A General, Context-Aware Pedestrian Trajectory Prediction Model

by

Nikita Jaipuria

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2018

Author ..... Signature redacted .....
Department of Mechanical Engineering
Aug 19, 2018

Certified by ..... Signature redacted .....
Jonathan P. How
Richard C. Maclaurin Professor of Aeronautics and Astronautics
Thesis Supervisor

Read by ..... Signature redacted .....
John Leonard
Samuel C. Collins Professor of Mechanical and Ocean Engineering
Thesis Reader

Accepted by ..... Signature redacted
Rohan Abeyaratne
Quentin Berg Professor of Mechanics
Chairman, Committee on Graduate Students

A General, Context-Aware Pedestrian Trajectory Prediction Model

by

Nikita Jaipuria

Submitted to the Department of Mechanical Engineering
on Aug 19, 2018, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Autonomous driving on highways and freeways, as a feature, is already available in quite a few high-end commercial vehicles being sold today. Autonomous driving in urban environments, on the other hand, is still an active area of academic and industrial research [6], because of its relatively complex nature. Urban driving requires the self-driving vehicle to interact with not just other vehicles, but also other moving agents such as cyclists and pedestrians. Pedestrian trajectory prediction is challenging because of the relatively higher number of degrees of freedom in pedestrian movement and absence of uniform rules across different cities and different scenarios within a city. Furthermore, in scenarios such as intersections, context, such as pedestrian traffic lights, stop signs and sidewalk geometry, significantly influences pedestrian movement. The objective of this thesis is to present a general, context-aware, long term (order of few seconds) trajectory prediction model for pedestrians in urban intersections. To meet this objective, first, the Augmented Semi Nonnegative Sparse Coding (ASNSC) [13] framework, for trajectory prediction, is extended to embed context, and build the Context-aware Augmented Semi Nonnegative Sparse Coding (CASNSC) algorithm. For prediction in new, unseen intersections with different curbside geometries (orthogonal versus skewed), CASNSC is further extended to build the Transferable Augmented Semi Nonnegative Sparse Coding (TASNSC) algorithm. Urban intersections can at times vary significantly in the type of pedestrian behaviors encountered, even across intersections with similar geometries. For instance, faster, rule breaking students near a college campus versus slower pedestrians in a residential area. While TASNSC is capable of successfully transferring knowledge from one intersection to another, it lacks the ability to update its prediction model as, and when, new intersections are visited and novel behaviors are encountered. An online model, based on TASNSC, is also presented in this thesis to account for this particular limitation.

Thesis Supervisor: Jonathan P. How
Title: Richard C. Maclaurin Professor of Aeronautics and Astronautics

Acknowledgments

First and foremost, I would like to thank Prof. Jonathan How for advising me throughout these past two years. Other than bring a brilliant mentor and thinker, he always encouraged me to see connections between my research and others', which would help me advance past any roadblocks I had created for myself in solving a research problem. I am also grateful to Prof. John Leonard for agreeing to be the ME reader for my thesis.

I would also like to specifically thank Golnaz Habibi, who I worked with jointly on the research presented in this thesis, for all the fun, engaging, and often animated research discussions. Other than being a great research partner, she has been a constant source of motivation and friendship to me. Additionally, I would like to thank Michael Everett, for his constant help with the data collection setup and for providing valuable feedback.

I would also like to acknowledge the financial supporters of my graduate education, including the Ford Motor Company and the Denso Corporation. All the research presented in this thesis is funded by a research grant from the Ford Motor Company.

Thank you to all my labmates at ACL, who maintained a warm, collaborative work environment. Dong-Ki Kim, Björn Lütjens and Macheng Shen - for all the fun times we shared together. Chris Fourie - for always making me laugh and feel good about how organized I am, relatively. Erin Evans - for being the funniest, weirdest, and for sure, one of the best people I have known at MIT. And Noam Buckman, for all the jokes and rants we shared together, but most of all, for being an invaluable friend. Outside of ACL, thank you Katie Cavanaugh, for always being there for me, at work, in classes and outside of MIT.

Lastly, a special thank you to my parents, sister and friends back home, who kept me grounded. And to Deepak Pathak, for being my biggest strength and motivator, all through graduate school.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Recent advances in sensor technologies, computing power and publicly available datasets have led to a surge in research on autonomous driving, motivated by various reasons, such as, improving road safety ([24, 3]), reducing traffic congestion ([55, 36]), improving vehicle utilization and also, reducing pollution [53]. Autonomous driving on highways and freeways, as a feature, is already available in quite a few high-end commercial vehicles being sold today. Autonomous driving in urban environments, on the other hand, is still an active area of academic and industrial research [6]. Urban driving is complex because of the huge variety of situations and moving objects that a vehicle may encounter. For safe and efficient autonomous driving in complex urban environments, a self-driving vehicle, in addition to interacting with other vehicles, must be able to interact with other moving objects like pedestrians and cyclists.

Trajectory prediction of pedestrians is challenging as compared to that of other cars and cyclists because of the absence of a regular flow, such as driving within lanes and staying within road boundaries, that results from a fairly uniform set of predefined "rules of the road" for cars (and to some extent cyclists). The complexity is increased further when the urban environment includes pedestrian traffic lights or tightly packed sidewalks with numerous pedestrian interactions. Context, such as pedestrian traffic lights, location of sidewalks and crosswalks, curbside geometry etc., significantly influences pedestrian movement. Fig. 1-1 shows an urban intersection scenario in which pedestrian choice between two crosswalks, at

Figure 1-1: Example intersection scenario. Dotted green line denotes a rectangular approximation to the curbside in view. Orange arrows denote relative distance of a pedestrian from the two curbsides, which can indicate pedestrian intention. Pedestrian traffic light status is highlighted in orange, which also influences pedestrian movement.

an intersection corner, is influenced by the status of pedestrian traffic lights for each of those crosswalks. Similarly, a comparison of the relative distance of the pedestrian to each curbside, can also be indicative of future direction of motion. A context-aware prediction model, that can capture such features, would be able to better infer pedestrian intent, and hence, have improved trajectory prediction accuracies as compared to a context-unaware prediction model, that is based on spatial features alone, such as pedestrian position and orientation.

Prior work on trajectory prediction of moving agents has focused on two main approaches [39]: prototype trajectories-based and maneuver intention estimation-based. Chen et al. [13] combined the two approaches, to inherit the benefits of both, in developing the Augmented Semi Nonnegative Sparse Coding (ASNSC) algorithm. Trajectory prediction using ASNSC comprises of learning a set of motion primitives and the pair-wise transition between them, as opposed to learning full trajectory prototypes. Such an approach addresses the issue of partial observability of trajectories caused by occlusions or a limited field of view of on-board perception sensors. ASNSC showed significant improvement in pedestrian trajectory prediction over state-of-art clustering based approach using Dirichlet Process mixture of Gaussian Process (DPGP). However, ASNSC learns from spatial features alone

and fails to incorporate available context. The accuracy of predictions using ASNSC can be improved by utilizing semantic features from the environment in the learning process.

Most of the previous work on context-based pedestrian trajectory prediction aims to identify stopping versus crossing intent ([49, 37, 58]), as opposed to long term trajectory prediction. The latter can can provide additional, useful information to the self-driving vehicle for planning its future course of action. Such as how much time would it take for someone to cross the road, which crosswalk/path would be used, etc. In addition to this limitation, some prior works also assume that only one context feature can be active at a time [28], which works for short-term, immediate prediction only. More recently, Schulz et al. [50] utilized head pose to predict future pedestrian trajectories for upto one second. Karasev et al. [34] successfully embedded semantic features such as traffic lights and crosswalks into their long-term pedestrian trajectory prediction model. However, the output of their model is an *occupancy map* of feasible trajectory predictions, as opposed to actual future trajectories, which can provide additional, useful information to the vehicle planner.

The first contribution of this thesis to lay down the framework for the Context-aware Augmented Semi Nonnegative Sparse Coding (CASNSC) algorithm. CASNSC is a novel, context-aware trajectory prediction model, applicable to pedestrians in intersection corners. It can predict for a long-term horizon of about 5 seconds and outputs a set of future trajectories along with the likelihood of each. Such an output is desired and can be easily incorporated by state-of-art probabilistic planners ([35, 17, 10, 11]).

As the name suggests, CASNSC is built on the ASNSC framework. First, a dictionary of motion primitives is learned using ASNSC, from spatial features such as pedestrian position and orientation in the car frame. Context affects the probability of transition between learned motion primitives. For instance, in Fig. 1-1, pedestrian traffic lights influence the probability of transition between motion primitives at the shown intersection corner. For the pedestrian in focus, the pedestrian traffic light for the crosswalk in front of him is red. He can either wait for the light to turn green or turn left and use the other crosswalk. In this situation,

transitioning from the pedestrian's current motion primitive, to one that represents moving straight ahead, would have a lower probability than transitioning to another motion primitive, which represents turning left. This aspect was not captured in ASNSC as the transition between motion primitives was modeled using the same set of spatial features (pedestrian position and orientation in the car frame) as that used for learning the motion primitives themselves. Any context, that may influence a pedestrian's intent and hence, transitions between motion primitives, was ignored. CASNSC addresses this shortcoming by using a combination of context features and spatial features to model the pair-wise transition between motion primitives.

Incorporating context, in addition to improving trajectory prediction accuracies, can also provide flexibility of application of the learned model to prediction in new, but similar environments, unexplored earlier. Data collection in the real world is expensive and time consuming. A model that can learn solely from context features, would be far superior to one that requires learning from spatial features, and hence, needs to be trained on every intersection.

The selection of right context features is also important to avoid training in every intersection. For instance, the use of context features like orthogonal distance to curbside makes the intent prediction models in prior works ([59, 37, 58]) dependent on the specific training intersection geometry. This prevents generalization of such models to prediction in new intersections with varying curbside and/or sidewalk geometries. There is a need for a general, transferable trajectory prediction algorithm, which when trained on one intersection, can be used for prediction in new, unseen intersections, with similar context but varying curbside and sidewalk geometries.

Ballan et al. [4] and Sadeghian et al. [48] demonstrate the ability to "transfer knowledge" by predicting in unseen locations with similar semantic elements. However, both approaches require a prior bird's eye view of the scene, in the form of high-definition prior maps, to compute semantic similarities between the test and train environment. Such priors are

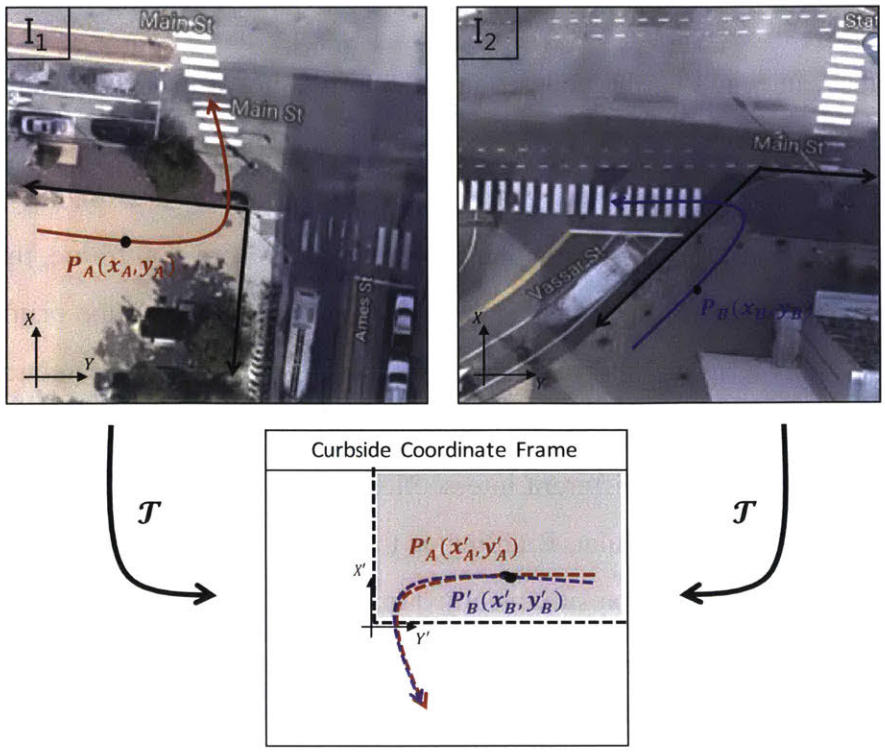


Figure 1-2: An illustration to show how points $P_A(x_A, y_A)$ on the red trajectory in intersection $\mathbf{I_1}$ and $P_B(x_B, y_B)$ on the purple trajectory in intersection $\mathbf{I_2}$, under the transformation $\mathscr{T}$, map to points $P'_A(x'_A, y'_A)$ and $P'_B(x'_B, y'_B)$ in the "curbside coordinate frame". In this work, $\mathscr{T}$ is defined such that it is an affine transformation. Pedestrian trajectories in urban intersections are significantly constrained by curbsides. Transforming trajectories into the curbside coordinate frame, using an affine transformation, intuitively would map trajectories with similar pedestrian intent approximately on top of each other. This insight helps in developing a general, transferable pedestrian trajectory prediction model.

expensive to create and maintain.

The second contribution of this thesis is the Transferable Augmented Semi Nonnegative Sparse Coding (TASNSC) algorithm. TASNSC encodes situational context and provides a transferable prediction model, which can be generalized to predict in corners of new, unseen intersections, without needing a high-definition prior map. The key idea that enables this generalization is the use of a simple prior on curbside geometry (i.e. angle made by intersecting curbs at the corner point of interest and the coordinates of the corner) to construct a common "curbside coordinate frame", such that trajectories with similar intent are spatially similar in this common frame.

For instance, in Fig. 1-2, the pedestrian trajectories shown in intersections $\mathbf{I_1}$ and $\mathbf{I_2}$ represent the same underlying intent of a pedestrian approaching an intersection corner and choosing the left crosswalk to cross the road. These trajectories are spatially dissimilar in the original car frames. However, when mapped into the common frame, they become spatially similar. Each trajectory can also be seen as a sequence of motion primitives, also referred to as "short-term intents" in this work. For example, moving straight and turning left are two different short-term intents. Similar short-term intents can also be spatially dissimilar in the car frame of different intersections, but when mapped into the common frame, they become spatially similar. Building on this important insight, a model comprising of motion primitives and their transition learned in the proposed common frame, instead of the original car frames, can be used to predict in corners of new, unseen intersections with similar semantic cues and different curbside geometries.

