bajcsy. "Active Perception." **76**(8), 1988.

[Bal91] D. H. Ballard. "Animate Vision." *Artificial Intelligence*, **48**(1), 1991.

[BCH12] T. Bandyopadhyay, Z. J. Chong, D. Hsu, M. Ang, D. Rus, and E. Frazzoli. "Intention-Aware Pedestrian Avoidance." In *International Symposium on Experimental Robotics (ISER)*, 2012.

[BE99] G. J. Brostow and I. Essa. "Motion based decompositing of video." In *ICCV*, 1999.

[BGF04] F. Bourgault, A. Göktogan, T. Furukawa, and H. F. Durrant-Whyte. "Coordinated search for a lost target in a Bayesian world." *Advanced Robotics*, **18**(10), 2004.

[Bit84] R. Bitmead. "Persistence of excitation conditions and the convergence of adaptive schemes." *IEEE Transactions on Information Theory*, **30**(2), 1984.

[BM98] L. Bergen and F. Meyer. "Motion Segmentation and Depth Ordering Based on Morphological Segmentation." In *ECCV*, 1998.

[BM10] T. Brox and J. Malik. "Object segmentation by long term analysis of point trajectories." In *ECCV*, 2010.

[BS09] A. Bissacco and S. Soatto. "Hybrid dynamical models of human motion for the recognition of human gaits." *IJCV*, 2009.

[BST09] C. L. Baker, R. Saxe, and J. B. Tenenbaum. "Action understanding as inverse planning." *Cognition*, 2009.

[BT09] W. Brendel and S. Todorovic. "Video object segmentation by tracking regions." In *ICCV*, 2009.

[BV04] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BWS09] X. Bai, J. Wang, D. Simons, and G. Sapiro. "Video snapcut: robust video object cutout using localized classifiers." In *ACM Transactions on Graphics (TOG)*, 2009.

[CF13] J. Chang and J. W. Fisher III. "Topology-Constrained Layered Tracking with Latent Flow." In *ICCV*, 2013.

[CLT10] M. J. Choi, J.J. Lim, A. Torralba, and A.S. Willsky. "Exploiting hierarchical context on a large database of object categories." In *CVPR*, 2010.

[CP11] A. Chambolle and T. Pock. "A first-order primal-dual algorithm for convex problems with applications to imaging." *Journal of Mathematical Imaging and Vision*, **40**(1), 2011.

[CPS13] W. Choi, C. Pantofaru, and S. Savarese. "A General Framework for Tracking Multiple People from a Moving Camera." *TPAMI*, 2013.

[CSD08] J. J. Corso, E. Sharon, S. Dube, S. El-Saden, U. Sinha, and A. Yuille. "Efficient Multilevel Brain Tumor Segmentation with Integrated Bayesian Model Classification." *IEEE Transactions on Medical Imaging*, **27**(5), 2008.

[CSM14] Y. Chen, H. Shioi, C. F. Montesinos, L. P. Koh, S. Wich, and A. Krause. "Active Detection via Adaptive Submodularity." In *ICML*, 2014.

[CT91] T. M. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[CWI13] J. Chang, D. Wei, and J. W. Fisher III. "A Video Representation Using Temporal Superpixels." In *CVPR*, 2013.

[DBP05] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. "Activity recognition and abnormality detection with the switching hidden semi-markov model." In *CVPR*, 2005.

[DDM00] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. "Rao-Blackwellised particle filtering for dynamic Bayesian networks." In *UAI*, 2000.

[DP91] T. Darrell and A. Pentland. "Robust estimation of a multi-layered motion representation." In *IEEE Workshop on Visual Motion*, 1991.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection." In *CVPR*, 2005.

[DZ13] P. Dollár and C. L. Zitnick. "Structured Forests for Fast Edge Detection." In *ICCV*, 2013.