In TASNSC, first an affine transformation is used to map training trajectories from the original, car frame of the training intersection into the proposed common frame, such that underlying intent is preserved. Motion primitives and their transition are then learned in this common frame. For prediction in a new, unseen intersection, an observed trajectory is mapped into the common frame using a prior on curbside geometry of the test intersection. A set of future trajectories is predicted in the common frame using the learned motion primitives and their transitions. The predicted trajectories are then mapped back into the car frame of the test intersection using the inverse transformation. While high-definition prior maps are not a limiting constraint for the application of TASNSC, if available, the TASNSC framework is general enough to incorporate context information embedded in such maps.

TASNSC, along with other prior works that demonstrate the ability to transfer knowledge ([4, 48, 52]), assumes that the training and test data consists of similar pedestrian behaviors in scenes with similar semantic cues. In general, the vast majority of current learning techniques, in both supervised and unsupervised settings, make the same assumption of the training and test data having similar feature spaces/distributions. However, in practicality,

models are typically learned for a specific domain and data type and, in most cases, cannot be generalized to new, related domains.

For instance, urban intersections can vary significantly in the type of pedestrian behaviors encountered, regardless of having similar semantic elements such as crosswalks, traffic lights, sidewalks, etc. These pedestrian behaviors can range from faster, rule breaking students near a college campus to slower, conservative pedestrians in a residential area. A trajectory prediction model trained on college campuses would not generalize well to residential areas and vice-versa, inspite of having similar semantic cues. A key challenge is to generalize the trained model to a variety of domains. This requires continually learning from data collected in new domains with as few data points as possible. A prediction model that can transfer knowledge from one intersection to another, while also updating its knowledge base with novel behaviors as, and when, new intersections are visited, is needed.

The third contribution of this thesis is an online, general model for predicting pedestrian trajectories in corners of urban intersections, that can transfer knowledge from one intersection to another, and also augment previously learned knowledge, using data collected from different intersections. This ensures that the model improves over time, as more data becomes available. Similar to TASNSC, a simple prior on curbside geometry (i.e. angle made by intersecting curbs at the corner point of interest and the coordinates of the corner) is sufficient to build this online model.

In the proposed online model, first, a set of motion primitives and their pair-wise transitions are learned in the common frame, using the initially available training data. Whenever more data is collected at the already visited intersections or at an entirely new intersection, a new set of motion primitives is learned. A newly learned motion primitive is added to the model if it is not similar to any of the existing motion primitives, to accommodate any novel primitives representing novel pedestrian behaviors. Relevant transitions between motion primitives are also simultaneously updated. The updated set of motion primitives and their transitions are consistently used for prediction in new, unseen

intersections. Thus, the prediction model improves as more intersections are visited and new data is collected.

1.1 Summary

The main contributions of this thesis are: (i) Context-aware Augmented Semi Nonnegative Sparse Coding (CASNSC) algorithm for long term ($\approx$ 5 seconds), context-aware pedestrian trajectory prediction in urban intersection corners; (ii) Transferable Augmented Semi Nonnegative Sparse Coding (TASNSC) algorithm for long term, context-aware pedestrian trajectory prediction in corners of new, unseen intersections with varying curbside geometries. TASNSC works for both skewed and orthogonal intersection geometries. A simple prior on curbside geometry is sufficient for its application. High-definition prior maps are not a prerequisite, but when available, they can be utilized for embedding additional context information into the TASNSC framework for improved prediction accuracies; (iii) An online, general, context-aware pedestrian trajectory prediction model, that extends the TASNSC framework to improve the prediction model over time, as more data is collected, by incorporating novel behaviors; (iv) Real world datasets of pedestrian trajectories in two intersections around the MIT campus, collected using 2D LiDARs and cameras mounted on-board a Polaris GEM electric golfcart [44].

Chapter 2 provides a brief review of the ASNSC algorithm, used for learning motion primitives in this thesis. The trajectory prediction approach of [13] is also described for completeness and for a better understanding of the proposed prediction models. This is followed by an overview of the use of Gaussian Processes (GPs) to model motion patterns as flow fields in the two-dimensional space ([33, 2]). All throughout this thesis, transitions between motion primitives are modeled using these GP motion patterns. Chapter 2 also discusses the online sparse GP algorithm [16] that is used to build all GP models.

Chapter 3 provides details about the data collection process. It also includes a description of the real world datasets used in this thesis, consisting of pedestrian trajectories in two

different intersections around the MIT campus.

Chapter 4 lays down the framework of CASNSC. Since CASNSC is based on ASNSC, its prediction model also comprises of a set of motion primitives and the transitions between them. Context is incorporated into the two-dimensional GP flow field based modeling of transitions between motion primitives. While the presented CASNSC framework is general enough to incorporate any context feature, the ones used to demonstrate improvement in prediction accuracies over ASNSC are - pedestrian traffic light, curbside orientation and relative distance to curbside. Different combinations of these context features are tested for an in-depth analysis of the algorithm. A squared exponential (SE) kernel function with automatic relevance determination (ARD) [47] is used to learn the relevance of each of the individual features from the available training data. A quantitative comparison of CASNSC with ASNSC shows a 12.5% increase in prediction accuracy, when tested on one of the two real world datasets described in Chapter 3.

Chapter 5 presents TASNSC, which builds upon CASNSC to provide a transferable prediction model. TASNSC can be used to predict pedestrian trajectories in new intersections, with varying curbside geometries, but similar semantic cues as the ones that it was trained on. Real pedestrian trajectories, collected at two intersections with different curbside geometries, are used to conduct the experiments discussed in this chapter. TASNSC achieves 7.2% improvement in prediction accuracy over ASNSC, when trained and tested on the same intersection. This improvement can be attributed to the implicit embedding of context in TASNSC, because of learning motion primitives and their transition in the curbside coordinate frame.

The prediction performance of TASNSC, when trained and tested on different intersections, is comparable to the baseline performance. Additionally, since TASNSC builds upon CASNSC, additional context features can be easily incorporated in learning the transition between motion primitives. Chapter 5 demonstrates this feature, in one of the experiments, by incorporating pedestrian traffic light in predictions using TASNSC. As expected, predic-

tion accuracies improve on incorporating additional context.

Chapter 6 presents the online, general trajectory prediction algorithm that builds upon TASNSC, to incorporate novel behaviors in the prediction model, as more intersections are visited and more data is collected. A detailed overview of the model update algorithm is provided, which includes updating both the motion primitives as well as their transitions, whenever novel/similar motion primitives are learned from new data. For updating motion primitives, a novel distance metric is defined to compute similarities between motion primitives. The online sparse GP algorithm, as in [16], is used for updating the relevant transitions between motion primitives. Preliminary results of improvement in prediction performance with the proposed online model, as more training data is collected, are discussed in this chapter.

Chapter 7 concludes the thesis by providing a detailed discussion on the limitations of the proposed prediction models and scope for future work.

1.2 Related Work

Several papers have been written on short-term prediction of human motion [37, 7, 28, 27], but understanding goals or intent is needed to plan for longer timescales [34, 1]. For instance, [31] demonstrates the ability to accurately predict the final destination of pedestrians using a probabilistic pedestrian modeling approach. The aim of this work, however, is to not just predict the final destination, but also the trajectory that leads to it. Previous work has focused on two main approaches for trajectory prediction [39]: prototype trajectories-based and maneuver intention estimation-based. In general, prototype trajectories-based approaches are more robust to measurement noise when compared to maneuver intention estimation-based approaches, which are mostly Markovian [42, 56, 50] and therefore, rely on the current state only for prediction. However, the prototype trajectories-based approaches can be computationally quite expensive [46, 25] and hence slow in detecting changes in pedestrian intent. They are also susceptible to issues like partial trajectories in the training

dataset being grouped together into a cluster and learned as a trajectory prototype.

As mentioned earlier, Chen et al. [13] combined these two approaches, to inherit the benefits of both, in developing a dictionary learning algorithm, called Augmented Semi Nonnegative Sparse Coding (ASNSC). They achieved significant improvement over state-of-art clustering based approach using Dirichlet Process mixture of Gaussian Process (DPGP). However, since the approach of [13] learns both the motion primitives and their transition using solely the spatial features of the training dataset (x and y position and orientation of pedestrians in the car frame), an important limitation of their work is that available context is not utilized for trajectory prediction.

Most of the previous work on context-based pedestrian trajectory prediction aims to identify stopping versus crossing intent [51, 28, 49, 37, 58, 59], as opposed to long term trajectory prediction which is the objective of this work. In addition, some are also based on the limiting assumption of only one context feature being active at a time, which works for short-term, immediate prediction only [28]. The CASNSC framework is more general and can incorporate multiple context features in the same model, as demonstrated in Chapter 4. Bonnin et al. [8] developed a more generic, context-based, multi-model system for predicting crossing behavior in inner-city situations and zebra crossings. However, the output of their prediction model is a crossing probability as opposed to a future trajectory. More recently, [50] use a combination of an Interacting Multiple Model (IMM) filter for tracking and Latent-dynamic Conditional Random Field (LDCRF) model for intention prediction. Their approach implicitly utilizes situational awareness by embedding human head pose into the LDCRF model and the prediction horizon is limited to 1 second. CASNSC, in contrast, predicts on explicit inclusion of context, for a long-term prediction horizon of 5 seconds. Furthermore, [34] used jump-Markov process for long term prediction of pedestrian motion by incorporating traffic light and crosswalks as semantic features. The output of their prediction model is an *occupancy map* of feasible trajectory predictions. CASNSC instead outputs a *set of likely trajectories* with increased accuracy by incorporating context in the ASNSC based prediction model [13].

Coscia et al. [15] forecast long-term behavior of pedestrians by making use of past observed patterns and semantic segmentation of a bird's eye view of the scene. Their approach, when applied in the real world, on board a self-driving vehicle, would require accurate high definition semantic priors/maps of each scene. High definition maps are expensive to create and maintain. It is also unclear if their prediction model can be generalized to predict in new, unseen scenes. Ballan et al. [4] and Sadeghian et al. [48] follow a similar approach to path prediction while also demonstrating the ability to "transfer knowledge". Their models can predict in unseen locations with similar semantic elements. However, a prior bird's eye view of the scene is needed for both these approaches as well.

Chapter 2

Preliminaries

In this chapter, first, a few important facts and notations are listed for brevity. As mentioned in Chapter 1, motion primitives form an integral component of this work. There exist several prior works on learning motion primitives for trajectory prediction [14, 54, 38, 57, 13]. Of these, the Augmented Semi Nonnegative Sparse Coding (ASNSC) [13] algorithm is used in this work, since it addresses the limitations of prior works. Therefore, a brief review of ASNSC is provided here for completeness. This is followed by a review of Gaussian Processes [47] (GPs) and two-dimensional GP flow fields [33, 2]. An overview of the online sparse GP algorithm [16] used to model all GPs in this work is also provided.

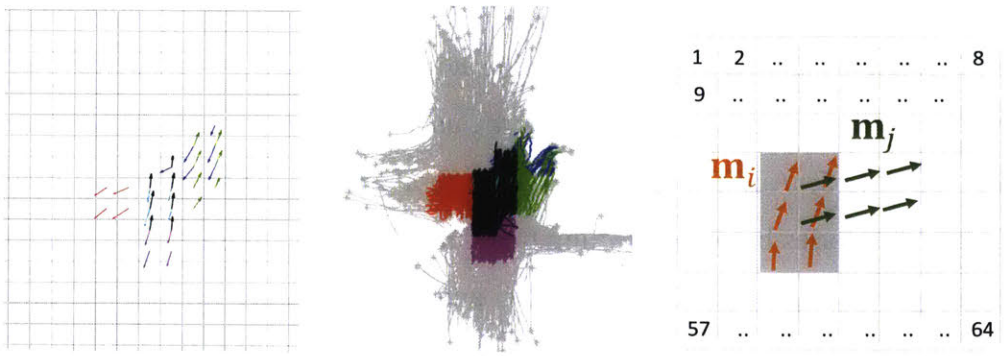


Figure 2-1: (a) Each color represents a single motion primitive $\mathbf{m}_i$ for real pedestrian trajectories; (b) Segmentation of training trajectories (in gray) into clusters, where each cluster is best explained by the motion primitive of the same color in (a); (c) Illustration of two motion primitives $\mathbf{m}_i$ and $\mathbf{m}_j$ in a grid-based world consisting of $L = 64$ cells, indexed as shown. The shaded gray region denotes A_i i.e. the active cells of $\mathbf{m}_i$

2.1 Facts and Notations

The proposed pedestrian trajectory prediction model M can be applied to pedestrians in any intersection corner. In all chapters, M is defined as a set of motion primitives and motion patterns representing the transition between motion primitives s.t. $M \triangleq \{B, U, W\}$. Here, B is the set of motion primitives i.e. $B \triangleq \{\mathbf{m}_i\}$, U is the set of unitary motion patterns [12] modeled as two-dimensional GP flow fields i.e. $U \triangleq \{GP_i^{uni}\}$ and W is the set of transition motion patterns [12], again modeled as two dimensional GP flow fields i.e. $W \triangleq \{GP_i^{trans}\}$. Each motion primitive $\mathbf{m}_i$, when learned using ASNSC, is mathematically represented by a set of indices of active cells (A_i) and a set of cell-wise velocity in the grid-based world i.e. $V_i \triangleq \{\mathbf{v}_i^k\}$. Here, $\mathbf{v}_i^k$ is the velocity of $\mathbf{m}_i$ in the k-th grid cell. $\mathbf{T}, \mathbf{R}$ denote the transition matrices for motion primitives. $\mathbf{T}(i, j)$ gives the number of trajectories transitioning from $\mathbf{m}_i$ to $\mathbf{m}_j$ for off-diagonal elements, and the number of trajectories ending in $\mathbf{m}_i$ for diagonal elements [12]. Similarly, $\mathbf{R}(i, j)$ is the set of trajectories transitioning from $\mathbf{m}_i$ to $\mathbf{m}_j$ for off-diagonal elements, and set of trajectories ending in $\mathbf{m}_i$ for diagonal elements. M is trained and tested on a vectorized representation of trajectories in a grid-based world, denoted as $\mathbf{t}_i$ for the i-th trajectory [13]. Datasets from real world intersections $(\mathbf{I_1}, \mathbf{I_2})$ around the MIT campus, consisting of pedestrian trajectories, were collected using a golfcart parked at intersection corners, equipped with 2D LiDARs and cameras [43, 44]. A prior on curbside geometry (i.e. angle made by curbsides meeting at a corner) is used for mapping trajectories into the common curbside coordinate frame $\mathscr{C}$, using an affine projection function $\mathscr{T}$, as defined in Chapter 5. Additionally, in Chapter 6, a prior on sidewalk width is used for normalizing trajectories before projecting them into the common frame $\mathscr{C}$. $\mathscr{D}$ denotes the set of training trajectories. For both ASNSC and Context-aware ASNSC (CASNSC) in Chapter 4, it is the set of trajectories in the original car frame. For Transferable ASNSC (TASNSC) in Chapter 5, $\mathscr{D}$ is the set of training trajectories mapped in $\mathscr{C}$. In Chapter 6, $\mathscr{D}$ denotes the set of training trajectories normalized with respect to sidewalk width and then mapped in $\mathscr{C}$. B, U, W, without an overhead bar, represent the current prediction model. When denoted with an overhead bar i.e. $\bar{B} \triangleq \{\bar{\mathbf{m}}_i\}, \bar{U} \triangleq \{\bar{GP}_i^{uni}\}, \bar{W} \triangleq \{\bar{GP}_i^{trans}\}$, they represent knowledge gained from a new intersection that is used for updating M.

2.2 Augmented Semi Nonnegative Sparse Coding [13]

Let the training dataset consist of n trajectories, where each trajectory is a sequence of two-dimensional position measurements taken at a fixed time interval. In a grid-based world, with K cells, the i-th trajectory can be represented as a column vector $\mathbf{t}_i \in \mathbb{R}^K$ such that the k-th element of $\mathbf{t}_i$ is the average velocity of the i-th trajectory in the k-th grid cell. Given this vectorized representation of training trajectories, ASNSC learns a set of L motion primitives, given by $B = \{\mathbf{m}_1, \ldots, \mathbf{m}_L\}$. Each color in Fig. 2-1(a) represents a single motion primitive $\mathbf{m}_i$, learned from the trajectories shown in gray in Fig. 2-1(b). $\mathbf{m}_i$ can be mathematically represented using two features: (1) A_i, defined as the set of indices of 'active cells' (see shaded region in Fig. 2-1(c)); and (2) $V_i \triangleq \{\mathbf{v}_i^k\}$ i.e. the set of cell-wise velocity. Here, $\mathbf{v}_i^k$ is the velocity of $\mathbf{m}_i$ in the k-th grid cell (out of $L = 64$ grid cells in Fig. 2-1(c)).

As shown in Fig. 2-1(b), B is used to segment the original training trajectories, shown in gray, into clusters. Each color in Fig. 2-1(b) is one such cluster i.e. a group of trajectory segments, best explained by the motion primitive in Fig. 2-1(a) in the same color. These clusters are used to create the transition matrices, $\mathbf{T} \in \mathbb{Z}^{L \times L}$ and $\mathbf{R}$, as described in Section 2.1. Each transition, i.e. a concatenation of two dictionary atoms $\{\mathbf{m}_i, \mathbf{m}_j | \mathbf{T}(i,j) > 0\}$, is modeled as a two-dimensional GP flow field [33, 2]. Two independent GPs, (GP_x, GP_y), called GP motion patterns, are used to learn a mapping from the two-dimensional position features to the x and y velocities respectively, for each transition [12].

2.3 Motion Patterns as Gaussian Process Flow Fields

The definition of motion patterns as flow fields of trajectory derivatives in the $x-y$ space was introduced in [33]. Mathematically, a motion pattern thus defined is a mapping from two dimensional positions (x,y) to a distribution over trajectory derivatives $(\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t})$, or simply velocities (v_x, v_y). Joseph et al. [33] show that modeling motion patterns as flow fields, rather than single representative trajectories, not only allows for grouping of trajectories with similar key characteristics, but also ensures that the representation is agnostic to

different lengths and discretization across trajectories. Therefore, following [33], this work also defines motion patterns as flow fields. GPs are used to model the mapping from pedestrian positions to velocities. To elaborate, each transition between motion primitives, $\{\mathbf{m}_i, \mathbf{m}_j | \mathbf{T}(i,j) > 0\}$, is modeled using two independent GP motion patterns (GP_x and GP_y), such that

$$GP_x : (x,y) \rightarrow v_x \; ; \qquad GP_y : (x,y) \rightarrow v_y \tag{2.1}$$

These independent GPs are fitted to the set of training trajectories in $\mathbf{R}(i,j)$. GP motion patterns fitted to the non-empty, diagonal elements in $\mathbf{R}$, together constitute U, i.e. the set of unitary motion patterns. Similarly, GP motion patterns fitted to the non-empty, off diagonal elements in $\mathbf{R}$, together constitute W, i.e. the set of transition motion patterns. Note that representing motion patterns as GP flow fields provides a distribution over velocities for every pedestrian position. This fact can be used to predict future pedestrian trajectories corresponding to a specific motion pattern. Given the position of a pedestrian at time t (x_t, y_t) and the GP motion pattern of interest (GP_x and GP_y), the expected velocity of the pedestrian at the given location (v_{x_t}, v_{y_t}), for the given motion pattern, can be obtained using (2.1). Future pedestrian position for the given motion pattern can thus be obtained as

$$(x_{t+1}, y_{t+1}) = (x_t, y_t) + (v_{x_t}, v_{y_t})\Delta t \tag{2.2}$$

2.4 Sparse Online Gaussian Processes

GP inference has a computation time complexity of $\mathcal{O}(n^3)$, where n is the total number of training data points [47]. For large datasets, a cubic growth in the time complexity of prediction is computationally prohibitive. Sparse representations of GPs aim to reduce this computational burden by performing the most time-consuming matrix operations in inference, such as inversions, on a smaller, representative subset of the training data. This helps bring down the computational time complexity of inference to $\mathcal{O}(nm^2)$, where m is the number of data points in the chosen representative subset [47, 16]. Therefore, all GP models

used in this work are sparse and hence, ensure that the computational time complexity of prediction grows linearly in the size of the training dataset.

An important contribution of this work is to build a general prediction model that can update knowledge as, and when, new training data is received. Recall that the proposed prediction model is a set of motion primitives and GP motion patterns. Two key challenges in updating the unitary and transition GP motion patterns and using them for trajectory prediction are: (1) online GP update and; (2) performing prediction using the updated GPs without having to store previous data. To address these challenges, the online sparse GP algorithm of [16] is used to model motion patterns in this work.

In [16], the GP parameters, as specified by α and $\mathbf{C}$, can be updated iteratively, in a single pass through the entire training set, for a given maximal size of the $\mathscr{BV}$ set. The $\mathscr{BV}$ set is the set of inducing inputs and outputs, or more simply, the representative subset of the full training data. This ensures that as, and when, new trajectories are observed, the GP parameters (α and $\mathbf{C}$) and the $\mathscr{BV}$ set can be updated by iterating through the new set of trajectories, as long as the set of sufficient statistics i.e. $\{, \mathbf{C}, \mathscr{BV}\}$ are retained. Furthermore, Csato et al. [16] also provide an approximate posterior kernel of the process, which can be used to represent predictive variance for a new set of inputs for a given set of updated GP parameters α and $\mathbf{C}$.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Data Collection

An increase in the number of publicly available, annotated datasets has been a significant driver of recent advances in machine learning techniques, alongside better computation power and more efficient algorithms. For instance, the success of Computer Vision closely followed the internet explosion. With millions of people, all over the world, posting images and videos over the Internet, it was much easier for Computer Vision scientists to create huge, organized datasets like Pascal VOC [23], COCO [40] and ImageNet [18]. These datasets enabled researchers to achieve human-like accuracies in visual object detection tasks using state-of-art deep learning techniques.

Recognizing the need for creating similar datasets for advancing research in the field of autonomous driving, the past few years saw the release of datasets collected by vehicles driving around in the real world, equipped with sensors such as LiDARs, GPS and cameras. The KITTI Vision Benchmark Suite [26], the Berkeley Deep Drive (BDD) 100K [60] and the Ford vision and LiDAR dataset [45] are some examples of such datasets. Given the relatively fewer number of scenarios with pedestrians in these more general datasets, and the challenging nature of the pedestrian detection and motion prediction tasks, pedestrian specific datasets have also been released. The Caltech pedestrian dataset ([19, 20]), the ETH BIWI pedestrian dataset [22] and the Daimle pedestrian dataset [22] are some examples. All these datasets consist of annotated videos with pedestrians, taken from a vehicle driving through regular traffic in urban environments. However, since the trajectory prediction

model proposed in this thesis is applicable to pedestrians specifically in intersections, a good number of training trajectories cannot be extracted from these datasets. Also, a prior on curbside geometry is difficult to extract from videos.

To fill the gap in the amount of available, relevant training data, two real world datasets of pedestrians in intersections were collected. The data collection platform and software architecture used to extract pedestrian trajectories from raw sensor data follows [30]. The data collection process involves parking the data collection platform on the sidewalk of an intersection corner, for long durations ($\approx$ 1-2 hours), such that pedestrian movement is not disturbed. Pedestrians are then detected and tracked using on-board sensors to extract their trajectories.

3.1 Data Collection Platform

3.1.1 Golfcart-based

It consists of a Polaris GEM electric golfcart[1] equipped with three Logitech C920 cameras and two SICK LMS151 LiDARs, as described in detail in [30]. The cameras provide a 270 degree field-of-view (FOV) in the front of the vehicle. The upper LiDAR is used for localization of the vehicle in a prior map and the lower LiDAR is used for pedestrian detection and tracking.

3.1.2 Tripod-based

Given the small size and nature of the golfcart, it is easy to park it on sidewalks at intersection corners for long durations. However, this was possible in intersections in and around the MIT campus only, since they have relatively wider sidewalks to account for high pedestrian density. To collect data in more interesting intersections, away from the MIT campus, a tripod setup, with the same sensors and software stack as that on the Golfcart, was built.

[1] https://gem.polaris.com/en-us/

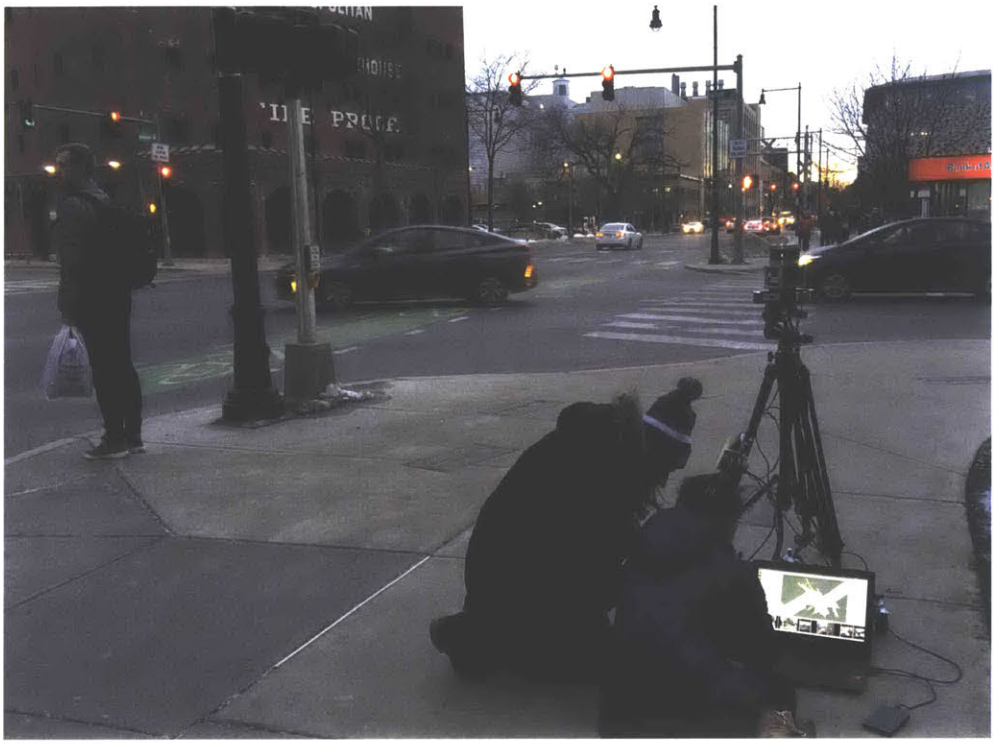


Figure 3-1: The tripod setup for data collection, consisting of 6 cameras and a Velodyne. It can be easily placed on the curbside of a busy intersection corner, with minimal disturbance to pedestrian movement around it.

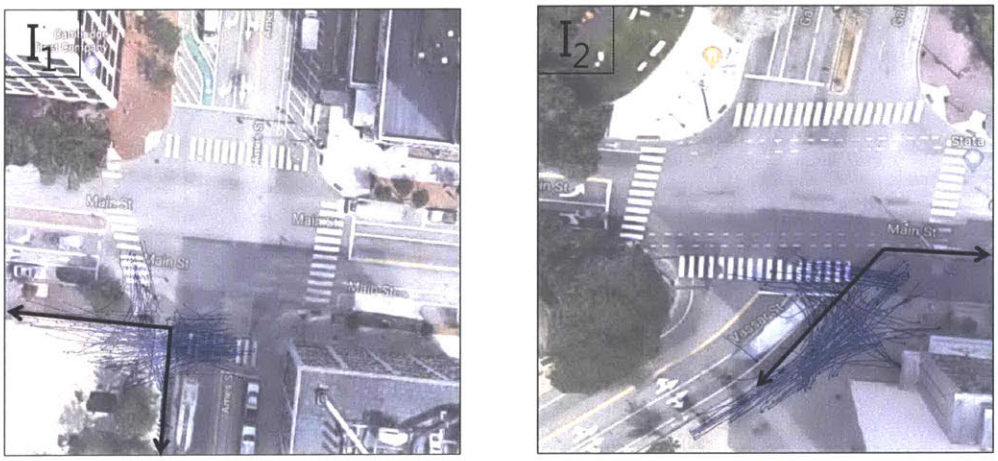


Figure 3-2: An overhead snapshot of intersection $\mathbf{I_1}$ with orthogonal curbsides (left) and intersection $\mathbf{I_2}$ with skewed curbsides (right). Pedestrian trajectories, shown in blue, were collected using a 2D LiDAR and cameras, on-board a Polaris GEM vehicle parked at the intersection corners.