[ED04] E. Eisemann and F. Durand. "Flash Photography Enhancement via Intrinsic Relighting." *ACM Trans. Graph.*, **23**(3), 2004.

[ES10] R. Eidenberger and J. Scharinger. "Active perception and scene modeling by planning with probabilistic 6D object poses." In *IROS*, 2010.

[ESR09] D. Ellis, E. Sommerlade, and I. Reid. "Modelling pedestrian trajectory patterns with Gaussian processes." In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 2009.

[EVW] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results.".

[FGM10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. "Object Detection with Discriminatively Trained Part Based Models." *TPAMI*, 2010.

[FT02] A. F. Foka and P. E. Trahanias. "Predictive autonomous robot navigation." In *IROS*, 2002.

[GB14] Michael Grant and Stephen Boyd. "CVX: Matlab Software for Disciplined Convex Programming, version 2.1." `http://cvxr.com/cvx`, March 2014.

[GCS12] F. Galasso, R. Cipolla, and B. Schiele. "Video Segmentation with Superpixels." In *ACCV*, 2012.

[GKH10] M. Grundmann, V. Kwatra, M. Han, and I. Essa. "Efficient Hierarchical Graph Based Video Segmentation." In *CVPR*, 2010.

[GLS13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets Robotics: The KITTI Dataset." *IJRR*, 2013.

[GNC13] F. Galasso, S. Naveen, T. Cardenas, T. Brox, and B. Schiele. "A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis." In *ICCV*, 2013.

[HBH08] M. F. Huber, T. Bailey, Durrant-Whyte H., and U. D. Hanebeck. "On Entropy Approximation for Gaussian Mixture Random Vectors." In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2008.

[HE05] Y. Huang and I. Essa. "Tracking multiple objects through occlusions." In *CVPR*, 2005.

[HHM93] D. Heckerman, E. Horvitz, and B. Middleton. "An Approximate Nonmyopic Computation for Value of Information." *TPAMI*, 1993.

[HK08] G. Heitz and D. Koller. "Learning Spatial Context: Using Stuff to Find Things." In *ECCV*, 2008.

[HMS11] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. "Active Classification: Theory and Application to Underwater Inspection." In *International Symposium on Robotics Research*, 2011.

[HO07] J. R. Hershey and P. A. Olsen. "Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models." *ICASSP*, **4**(6), 2007.

[How66] R. A. Howard. "Information Value Theory." *IEEE Trans. Systems Science and Cybernetics*, 1966.

[HR70] M. E. Hellman and J. Raviv. "Probability of error, equivocation and the Chernoff bound." *IEEE Transactions on Information Theory*, **16**, 1970.

[HT10] G. M. Hoffmann and C. J. Tomlin. "Mobile Sensor Network Control Using Mutual Information Methods and Particle Filters." *IEEE Transactions on Automatic Control*, **55**(1), 2010.

[Jaz70] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Mathematics in science and engineering. 1970.

[JF01] N. Jojic and B. J. Frey. "Learning flexible sprites in video layers." In *CVPR*, 2001.

[JSC11] Z. Jia, A. Saxena, and T. Chen. "Robotic Object Detection: Learning to Improve the Classifiers Using Sparse Graphs for Path Planning." In *IJCAI*, 2011.

[JYS08] J. Jackson, A. J. Yezzi, and S. Soatto. "Dynamic shape and appearance modeling via moving and deforming layers." *IJCV*, 2008.

[KBF12] S. Karayev, T. Baumgartner, M. Fritz, and T. Darrell. "Timely Object Recognition." In *NIPS*, 2012.

[KCF11] M. Krainin, B. Curless, and D. Fox. "Autonomous generation of complete 3D object models using next best view manipulation planning." In *ICRA*, 2011.

[KG05] A. Krause and C. Guestrin. "Near-optimal Nonmyopic Value of Information in Graphical Models." In *Uncertainty in Artificial Intelligence*, 2005.