As shown in Fig. 3-1, the tripod setup consists of six Logitech C920 cameras. The two LiDARs in the Golfcart are replaced by a single HDL-32E Velodyne. It is easy to extract two different laserscans from its pointcloud, which can serve the purpose of the upper and lower LiDAR in the Golfcart setup. However, this setup is still in the testing phase and all the data mentioned in this thesis was collected using the Golfcart.

3.2 Software Architecture

The Very Fast pedestrian detection package [5] is applied to camera images to produce bounding boxes indicating pedestrian locations. A prior map is used to filter the LiDAR scan, followed by clustering using Dynamic Means [9]. The output of the vision and LiDAR modules is provided to a fusion module that outputs pedestrian trajectories. More details on the individual vision and LiDAR modules and the fusion module can be found in [5].

3.3 Dataset Description

Data was collected at two intersections in the MIT campus, with different curbside geometries and high footfall during the day. Fig. 3-2 shows an overhead screenshot from Google Maps of the chosen intersections. $\mathbf{I_1}$ has more or less orthogonal curbsides while $\mathbf{I_2}$ has skewed curbsides. A subset of the entire training data, consisting of pedestrian trajectories, is shown in blue and overlayed on the Google Maps screenshot for better visualization. 997 pedestrian trajectories were collected in $\mathbf{I_1}$ and 575 trajectories were collected in $\mathbf{I_2}$.

Chapter 4

Context-aware motion prediction

Chen et al. [13] showed significant improvement in pedestrian trajectory prediction, using Augmented Semi Nonnegative Sparse Coding (ASNSC), as compared to that using state-of-art clustering based approach, using Dirichlet Process mixture of Gaussian Process (DPGP) [25]. However, ASNSC uses spatial features (pedestrian position and orientation in the car frame) only for trajectory prediction. Any context that may influence a pedestrian's intent is ignored. As already motivated in Chapter 1, context plays an important role in urban environments, such as intersections with pedestrian traffic lights and/or tightly packed sidewalks with numerous pedestrian interactions. In such environments, a context-aware prediction model would better infer pedestrian intent, and thus, have better trajectory prediction accuracies than that of ASNSC.

This chapter presents Context-aware Augmented Semi Nonnegative Sparse Coding (CASNSC), as an extension of ASNSC. In CASNSC, first a dictionary of motion primitives is learned using ASNSC. Context features, such as curbside orientation, relative distance to curbside and pedestrian traffic light status, are then incorporated into the Gaussian Process (GP) flow fields based modeling of pair-wise transitions between the learned motion primitives. This helps improve prediction accuracies as context influences the probability of transition between two motion primitives. For instance, consider the pedestrian motion primitives at an intersection corner, as shown in Fig. 4-1. The transition between motion primitives shown in magenta and blue, has a higher probability, than that between motion

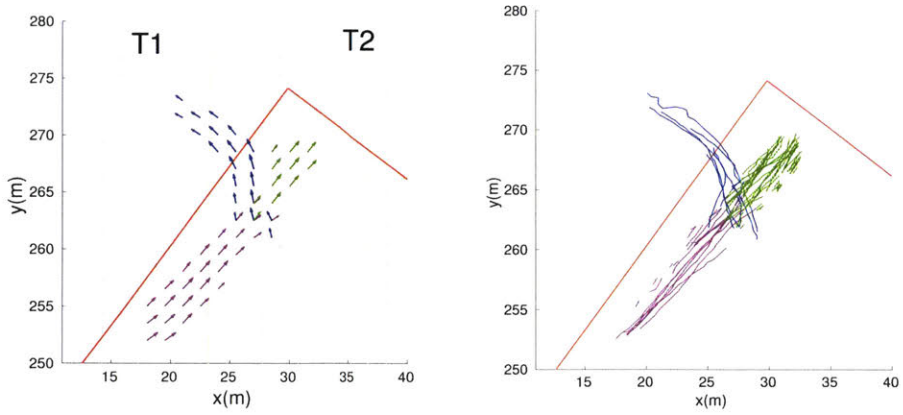


Figure 4-1: Motion primitives learned using the ASNSC framework (left) and clustering of training trajectories on the basis of the learned motion primitives (right). Each dictionary atom is shown in a different color. T1 and T2 denote two different traffic lights, the status of which influences transition between dictionary atoms. For e.g., the transition between dictionary atoms shown in magenta and blue has a higher probability than that between dictionary atoms shown in magenta and green for T1 = 1 (crosswalk clear for pedestrians to cross), T2 = 0

primitives shown in magenta and green, for the scenario in which T1 = 1 (left crosswalk is clear for pedestrians to cross) and T2 = 0. Here, T1 and T2 together denote the pedestrian traffic light context feature.

The main contributions of this chapter are: (i) Utilization of context to map pedestrian trajectories in the car's $x-y$ coordinate frame, into a rotated $x'-y'$ coordinate frame, in which the two coordinates are independent of each other (see Fig. 4-2). As discussed in Chapter 2, ASNSC assumes $x-y$ independence in modeling the transition between motion primitives. Therefore, such a mapping improves prediction accuracies by improving the GP modeling accuracy; (ii) Use of context features, such as traffic lights, relative distance to curbside and curbside orientation, that are independent of the training intersection geometry (for orthogonal intersections). This lays the foundations for building a transferable prediction model, that can be generalized to predict in intersections other than the one it has been trained on; (iii) CASNSC framework for embedding context such as traffic lights, relative distance to curbside and curbside orientation in [13] in ASNSC; (iv) Comparison of three different variations of CASNSC with ASNSC to show improvement in prediction accuracy.

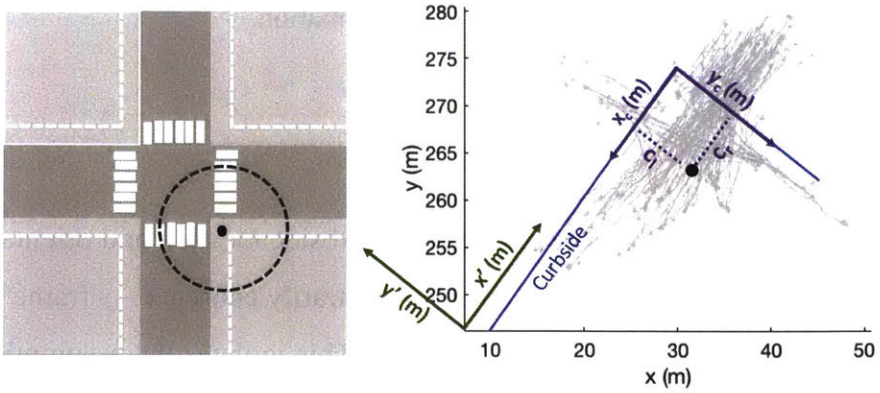


Figure 4-2: A typical four-way intersection (left) is used to explain the *curbside orientation* and *relative distance to curbside* context features. The zoomed portion (right) shows a pedestrian location as a black dot. $(c_l, c_r)^T$ denotes the vector of distance to the two curbsides of interest and is used as the *relative distance to curbside* context feature. The signs of vector elements c_l and c_r are determined using the curbside coordinate frame $x_c - y_c$. Pedestrian position in the rotated coordinate frame $x' - y'$, which has the same orientation as that of the curbside in the global coordinate frame $x - y$, is used as the *curbside orientation* context feature.

4.1 Algorithm

The proposed approach uses two sets of features: 1) *dictionary features*, $\mathbf{X_d}$, which are used for learning the dictionary (B) i.e. the set of motion primitives (Algorithm 1, lines 1-4); and 2) *transition features*, $\mathbf{X_t}$, which are used for learning the sets of unitary and transition GPs, U and W respectively (Algorithm 1, lines 5-10). ASNSC uses the same set of two-dimensional position feature, $(x, y)^T$, as both $\mathbf{X_d}$ and $\mathbf{X_t}$. CASNSC, in contrast, uses $(x, y)^T$ as $\mathbf{X_d}$, but a combination of position and context features as $\mathbf{X_t}$. Three different combinations are used as $\mathbf{X_t}$ in the experiments included in this chapter, for an in-depth analysis of the algorithm.

4.1.1 Context features

Pedestrian traffic light

A pedestrian's decision to go left or right is influenced by the status of two pedestrian traffic lights (T1, T2) in a four-way intersection scenario. A single-dimensional feature vector, (tr), is sufficient to capture the environment context with respect to both the traffic lights as the

change in status of (T1, T2) captures redundant information.

Curbside orientation

Pedestrian motion in sidewalks is constrained by curbside location and orientation. Thus, the x, y position coordinates of trajectories in an arbitrarily chosen $x-y$ frame (car frame) would be dependent on each other (see Fig. 4-2). This violates the assumption of $x-y$ independence in the squared exponential (SE) kernel function used by the GP transition models in U and W. As shown in Fig. 4-2, rotating the $x-y$ frame into the $x'-y'$ frame, which has the same orientation as that of the curbsides, can reduce this dependence. Such a transformation improves GP modelling, and consequently, trajectory prediction accuracy. Furthermore, the described transformation is equivalent to embedding curbside orientation as a context feature in the prediction model.

Relative distance to curbside

In addition to curbside orientation, the relative distance of a pedestrian (treated as a point mass), to the two curbsides intersecting at a corner, also provides useful contextual information. This distance can be computed using either a prior map of the environment or by online curb identification. As, shown in Fig. 4-2, a two-dimensional vector, $(c_l, c_r)^T$ is used as the relative distance to curbside feature, which is equivalent to transforming the arbitrarily chosen $x-y$ coordinate frame into the x_c-y_c coordinate frame, that is exactly aligned with the curbsides of interest.

4.1.2 Feature sets used as transition features ($\mathbf{X_t}$)

Position and pedestrian traffic light

The first feature set is a combination of the two-dimensional pedestrian position feature and the *pedestrian traffic light* context feature, i.e., $\mathbf{X_t} = (x, y, tr)^T$. Application of the CASNSC framework with this particular feature set will be referred to as CASNSC-1.

Curbside orientation and pedestrian traffic light

As described earlier, an inherent limitation of the first feature set is the fact that x, y are dependent on each other because of trajectories being constrained by curbside geometry (see Fig. 4-2). This violates the $x-y$ independence assumption made in the GP transition models in sets U and W. To address this issue, *curbside orientation* is combined with *pedestrian traffic light* to create another feature set $\mathbf{X_t} = (x', y', tr)^T$. The specific application of CASNSC with this feature set will be referred to as CASNSC-2.

Relative distance to curbside and pedestrian traffic light

Another important piece of contextual information missing in the second feature set is the actual location of the intersection corner and curbsides, which can also be an important indicator of pedestrian intent. This missing piece of information is incorporated by combining the *relative distance to curbside* context feature with the *pedestrian traffic light* context feature to create the third feature set $\mathbf{X_t} = (c_l, c_r, tr)^T$. The CASNSC framework with this feature set will be referred to as CASNSC-3.

4.1.3 Kernel function

A squared exponential (SE) kernel function with automatic relevance determination (ARD) is used for modeling GP motion patterns, as it allows for combination of features with different characteristics and scales each feature in accordance with its relevance [47]. Mathematically, it is given by the following form:

$$k(\mathbf{X}, \mathbf{X}') = \sigma_f^2 \exp(-\sum_{i=1}^{m} \frac{1}{2{l_i}^2}(x_i - x_i')^2) \tag{4.1}$$

where, $\mathbf{X} = \{x_i\}$ s.t. $i = \{1, ..., m\}$. Here, x_i is the i-th feature, out of a total of m features and l_i is the characteristic length of this feature. The characteristic lengths along with the pre-multiplication factor σ_f constitute the set of hyper-parameters, which needs to be either tuned or learned. For instance, for predictions using CASNSC-1 where $\mathbf{X} = \mathbf{X_t} = (x, y, tr)^T$, $m = 3$ and the set of hyper-parameters would be given by the column vector $h = (l_x, l_y, l_{tr}, \sigma_f)^T$.

4.2 Results

CASNSC is tested on real pedestrian data collected by a Polaris GEM vehicle equipped with cameras and LiDARs, as described in Chapter 3. The dataset used in this chapter is a subset of the data collected in intersection $\mathbf{I_1}$, with approximately orthogonal curbsides. The training data consists of 218 training trajectories, randomly sampled from the whole dataset. Out of the leftover trajectories, 32 were selected as test trajectories.

Algorithm 1: Context-aware Augmented Semi Nonnegative Sparse Coding (CASNSC)

input : set of training trajectories in car frame ($\mathscr{D}$), dictionary features ($\mathbf{X_d}$), transition features ($\mathbf{X_t}$), observed trajectory in car frame (t_o)
output : set of predicted trajectories in car frame (t_p)

/* Dictionary Learning Phase */
$B \leftarrow \emptyset, S \leftarrow \emptyset$; // S is the set of sparse coefficients
while *not converged* **do**
  $\{B,S\} = \text{ASNSC}(\mathscr{D}, \mathbf{X_d}, \lambda)$; // λ is regularization parameter
$\mathbf{T}, \mathbf{R} \leftarrow \text{transitionMatrix}(B, S, \mathscr{D})$
/* Transition Learning Phase */
$U \leftarrow \emptyset, W \leftarrow \emptyset$
for $\forall\ (i,j)\ s.t.\ \{\mathbf{T}(i,j) > 0\}$ **do**
  if $i == j$ **then**
    $U \leftarrow \{U, GP_i^{uni}(\mathbf{X_t}, \mathbf{R}(i,i))\}$; // fit unitary GP to trajectories in $\mathbf{R}(i,i)$ using features specified by $\mathbf{X_t}$
  else
    $W \leftarrow \{W, GP_{ij}^{trans}(\mathbf{X_t}, \mathbf{R}(i,j))\}$; // fit transition GP to trajectories in $\mathbf{R}(i,j)$ using features specified by $\mathbf{X_t}$
/* Prediction Phase */
$\hat{k} = \text{argmax}_k\ \mathbb{P}(t_o | GP_k^{uni})$; // pick most likely unitary GP
for $\forall\ j\ s.t.\ \mathbf{T}(\hat{k}, j) > 0;\ \hat{k} \neq j$ **do**
  $t_p \leftarrow \{t_p, \text{predict}(t_o, GP_{\hat{k}j}^{trans})\}$; // predict a trajectory for each valid transition from the most likely unitary motion pattern

A prior map of the environment is used to extract curbside boundaries. Pedestrian traffic light status is manually annotated. An observation history of 2.5 seconds prior to the pedestrian entering the intersection is used to predict 5 seconds ahead in time. Fig. 4-3 provides

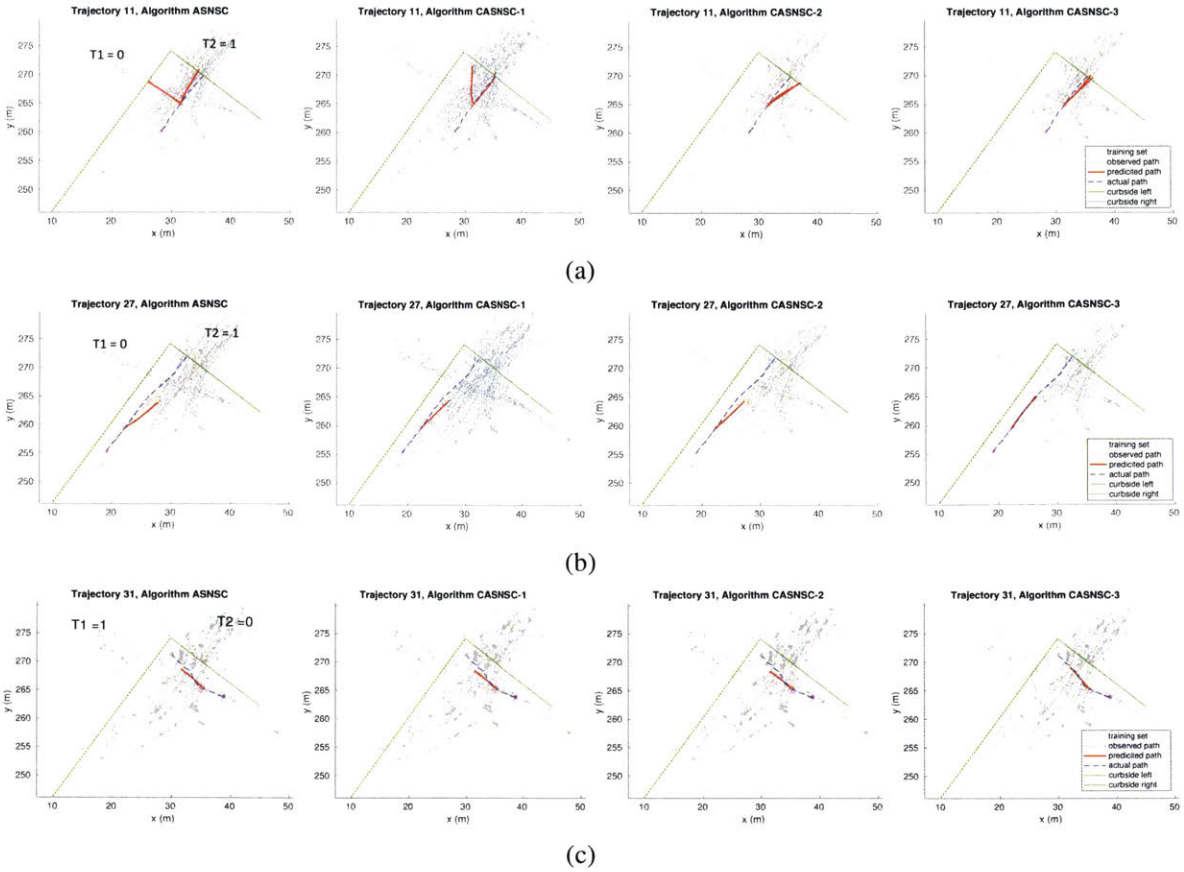


Figure 4-3: Comparison of prediction results of ASNSC (first column) with those of CASNSC-1: $\mathbf{X_t} = (x, y, tr)^T$ (second column), CASNSC-2: $\mathbf{X_t} = (x', y', tr)^T$ (third column) and CASNSC-3: $\mathbf{X_t} = (c_l, c_r, tr)^T$ (fourth column). Each row represents a different test trajectory. The curbside is shown in green, training trajectories in gray, observed trajectory in pink, actual future trajectory in dotted blue and predicted trajectories in red.

a qualitative comparison of CASNSC with ASNSC using all 3 feature sets described in the previous section. While ASNSC provides the set of all feasible trajectories given the intersection geometry, CASNSC picks those that are closest to the actual trajectory, in the correct direction, taking context into account.

In the first scenario (trajectory 11), *pedestrian traffic lights'* status is given by T1 = 0, T2 = 1. The pedestrian enters the intersection and is faced with a choice between continuing to move straight or turning left. While ASNSC predicts a set of all feasible trajectories, completely ignoring the context, CASNSC-1 predicts the correct future direction of motion as it can incorporate context (T2 = 1) into account. CASNSC-2 provides an even better

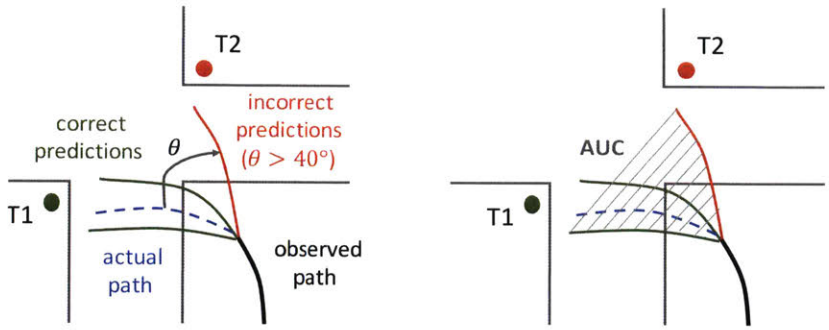


Figure 4-4: (Left) *Incorrect* and *correct* predictions at an intersection scenario. (Right) Use of AUC as a metric for measuring variance in prediction.

prediction owing to the more accurate GP models created by incorporating *curbside orientation*. CASNSC-3 outperforms all and its prediction is not just most correct in terms of the predicted future direction of motion, but also in terms of following the actual trajectory almost exactly. In the second scenario (trajectory 27), traffic light status is the same and while all four predictions are in the right direction, CASNSC-3 is again the most accurate. In the third scenario (trajectory 31), traffic light status is given by T1 = 1, T2 = 0. Again, while all four predictions are in the right direction, CASNSC-3 is most accurate and follows the actual trajectory almost exactly.

Fig. 4-4 illustrates the metrics used for performance evaluation and Table 4.1 provides a quantitative comparison of ASNSC with CASNSC-1, CASNSC-2 and CASNSC-3. As illustrated in Fig. 4-4, *Area Under the Curve (AUC)* [29] is used as a metric for measuring the correctness of predicted future direction of motion, such that a larger *AUC* corresponds to a better prediction. Table 4.1 indicates that *AUC* for predictions using CASNSC-3 is the lowest, confirming that embedding context provides better predictions of future direction of motion. *Classification accuracy* is also measured, which represents the fraction of *correct* predictions, weighted by their likelihood for a more realistic estimate of the metric. Mathematically, if a set of n trajectories is predicted as $\{t_1,\ldots,t_n\}$, with their likelihood of prediction given by $\{l_1,\ldots,l_n\}$, and the *correct* predictions are identified as $\{t_i\}\ \forall\ i\ \in \mathbf{C} \subset \{1,\ldots,n\}$, the

Algorithm	Classification accuracy(%)	MHD(m)	AUC(m^2)	Computation time(s)
ASNSC	83.71	2.09	131.13	**0.03**
CASNSC-1	85.25	2.33	105.50	0.51
CASNSC-2	90.00	2.05	85.23	0.48
CASNSC-3	**94.20**	**1.77**	**49.44**	0.04

Table 4.1: Performance evaluation comparison of CASNSC with ASNSC

classification accuracy is given by:

$$\text{Classification accuracy } \% = \frac{\sum_{i \in \mathbf{C}} l_i}{\sum_{k=1}^{n} l_k} \times 100\%. \tag{4.2}$$

As seen in Fig. 4-4, *correct* predictions are defined as those in which the angular deviation from the observed trajectory i.e. θ is less than 40 degrees. In addition to the illustrated metrics, the *Modified Hausdorff distance (MHD)* [21] is used to compare predicted pedestrian trajectories with the ground truth. We again use the likelihood of predicted trajectories to compute the weighted average of MHD for a more accurate quantification of the metric.

Table 4.1 shows an improvement in all the chosen metrics, with only a slight increase in computation time. All computations were performed on an Intel Core i7-7700HQ processor in Matlab R2016b. CASNSC-3, which uses a combination of relative distance to curbside and pedestrian traffic light as *transition features*, shows a 12.5% improvement in *classification accuracy*, 15.3% improvement in *MHD* and reduces *AUC*, by a factor of 2.65. There is scope for further improvement on incorporation of other features like crosswalks, location of subway stations etc.

Context features, like curbside orientation, provide spatial information that is independent of intersection geometries. A trajectory prediction model, based on such context features alone, will therefore also be independent of the specific intersection geometry it is trained on. This insight is used in the following chapter to develop a context-aware pedestrian trajectory prediction model, for urban intersections, that can transfer knowledge from one intersection to another.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Transferable motion prediction model

As shown in the previous chapter, CASNSC incorporates available context into the ASNSC framework, to significantly improve pedestrian trajectory prediction accuracies in urban intersections. CASNSC demonstrated the ability of modeling pair-wise transition between motion primitives using context features alone (refer CASNSC-3 in Chapter 4). However, it still uses spatial features (pedestrian position and orientation in car frame) in learning the motion primitives themselves. This makes the prediction model of CASNSC dependent on the specific training intersection geometry, and prevents its generalization to prediction in new intersections.

Going one step further, a model trained on solely context features, such as distance to curbside, pedestrian traffic lights etc., would be independent of the specific training intersection geometry. Such a model can be generalized to predict in new, unseen intersections with similar semantic cues. Data collection in the real world is both expensive and time consuming. A model trained on context features alone, can be used to transfer knowledge from one intersection to another. Such a model would be superior to one that uses both context and spatial features, such as CASNSC, and hence needs to be trained on every intersection.

The main contributions of this chapter are: (i) Introduction of the "curbside coordinate frame" as a common frame in which spatially dissimilar trajectories from different intersec-

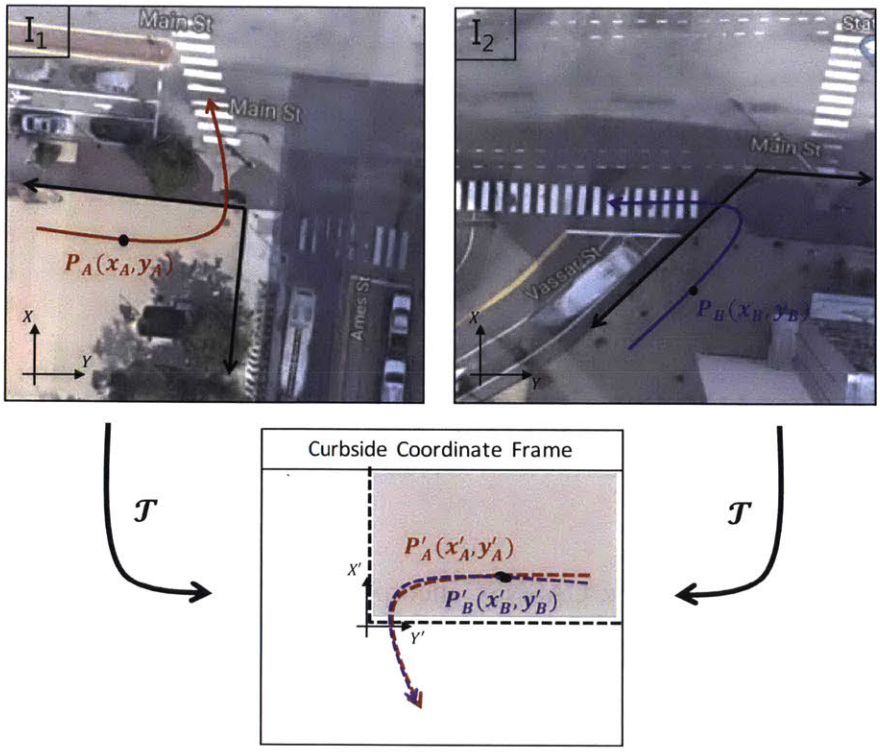


Figure 5-1: An illustration to show how points $P_A(x_A, y_A)$ on the red trajectory in intersection $\mathbf{I_1}$ and $P_B(x_B, y_B)$ on the purple trajectory in intersection $\mathbf{I_2}$, under the transformation $\mathcal{T}$, map to points $P'_A(x'_A, y'_A)$ and $P'_B(x'_B, y'_B)$ in the curbside coordinate frame. We show that $\mathcal{T}$ is in general an affine transformation. Since pedestrian trajectories in urban intersections are significantly constrained by the curbsides, transforming them into the curbside coordinate frame using an affine transformation, intuitively would map trajectories with similar pedestrian intent approximately on top of each other in the curbside coordinate frame. This insight helps in developing a general, transferable pedestrian trajectory prediction model.

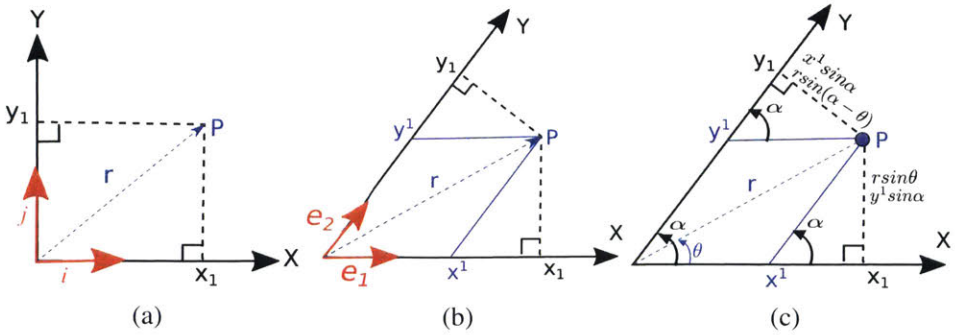


Figure 5-2: (a) Orthogonal coordinate system; (b) Skewed coordinate system; (c) Calculation of contravariant components in a skewed coordinate system using trigonometry

tions, representing the same underlying pedestrian intent, are spatially similar (see Fig. 5-1); (ii) Introduction of a novel representation of distance to curbside as the contravariant components of pedestrian positions in the curbside coordinate frame (can be orthogonal or skewed). This representation ensures that distance to curbside, as a context feature, is independent of intersection geometry (as opposed to other representations such as orthogonal distance to curbside in CASNSC and other prior works [59, 37, 58]); (iii) Proof of the fact that the transformation of pedestrian trajectories from the car frame, into the curbside coordinate frame, is affine. Such a transformation, therefore, preserves properties such as collinearity, parallelism etc. across intersections while encoding context (see Fig. 5-1); (iv) Transferable Augmented Semi Nonnegative Sparse Coding (TASNSC), as a solely context-based pedestrian trajectory prediction model for accurate, long term ($\approx$ 5 seconds) trajectory prediction in new, unseen intersections with similar semantic cues as the ones that the model is trained on.