[KG09] A. Krause and C. Guestrin. "Optimal Value of Information in Graphical Models." *JAIR*, 2009.

[KPP00] H. Kopp-Borotschnig, L. Paletta, M. Prantl, and A. Pinz. "Appearance-based active object recognition." *Image and Vision Computing*, **18**(9), 2000.

[KSF14] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. "Context-Based Pedestrian Path Prediction." In *ECCV*, 2014.

[KTZ08] M. P. Kumar, P. H. S. Torr, and A. Zisserman. "Learning layered motion segmentations of video." *IJCV*, **76**(3), 2008.

[KZB12] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert. "Activity Forecasting." In *ECCV*, 2012.

[LFA08] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. "Human-assisted motion annotation." In *CVPR*, 2008.

[Lin56] D. Lindley. "On the measure of the information provided by an experiment." *Annals of Mathematical Statistics*, 1956.

[Lju97] L. Ljung. *System Identification, Theory for the User*. Prentice Hall, 1997.

[LKG11] Y. J. Lee, J. Kim, and K. Grauman. "Key-segments for video object segmentation." In *ICCV*, 2011.

[MB10] J. Ma and J. W. Burdick. "Dynamic sensor planning with stereo for model identification on a mobile platform." In *ICRA*, 2010.

[MCK14] O. Mac Aodha, N. D. F. Campbell, J. Kautz, and G. J. Brostow. "Hierarchical Subquery Evaluation for Active Learning on a Graph." In *CVPR*, 2014.

[NR00] Andrew Y. Ng and Stuart J. Russell. "Algorithms for Inverse Reinforcement Learning." In *ICML*, 2000.

[NWF78] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. "An analysis of approximations for maximizing submodular set functions." *Mathematical Programming*, 1978.

[OMB14] P. Ochs, J. Malik, and T. Brox. "Segmentation of Moving Objects by Long Term Video Analysis." *TPAMI*, **36**(6), 2014.

[PCC09] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. "A convex relaxation approach for computing minimal partitions." In *CVPR*, 2009.

[PES09] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. "You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking." In *International Conference on Computer Vision*, 2009.

[PF13] A. Papazoglou and V. Ferrari. "Fast object segmentation in unconstrained video." In *ICCV*, 2013.

[Pro08] L. Pronzato. "Optimal experimental design and some related control problems." *Automatica*, **44**, 2008.

[Put94] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

[RA07] D. Ramachandran and E. Amir. "Bayesian inverse reinforcement learning." In *IJCAI*, 2007.

[RGT05] N. Roy, G. Gordon Gordon, and S. Thrun. "Finding Approximate POMDP solutions Through Belief Compression." *JAIR*, **23**, 2005.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. “”GrabCut”: interactive foreground extraction using iterated graph cuts.” *ACM Trans. Graph.*, 2004.

[RM07] X. Ren and J. Malik. “Tracking as Repeated Figure/Ground Segmentation.” In *CVPR*, 2007.

[SBM11a] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. “Occlusion boundary detection and figure/ground assignment from optical flow.” In *CVPR*, 2011.

[SBM11b] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. “Occlusion boundary detection and figure/ground assignment from optical flow.” In *CVPR*, 2011.

[Sch] M. Schmidt. “UGM:MATLAB toolbox for probabilistic undirected graphical models.” www.di.ens.fr/mschmidt/Software/UGM.html.

[SDC04] P. Smith, Tom Drummond, and R. Cipolla. “Layered motion segmentation and depth ordering by tracking edges.” *TPAMI*, **26**(4), 2004.

[SG13] N. Schneider and D. Gavrila. “Pedestrian path prediction with recursive Bayesian filters: A comparative study.” In *Pattern Recognition*. Springer, 2013.

[Soa11] S. Soatto. “Steps Towards a Theory of Visual Information: Active Perception, Signal-to-Symbol Conversion and the Interplay Between Sensing and Control.” *arXiv:1110.2053*, 2011.