5.1 Skewed coordinate systems & covariant versus contravariant components of two-dimensional vectors

As shown in Fig. 5-2(a) and Fig. 5-2(b), a coordinate system can be either orthogonal (represented by unit vectors $\vec{i}, \vec{j}$) or skewed (represented by unit vectors $\vec{e_1}, \vec{e_2}$). in an orthogonal coordinate system, covariant and contravariant components of a position vector are perfectly align. A position vector in such a system has only one representation i.e. $\vec{r} = x_1\vec{i} + y_1\vec{j}$ (see Fig. 5-2(a)). In a skewed coordinate system, the covariant components (x_1, y_1) and contravariant components (x^1, y^1) of a position vector do not align. The same position vector, in such a system, can be represented using both its covariant and contravariant components. Representing it using the contravariant components is more standard since this representation is compatible with the rule of vector sums, i.e. $\vec{r} = x^1\vec{e_1} + y^1\vec{e_2}$ (see Fig. 5-2(b)). Since $(\vec{e_1} \cdot \vec{e_2}) \neq 0$ in a skewed coordinate system, $r^2 \neq (x^1)^2 + (y^1)^2$ in general. As shown in Fig. 5-2(c), basic trigonometric identities can be used for computing the

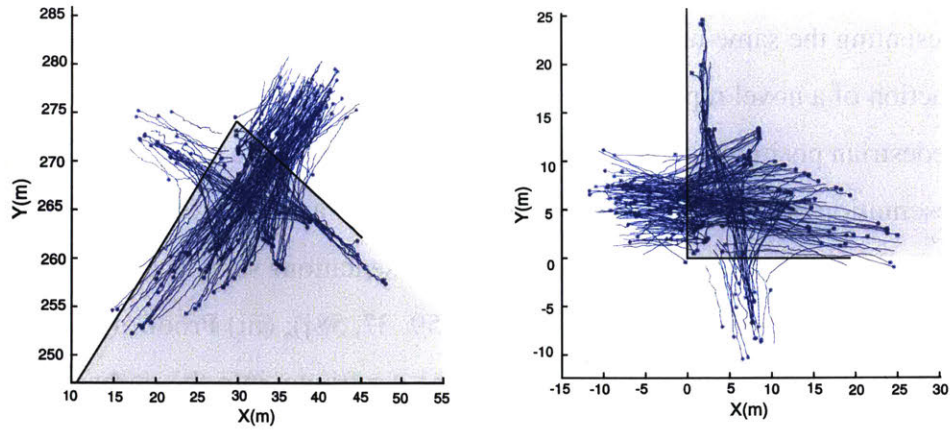


Figure 5-3: Original (left) and transformed trajectories in the curbside coordinate frame (right) under the transformation $\mathscr{T}$, when the curbs are orthogonal to each other. Trajectories are shown in blue and shaded gray area denotes the sidewalk.

contravariant components of a position vector in a skewed coordinate system.

$$x^1 = r\sin(\alpha - \theta)/\sin\alpha \tag{5.1}$$

$$y^1 = r\sin\theta/\sin\alpha \tag{5.2}$$

To meet the objective of pedestrian trajectory prediction in urban intersections, where curbside geometry significantly constraints pedestrian motion, learning motion primitives and their transition in the curbside coordinate frame $X'Y'$, as shown in Fig. 5-1 (instead of an arbitrarily placed, car frame XY, as in [13]), can help improve prediction accuracies because of the addition of context. Furthermore, it can be shown that pedestrian trajectories, when represented using contravariant components of position coordinates in the curbside coordinate frame, undergo an affine transformation across intersections with varying curbside geometries. This aids in developing a context-aware prediction model that can be generalized to any intersection.

5.2 Algorithm

Designing a general, transferable prediction model needs features that are independent of the specific training intersection geometry. In this section, it is shown that any point

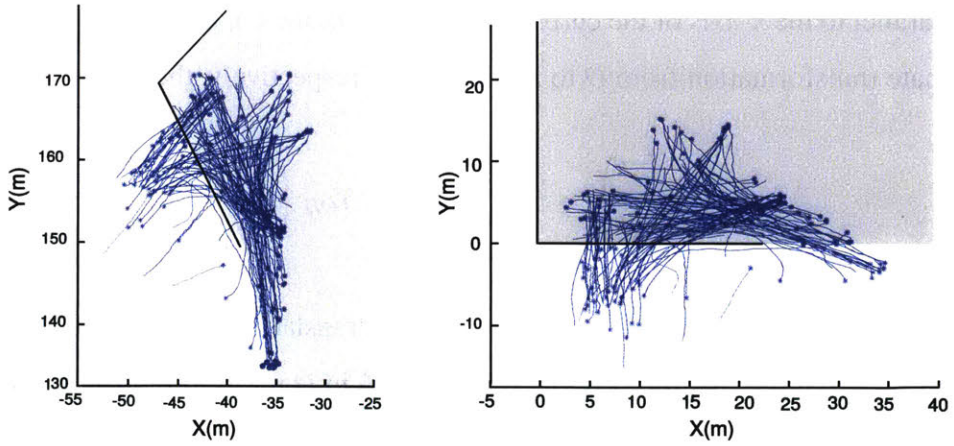


Figure 5-4: Original (left) and transformed trajectories in the curbside coordinate frame (right) under the transformation $\mathscr{T}$, when the curbs are skewed. Trajectories are shown in blue and shaded gray area denotes the sidewalk.

on a pedestrian trajectory, when mapped from the arbitrarily placed car frame, into the common curbside coordinate frame, using its contravariant components, undergoes an affine transformation. The choice of the curbside coordinate frame as the frame in which trajectories are mapped can be justified by the fact that pedestrian trajectories are significantly constrained by curbsides in intersection scenarios. Since an affine transformation preserves properties like collinearity, ratios of distances, parallelism etc., the situational context of pedestrian trajectories i.e. shape and relative distance with respect to curbside, is preserved under this transformation (see Fig. 5-3 and Fig. 5-4).

Definition 1 *A coordinate frame with its origin at the intersection corner of interest, and its axes along the two curbsides intersecting at the chosen corner, is defined as the "curbside coordinate frame" (see Fig. 5-1).*

Definition 2 *Given a point $P(x,y)$ on an observed trajectory t_o in the arbitrarily placed car frame of an intersection (i.e. XY frame in $\boldsymbol{I_1}$ and $\boldsymbol{I_2}$ in Fig. 5-1), let us define a transformation $\mathscr{T} : t_o \rightarrow t_o'$ s.t. $P(x,y) \rightarrow P'(x',y')$, where x', y' are the contravariant components of P' in the curbside coordinate frame.*

Lemma 1 *$\mathscr{T}$ is an affine transformation*

Proof: Given the original, orthogonal, local coordinate system O and an intermediate, helper coordinate system H (also orthogonal but with its origin at the intersection corner and

its x-axis parallel to the x-axis of the curbside coordinate frame C), if T_{OH} and T_{HC} represent the coordinate transformation from O to H and H to C respectively, then $\mathscr{T} = T_{HC}T_{OH}$.

$$\implies \begin{pmatrix} x' \\ y' \end{pmatrix} = \mathscr{T} \begin{pmatrix} x \\ y \end{pmatrix} = T_{HC}T_{OH} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.3}$$

Since, T_{OH} is simply a combination of rotation and translation, it is an affine transformation. Let us now assume that the original point $P(x,y)$ in O maps to $P^*(x^*,y^*)$ in H, such that $(x^*)^2 + (y^*)^2 = r^2$. Note that, by definition, the origin and x-axis of H overlap with the origin and x-axis of C. From Fig. 5-2(c), if θ is the angle made by the position vector with the x-axes,

$$x^* = rcos\theta, y^* = rsin\theta \tag{5.4}$$

Therefore, from (5.1), (5.2) and (5.4), if α is the angle between the intersecting curbsides, $P'(x',y')$ can be written as

$$x' = (r\cos\theta\sin\alpha - r\sin\theta\cos\alpha)/\sin\alpha \tag{5.5}$$

$$\implies x' = x^* - y^*/\tan\alpha \tag{5.6}$$

$$y' = r\sin\theta/\sin\alpha = y^*/sin\alpha \tag{5.7}$$

Note that (5.6), (5.7) can be combined and written in matrix form as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T_{HC} \begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} 1 & -1/tan\alpha \\ 0 & 1/sin\alpha \end{pmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix} \tag{5.8}$$

For intersections with orthogonal curbsides and therefore an orthogonal curbside coordinate frame C, $\alpha = \pi/2$ and T_{HC} is the identity matrix. Since, T_{HC} linearly maps (x^*,y^*) to (x',y'), it is an affine transformation. Furthermore, since T_{OH} and T_{HC} are both affine transformations, $\mathscr{T}$ is also an affine transformation by (5.3).

Since $\mathscr{T}$ is affine, all general properties of an affine transformation hold under $\mathscr{T}$, i.e. (i) collinearity is preserved; (ii) parallel lines remain parallel; (iii) convexity of sets is preserved;

(iv) ratios of distances are preserved i.e. the midpoint of a line segment remains the midpoint of the transformed line segment. Since the objective of this thesis is pedestrian trajectory prediction in urban intersections, which is highly constrained by curbside geometry, transforming pedestrian trajectories into the curbside coordinate frame helps in representing trajectories from different intersection geometries in a common frame. This aids in building a context-aware, general prediction model.

Algorithm 3 describes TASNSC as a transferable version of the ASNSC algorithm that can accurately predict trajectories in unseen intersections with similar semantics as those that it learned on. Given the curbside coordinate vectors $(\vec{e_1}, \vec{e_2})$ of the training intersection, $\mathscr{T}$ is used to map training trajectories from the local, arbitrary placed car frame into the curbside coordinate frame. Motion primitives are then learned in the curbside coordinate frame using ASNSC (line 5). For trajectory prediction in an unseen intersection, first the observed trajectory is transformed into the curbside coordinate frame of the test intersection using $\mathscr{T}$ (line 6). Motion primitives and their transition learned in the common curbside coordinate frame are then used for prediction, followed by a transformation of the predicted trajectory into the original, car frame of the test intersection using $\mathscr{T}^{-1}$ (line 8). Algorithm 2 describes the procedure for transformation of pedestrian trajectories under $\mathscr{T}$. Fig. 5-3 and Fig. 5-4 show the transformation of trajectories into the curbside coordinate frame under $\mathscr{T}$ for an orthogonal and skewed coordinate system respectively.

Algorithm 2: Transformation $\mathscr{T}$

input : curbside unit vectors $((\vec{e_1}, \vec{e_2})$, trajectory in car frame (t_i)
output : transformed trajectory in common curbside frame (t_i')

1 $\alpha \leftarrow cos^{-1}(e_1.e_2)$; // angle of skewed coordinate system
2 **for** $\forall\, P_j(x_j, y_j) \in t_i$ **do**
3 $\quad x_j' \leftarrow \sin(\alpha - \theta)/\sin\alpha$; // refer Fig. 5-2(c), $0 \leq \theta \leq 2\pi$
4 $\quad y_j' \leftarrow \sin\theta/\sin\alpha$
5 **return** $t_i' = \{(x_j', y_j')\}$

Algorithm 3: Transferable ASNSC (TASNSC)

input : curbside unit vectors of training intersection $(\vec{e_1}, \vec{e_2})$, set of training trajectories in car frame $(\mathscr{D}_c)$, curbside unit vectors of test intersection $(\vec{e_1'}, \vec{e_2'})$

output : set of predicted trajectories in car frame (t_p)

/* Training Phase */

$\mathscr{D} = \{\}$

for $\forall\, t_i \in \mathscr{D}_c$ **do**

  $t_i' = \mathscr{T}(\vec{e_1}, \vec{e_2}, t_i)$; // map training trajectories into curbside frame

  $\mathscr{D} \leftarrow \{\mathscr{D}, t_i'\}$

$\{B, S\} = \text{ASNSC}(\mathscr{D})$; // learn motion primitives in curbside frame

/* Prediction Phase */

$t_o' = \mathscr{T}(\vec{e_1'}, \vec{e_2'}, t_o)$; // map observed trajectory into curbside frame

$t_p' = \text{predict}(B, t_o')$; // set of predicted trajectories in curbside frame

$t_p = \mathscr{T}^{-1}(t_p')$

return t_p

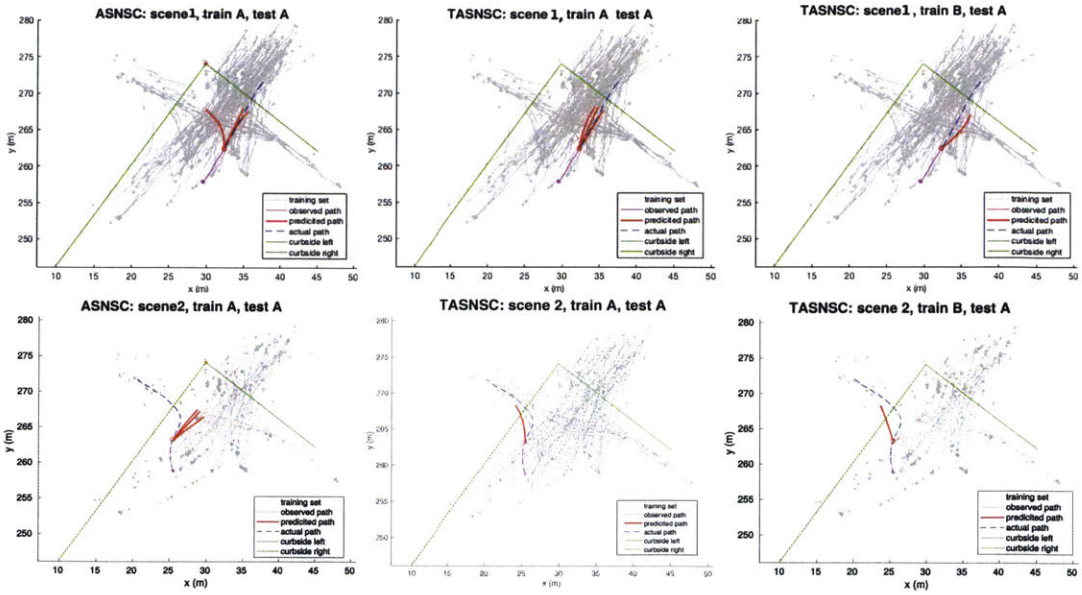


Figure 5-5: Prediction results in $\mathbf{I_1}$ of ASNSC (left), TASNSC trained on $\mathbf{I_1}$ (center) and TASNSC trained on $\mathbf{I_2}$ (right). Ground truth is in dotted blue, observed trajectory in pink & predicted trajectory in red. In the first scenario (first row), a pedestrian approaches the intersection corner, is faced with a choice between two crosswalks and decides to continue moving straight. In the second scenario (second row), another pedestrian approaches the intersection corner and is faced with the same choice, but decides to turn left.

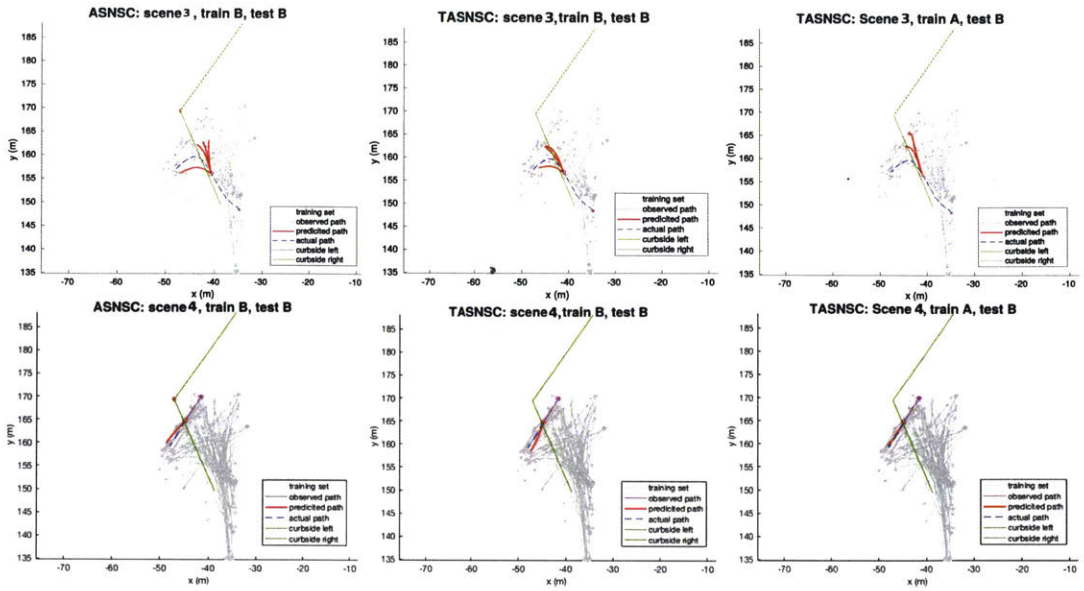


Figure 5-6: Prediction results in $\mathbf{I_2}$ of ASNSC (left), TASNSC trained on $\mathbf{I_2}$ (center) and TASNSC trained on $\mathbf{I_1}$ (right). Ground truth is in dotted blue, observed trajectory in pink & predicted trajectory in red. In the first scenario (first row), a pedestrian exits the curbside and starts walking along the left crosswalk. In the second scenario (second row), a pedestrian approaches the intersection corner, from inside of the sidewalk and continues walking straight to cross the street on the left.

5.3 Results

5.3.1 Dataset description

TASNSC is tested on real pedestrian data collected by a Polaris GEM vehicle equipped with cameras and 2D LiDARs, as described in Chapter 3. A prior occupancy grip map of the environment, created using the on-board LiDARs, is used to extract curbsides. However, as long as the intersection corner is not crowded by obstructions such as trees, it is possible to detect the curbside online. To test the transferable feature of TASNSC, real pedestrian trajectories collected in two intersections, with different curbside geometries (see Fig. 3-2), were used for the experiments included in this chapter. A small subset of the entire dataset from intersection $\mathbf{I_1}$ (with nearly orthogonal curbsides), consisting of 186 training and 32 test trajectories was used. Similarly, a small subset of data collected in intersection $\mathbf{I_2}$ (with skewed curbsides), consisting of 114 training and 22 test trajectories was used. An observation history of 2.5 seconds prior to the pedestrian entering the intersection is used to

predict 5 seconds ahead in time.

5.3.2 Experiment details

Two experiments were conducted for evaluating the prediction performance of TASNSC. In the first experiment, the training and test intersections are the same. While in the second experiment, the training and test intersections are different. The prediction performance of TASNSC in both these experiments is compared with ASNSC, which is used as a baseline. Fig. 5-5 and Fig. 5-6 show a qualitative comparison of prediction performance of TASNSC with ASNSC for both the experiments in intersections $\mathbf{I_1}$ and $\mathbf{I_2}$ respectively. As is clear from the trajectory prediction plots, TASNSC improves prediction performance over ASNSC in all scenarios when trained and tested on the same intersection. Furthermore, TASNSC shows comparable prediction performance with the baseline when trained and tested in different intersections.

5.3.3 Quantitative performance evaluation

Table 5.1 provides a quantitative comparison of TASNSC with ASNSC using two different metrics. The first metric, *classification accuracy* represents the percentage of *correct* predictions (see Fig. 4-4) weighted by their likelihood of prediction. Mathematically, if a set of n trajectories is predicted as $\{\mathbf{t}_1, \ldots, \mathbf{t}_n\}$, with their likelihood of prediction given by $\{l_1, \ldots, l_n\}$, and the *correct* predictions are identified as $\{\mathbf{t}_i\}\ \forall\ i\ \in \mathbf{C} \subset \{1, \ldots, n\}$, the *classification accuracy* is given by:

$$\text{Classification accuracy } \% = \frac{\sum_{i \in \mathbf{C}} l_i}{\sum_{k=1}^{n} l_k} \times 100\%. \tag{5.9}$$

The second metric, *Modified Hausdorff Distance (MHD)* [21] is used to compare predicted trajectories with ground truth. As is clear from the comparison in Table 5.1, TASNSC significantly outperforms ASNSC in *classification accuracy*, while *MHD* of TASNSC is either similar to or better than ASNSC when trained and tested on the same intersection. TASNSC also performs well in the case of different training and test intersections. In those

Algorithm	Classification Accuracy (%)	MHD (m)	Train In	Test In	tr
ASNSC	84.39	2.267	**A**	**A**	N
TASNSC	**90.47**	**2.031**	**A**	**A**	N
TASNSC	79.43	2.557	**B**	**A**	N
TASNSC	81.73	2.284	**B**	**A**	Y
ASNSC	76.94	2.506	**B**	**B**	N
TASNSC	82.79	2.637	**B**	**B**	N
TASNSC	75.92	2.95	**A**	**B**	N
TASNSC	79.51	2.859	**A**	**B**	Y

Table 5.1: Quantitative performance comparison of TASNSC with ASNSC

experiments, adding pedestrian traffic light (shown as 'tr' in the table) as an additional context feature in the GP based transition models [32], boosts prediction performance. Furthermore, the best prediction performance, in terms of both *MHD* and *classification accuracy* is achieved by TASNSC when trained and tested in intersection $\mathbf{I_1}$. This makes sense as the data collected in $\mathbf{I_1}$ is richer in terms of the number of trajectories and variety in maneuvers/behaviors, which leads to better prediction performance, in general, when trained in $\mathbf{I_1}$.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

General Model

As shown in the previous chapter, TASNSC can transfer knowledge from one intersection to another, regardless of the difference in curbside geometries, by learning motion primitives and their transition in the common, curbside coordinate frame. However, urban intersections, at times, can differ significantly in the type of pedestrian behaviors encountered. For instance, TASNSC trained on an intersection with orthogonal curbsides in a college campus would only recognize and predict faster, rule breaking college student-like behaviors. When applied to predict in another intersection, also with orthogonal curbsides, but in a residential area, it will fail to predict novel behaviors such as slower, conservative pedestrians. To account for this limitation, an online prediction model is presented in this chapter, that updates its knowledge base with novel behaviors, as new intersections are presented. Experimental results for the model update step are presented. However, prediction results will be part of future work as setting up the right experiments for demonstrating improvement in prediction performance on the addition of novel behaviors requires data collection in residential/commercial intersections, other than $\mathbf{I_1}$ and $\mathbf{I_2}$, which are both college campus intersections.

6.1 Common Frame for Learning Motion Primitives ($\mathscr{C}$)

A skewed coordinate system with its origin at the intersection corner of interest and axes along the intersecting curbsides, was introduced as the "curbside coordinate frame" in

Chapter 5. Representing trajectories in the curbside coordinate frame, using an affine projection function $\mathscr{T}$, encodes situational context and provides a common frame $\mathscr{C}$ in which spatially dissimilar trajectories from different intersections, with the same intent, would be spatially similar (see Fig. 5-1). However, this holds true for intersections with similar sidewalk width only. To solve this issue, the radial distance from the intersection corner to the point where intersecting sidewalks meet, is used for normalizing trajectories, before mapping them into the common frame $\mathscr{C}$. Such a metric helps account for intersections with different sidewalk widths.

6.2 Algorithm

The objective of this chapter is to build an online model in which motion primitives are continually learned from new data and used to improve the prediction model M. There are two main folds to this approach: (1) learning motion primitives from new training data in the common frame $\mathscr{C}$, and; (2) updating the prediction model $M(l-1) \rightarrow M(l)$, as new knowledge is gained in the l-th round of training. To this end, the proposed algorithm has three main steps: (1) Trajectories received in round l are normalized with respect to sidewalk width, as described in Section 6.1, and then mapped into $\mathscr{C}$, using $\mathscr{T}$. $\mathscr{T}$ is mathematically defined as $\mathscr{T} : t_o \rightarrow t_o{}'$ s.t. any point on trajectory t_o, denoted by $P(x,y) \rightarrow P'(x',y')$, where (x',y') are the contravariant components of normalized pedestrian positions in $\mathscr{C}$. (2) Normalized and transformed trajectories $\mathscr{D}(l)$ are used to learn a new set of motion primitives in $\mathscr{C}$ using ASNSC. (3) Model update: previously learned prediction model is updated using the knowledge obtained from new data.

The remaining section explains the model update step, which consists of updating the existing set of motion primitives B and the set of corresponding GP flow fields U, W. This is followed by an overview of pedestrian trajectory prediction using the proposed model M.

6.2.1 Knowledge/Model Update

As described in Chapter 2, the pedestrian trajectory prediction model M has two components: (1) set of motion primitives B learned using a dictionary learning algorithm, and; (2) sets

of GPs U, W that represent the transition between motion primitives. Updating M requires updating B, U and W.

Updating Motion Primitives

First, pairwise similarity between motion primitives of the existing set B and those of the new set $\bar{B}$ is computed. If a newly learned motion primitive $\bar{\mathbf{m}}_j$ is similar to another, existing motion primitive $\mathbf{m}_i$; the model is updated by replacing $\mathbf{m}_i$ with the fused motion primitive $\mathbf{m}_{ij}$ (see Fig. 6-2(a)). Any novel motion primitives, such as $\bar{\mathbf{m}}_k$ in Fig. 6-2(a) are simply added to the model. Similarity between two motion primitives is mathematically defined as the inverse of distance between them, since, constructing a distance metric is usually easier and more intuitive, for instance Euclidean distance, Manhattan distance, etc.

Distance between motion primitives (D) is defined as the weighted sum of 'overlapping distance' (d^o) and 'heading distance' (d^h). Here, d^o, represents the fraction of non-overlapping 'active cells' between two motion primitives and d^h, represents the cell-wise difference in heading of motion primitives. If $D(\mathbf{m}_i, \mathbf{m}_j)$ is below a pre-defined threshold (γ), $\mathbf{m}_i$ and $\mathbf{m}_j$ are considered similar and fused. Fig. 6-1 shows pairs of similar motion primitives from $\mathbf{I_1}$ and $\mathbf{I_2}$ for .

$$D(\mathbf{m}_i, \mathbf{m}_j) \triangleq \alpha_1 \log\left(d^o(\mathbf{m}_i, \mathbf{m}_j)\right) + \alpha_2 \log\left(d^h(\mathbf{m}_i, \mathbf{m}_j)\right) \tag{6.1}$$

$$d^o(\mathbf{m}_i, \mathbf{m}_j) \triangleq 1 - |A_i \cap A_j| / |A_i \cup A_j| \tag{6.2}$$

$$d^h(\mathbf{m}_i, \mathbf{m}_j) \triangleq \frac{1}{|A_i \cap A_j|} \sum_{k \in A_i \cap A_j} \angle \mathbf{v}_i^k \mathbf{v}_j^k \tag{6.3}$$

Fusion of motion primitives $\mathbf{m}_i$ and $\mathbf{m}_j$ is also a motion primitive $\mathbf{m}_{ij}$, which is mathematically represented by the features A_{ij} and V_{ij} (refer Chapter 2). Adequate fusion must retain unique information from each motion primitive, while simultaneously updating common information. This insight is used to define A_{ij} and V_{ij} as follows. Fig. 6-1 shows fused

motion primitives for pairs of similar motion primitives from $\mathbf{I_1}$ and $\mathbf{I_2}$.

$$A_{ij} \triangleq A_i \cup A_j, \quad \mathbf{v}_{ij}^k \triangleq \begin{cases} 0 & k \notin A_i \cup A_j \\ \mathbf{v}_i^k & k \in A_i, k \notin A_j \\ \mathbf{v}_j^k & k \notin A_i, k \in A_j \\ average(\mathbf{v}_i^k, \mathbf{v}_j^k) & k \in A_i \cap A_j \end{cases} \tag{6.4}$$