[SRB10] D. Sun, S. Roth, and M. J. Black. “Secrets of Optical Flow Estimation and Their Principles.” In *CVPR*, 2010.

[Stu98] J. F. Sturm. “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones.”, 1998.

[SUG11] K. E. A. van de Sande, J.R.R. Uijlings, T Gevers, and A.W.M. Smeulders. “Segmentation as Selective Search for Object Recognition.” In *ICCV*, 2011.

[TAR13] B. Taylor, A. Ayvaci, A. Ravichandran, and S. Soatto. “Semantic Video Segmentation From Occlusion Relations Via Convex Optimization.” In *EMMCVPR*, 2013.

[TKS15] B. Taylor, V. Karasev, and S. Soatto. “Causal video object segmentation from persistence of occlusion relations.” Technical Report 150002, UCLA, CSD, 2015.

[TL13] J. Tighe and S. Lazebnik. “Finding Things: Image Parsing with Regions and Per-Exemplar Detectors.” In *CVPR*, 2013.

[TTT99] K. C. Toh, M.J. Todd, and R. H. Ttnc. “SDPT3 – a MATLAB software package for semidefinite programming.” *Optimization Methods and Software*, **11**, 1999.

[VAP10] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. “Multiple Hypothesis Video Segmentation from Superpixel Flows.” In *ECCV*, 2010.

[VBF12] A. Vezhnevets, J.M. Buhmann, and V. Ferrari. "Active learning for semantic segmentation with expected change." In *CVPR*, 2012.

[VC10] R. Vezzani and R. Cucchiara. "Video Surveillance Online Repository (ViSOR): an integrated framework." *Multimedia Tools Appl.*, 2010.

[VG09] S. Vijayanarasimhan and K. Grauman. "What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations." In *CVPR*, 2009.

[VG12] S. Vijayanarasimhan and K. Grauman. "Active Frame Selection for Label Propagation in Videos." In *ECCV*, 2012.

[VR11] C. Vondrick and D. Ramanan. "Video Annotation and Tracking with Active Learning." In *NIPS*, 2011.

[VSK02] R. Vidal, Omid Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation." *IEEE Transactions on Robotics*, **18**(5), 2002.

[VTS12] L. Valente, R. Tsai, and S. Soatto. "Information Gathering Control via Exploratory Path Planning." In *Proceedings of the Conference on Information Sciences and Systems*. 2012.

[WA94] J. Y. Wang and E. H. Adelson. "Representing moving images with layers." *TIP*, **3**(5), 1994.

[Wer39] M. Wertheimer. *Laws of organization in perceptual forms*. W. D. Ellis, 1939.

[XTZ13] D. Xie, S. Todorovic, and S.C. Zhu. "Inferring "Dark Matter" and "Dark Energy" from Videos." In *ICCV*, 2013.

[XXC12] C. Xu, C. Xiong, and J. Corso. "Streaming Hierarchical Video Segmentation." In *ECCV*, 2012.

[YBO11] K. Yamaguchi, A. Berg, L. Ortiz, and T. Berg. "Who are you with and Where are you going?" In *CVPR*, 2011.

[YFU12] J. Yao, S. Fidler, and R. Urtasun. "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation." In *CVPR*, 2012.

[YGL12] A. Yao, J. Gall, C. Leistner, and L. J. Van Gool. "Interactive object detection." In *CVPR*, 2012.

[ZJS13] D. Zhang, O. Javed, and M. Shah. "Video Object Segmentation through Spatially Accurate and Temporally Dense Extraction of Primary Object Regions." In *CVPR*, 2013.

[ZMB08] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. "Maximum Entropy Inverse Reinforcement Learning." In *UAI*, 2008.

[ZRG09] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. "Planning-based Prediction for Pedestrians." In *IROS*, 2009.

[ZYC06] Q. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. "Fast human detection using a cascade of histograms of oriented gradients." In *CVPR*, 2006.