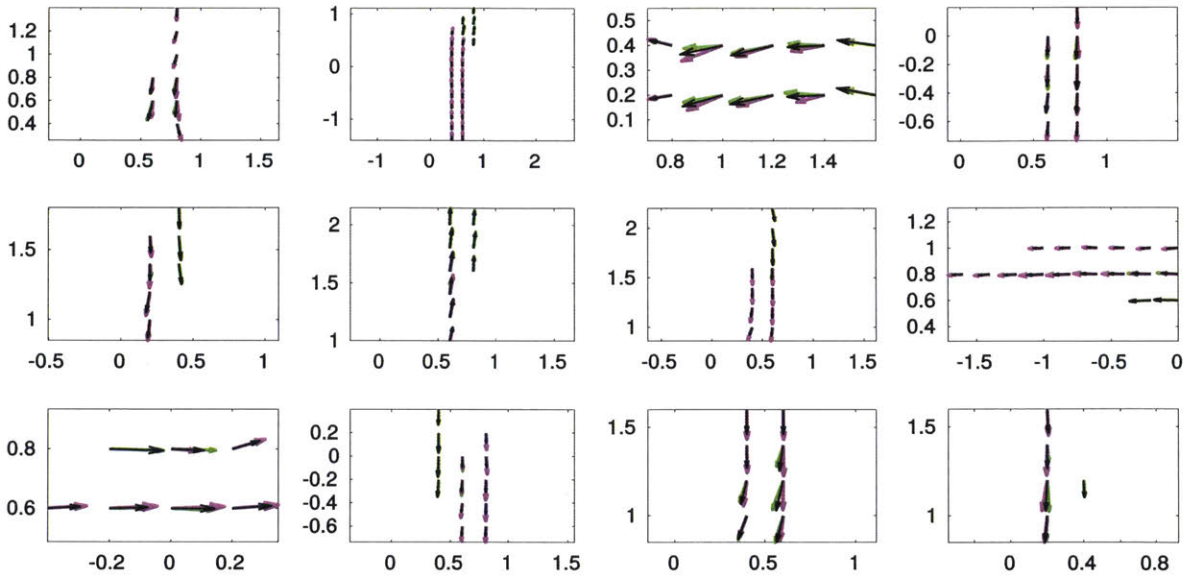
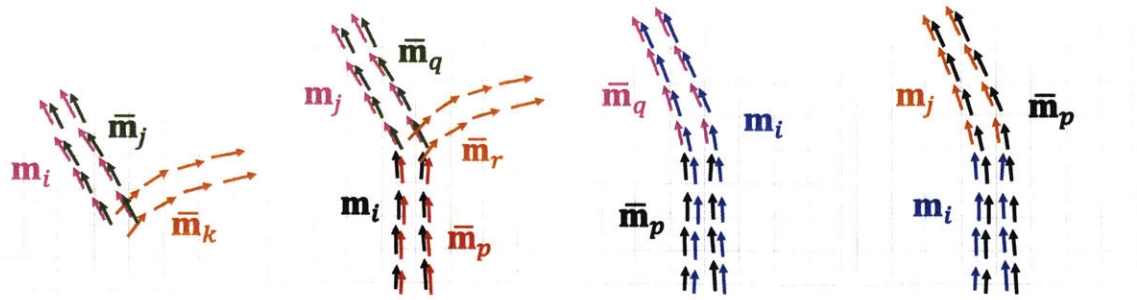


Figure 6-1: Each subplot shows a pair of similar motion primitives from intersection $\mathbf{I_1}$ (in green) and $\mathbf{I_2}$ (in magenta). The fused motion primitive (in black), as described in Section 6.2.1, retains unique information while updating common information. The total number of motion primitives learned from trajectories in $\mathbf{I_1}$ and $\mathbf{I_2}$ is respectively. As shown, motion primitives are similar between the two intersections i.e. they represent similar behaviors or short-term intents.

Updating GPs

Two types of GPs have to be updated as new data comes in: (1) unitary GPs, representing a single motion primitive, and; (2) transition GPs, representing the transition from one motion primitive to another (refer Chapter 2).

Updating unitary GPs: There are two ways in which unitary GPs are updated (see Fig. 6-2(a)): (1) If a newly learned motion primitive $\bar{\mathbf{m}}_j$ is similar to another, existing motion



(a) (b) (c) (d)

Figure 6-2: (a) An illustration to show how U is updated. Here, current model M has a single motion primitive $\mathbf{m}_i$. Motion primitives $\bar{\mathbf{m}}_j, \bar{\mathbf{m}}_k$ are learned from new data. Since $\mathbf{m}_i$ and $\bar{\mathbf{m}}_j$ are similar, they are fused and GP_i^{uni} is updated using new trajectories. Furthermore, since $\bar{\mathbf{m}}_k$ is not similar to any existing motion primitive, $\bar{GP}_k^{uni}$ is added to U; (b) An illustration to show how W is updated. Here, current model M has 2 motion primitives $\mathbf{m}_i, \mathbf{m}_j$. Motion primitives $\bar{\mathbf{m}}_p, \bar{\mathbf{m}}_q, \bar{\mathbf{m}}_r$ are learned from new data. Since $\bar{\mathbf{m}}_p, \bar{\mathbf{m}}_q$ are similar to $\mathbf{m}_i, \mathbf{m}_j$ respectively, GP_{ij}^{trans} is updated. However, since $\bar{\mathbf{m}}_p$ similar to $\mathbf{m}_i$ and it also transitions into $\bar{\mathbf{m}}_r$, $\bar{GP}_{pr}^{trans}$ is added to W; (c) First special case of model update in which new motion primitives $\bar{\mathbf{m}}_p, \bar{\mathbf{m}}_q$ are similar to the same existing motion primitive $\mathbf{m}_i$ s.t. $\bar{\mathbf{T}}(p,q) > 0$; (d) Second special case of model update in which new motion primitive $\bar{\mathbf{m}}_p$ is similar to two existing motion primitives $\mathbf{m}_i, \mathbf{m}_j$ s.t. $\mathbf{T}(i,j) > 0$.

primitive $\mathbf{m}_i$, GP_i^{uni} is updated using Algorithm 5. (2) For any novel motion primitives, such as $\bar{\mathbf{m}}_k$, $\bar{GP}_k^{uni}$ is added to the model.

Updating transition GPs: The transition GPs are updated in the following scenarios (see Fig. 6-2(b)): (1) If an existing pair of motion primitives, for which a valid transition exists i.e. $\mathbf{T}(i,j) > 0$, is similar to a newly learned pair of motion primitives with a valid transition i.e. $\bar{\mathbf{T}}(p,q) > 0$, GP_{ij}^{trans} is updated using Algorithm 5. (2) All other, novel transition GPs, such as $\bar{GP}_{pr}^{trans}$, are simply added to the model.

Two key challenges in updating the unitary and transition GPs and using them for trajectory prediction are: (1) online GP update and; (2) perform prediction using the updated GPs without having to store previous data. To meet these challenges, the online sparse GP algorithm, as in [16] and described in Chapter 2, is used for modeling GPs in this work. The GP parameters α and $\mathbf{C}$ are updated iteratively, in a single pass through the entire training set, for a given maximal size of the $\mathscr{BV}$ set i.e. the set of inducing inputs and outputs. This ensures that as, and when, new trajectories are observed, the GP parameters can be updated by iterating through the new set of trajectories as long as the sufficient statistics $(\alpha, \mathbf{C}, \mathscr{BV})$ are retained.

Algorithm 4: Model Update (Round l)

input : $M(l-1) \triangleq \{B(l-1), U(l-1), W(l-1)\}, \mathscr{D}(l)$

output : $M(l) \triangleq \{B(l), U(l), W(l)\}$

$\{\bar{B}(l), \bar{\mathbf{T}}(l), \bar{\mathbf{R}}(l), \bar{U}(l), \bar{W}(l)\} = \text{ASNSC}(\mathscr{D}(l))$; // learn from new data

$B(l) \leftarrow B(l-1), U(l) \leftarrow U(l-1), W(l) \leftarrow W(l-1)$; // initialization

$S_B = \text{zeros}(\text{sizeOf}(\bar{B}(l)))$, $I_B = \text{zeros}(\text{sizeOf}(\bar{B}(l)))$ **for** $\bar{\mathbf{m}}_j \in \bar{B}(l)$ **do**

  for $\mathbf{m}_i \in B(l-1)$ **do**

    if $D(\mathbf{m}_i, \bar{\mathbf{m}}_j) < \gamma$; // γ is the distance threshold

    then

      $S_B[j] = 1, I_B[j] = i$ $\mathbf{m}_{ij} = \text{fuse}(\mathbf{m}_i, \bar{\mathbf{m}}_j)$; // fuse similar primitives

      $\mathbf{m}_i \leftarrow \mathbf{m}_{ij}$

      $GP_i^{uni} = \text{GP\_update}(GP_i^{uni}, i, i, \mathscr{D}(l), \bar{\mathbf{R}}(l))$; // Algorithm 5

for $\bar{\mathbf{m}}_k \in \bar{B}(l)$ **do**

  if $S_B[k] == 0$; // check if novel primitive

  then

    $B(l) \leftarrow \{B(l), \bar{\mathbf{m}}_k\}$, $U(l) \leftarrow \{U(l), \bar{GP}_k^{uni}\}$; // add primitive/unitary GP

for $\bar{\mathbf{m}}_p \in \bar{B}(l)$ **do**

  for $\bar{\mathbf{m}}_q \in \bar{B}(l)$ **do**

    if $p \neq q \wedge \bar{\mathbf{T}}(l)(p,q) > 0$; // loop through all transitions in $\mathscr{D}(l)$

    then

      if $S_B[p] == 1 \wedge S_B[q] == 1$; // check if $\exists$ similar transition GP

      then

        $i = I_B[p]$, $j = I_B[q]$ $GP_{ij}^{trans} = \text{GP\_update}(GP_{ij}^{trans}, i, j, \mathscr{D}(l), \bar{\mathbf{R}}(l))$; // Algorithm 5

      else if $S_B[p] == 1 \wedge S_B[q] == 0$; // check if novel transition GP

      then

        $W(l) \leftarrow \{W(l), \bar{GP}_{pq}^{trans}\}$

      else if $S_B[p] == 0 \wedge S_B[q] == 0$; // check if novel transition GP

      then

        $W(l) \leftarrow \{W(l), \bar{GP}_{pq}^{trans}\}$

return $\{B(l), U(l), W(l)\}$

Algorithm 5: GP_update

input : $GP_{current}, i, j, \mathscr{D}(l), \bar{\mathbf{R}}(l)$
output : $GP_{updated}$

$S = \bar{\mathbf{R}}(l)(i,j)$; // set of trajectories transitioning from $\bar{\mathbf{m}}_i$ to $\bar{\mathbf{m}}_j$/ending in $\bar{\mathbf{m}}_i$
$n_{traj} = \text{sizeOf}(S)$ **for** $k = 1 : n_{traj}$ **do**
  $GP_{updated} = \text{Csato\_sparse\_GP}(GP_{current}, S\{k\})$; // from [16]
  $GP_{current} \leftarrow GP_{updated}$
return $GP_{updated}$

6.2.2 Trajectory Prediction using M

First, the observed trajectory is normalized with respect to sidewalk width of the test intersection and mapped into $\mathscr{C}$ using $\mathscr{T}$. Future trajectory is then predicted using M, which consists of the following steps: (1) Find $GP_i^{uni} \in U$ which best explains the observed trajectory; (2) Find all $GP_{ij}^{trans} \in W$ s.t. $\mathbf{T}(i,j) > 0$ and use them for predicting a set of future trajectories along with the likelihood of each. $\mathscr{T}^{-1}$, which is the inverse of the projection function $\mathscr{T}$, is then used to map the predicted set of trajectories, from the common frame $\mathscr{C}$ into the original, test intersection.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Conclusion and Future Work

This thesis presents a general, context-aware, long term (order of few seconds) trajectory prediction model applicable to pedestrians in urban intersections. The proposed prediction model is built in three steps and comprises of a set of motion primitives, learned using sparse coding, and the pair-wise transition between motion primitives, modeled as motion patterns using two-dimensional GP flow fields. [33, 2]

First, context is embedded into the previously published ASNSC framework [13, 12], to build the CASNSC algorithm. CASNSC utilizes context features, such as pedestrian traffic lights, curbside orientation and relative distance to curbside, in modeling the pair-wise transition between motion primitives. It achieves 12.5% improvement in prediction accuracy when compared with the prediction performance of the baseline, ASNSC on a subset of pedestrian trajectories collected at intersection $\mathbf{I_1}$, with nearly orthogonal curbsides.

An important limitation of CASNSC that prevents it from transferring knowledge from one intersection to another, is the use of spatial features, such as pedestrian position and orientation in the car frame, to learn motion primitives. TASNSC addresses this limitation by learning motion primitives and their transition in a common, curbside coordinate frame instead of the intersection specific car frame. This helps build a more superior model that can transfer knowledge from one intersection to another, regardless of the difference in curbside geometries (skewed versus orthogonal). TASNSC achieves 7.2% improvement in

prediction accuracy over ASNSC, when trained and tested on the same intersection because of the implicit embedding of context due to learning motion primitives and their transition in the curbside coordinate frame. A comparable prediction performance with the baseline is achieved when trained and tested on different intersections. Subsets of pedestrian trajectories from intersections $\mathbf{I_1}$ (with nearly orthogonal curbsides) and intersection $\mathbf{I_2}$ (with skewed curbsides) are used for evaluating the prediction performance of TASNSC.

Lastly, while TASNSC can successfully transfer knowledge from one intersection to another, it cannot account for novel behaviors encountered as more intersections are visited. An online model, based on TASNSC, is presented to account for these novel behaviors by updating the prediction model as, and when, new data is collected. A novel similarity metric is defined and used to compute the pair-wise similarity between existing and new motion primitives. Any similar motion primitives are fused with the existing ones and the corresponding unitary and transition GPs are updated. All novel primitives and their corresponding transition GPs are added to the model. While the algorithm is explained in detail in this thesis, including results of the model update steps, prediction performance improvement results, on real data, will be part of future work.

One might argue that the best prediction accuracies ($\approx 91\%$) obtained by model(s) proposed in this thesis are not good enough in an absolute sense, as compared to state-of-art deep learning based techniques, which are capable of achieving higher absolute prediction accuracies. However, note that the objective of the prediction models proposed in this thesis is to output a set of all possible future trajectories, along with the likelihood of each, instead of a single future trajectory as provided by other prior works with better accuracy numbers. Such a prediction output is desired to account for uncertainty in prediction and can be easily incorporated by state-of-art probabilistic planners ([35, 17, 10, 11]).

Two important limitations of the prediction model(s) presented in this thesis is their inability to recognize stopping intent and learn full trajectories as motion primitives, if the training data predominantly consists of a single pedestrian behavior. The former limitation

affects prediction accuracies in cases where the pedestrian stops to wait at an intersection corner for a change in pedestrian traffic light status or for the crosswalk to clear out in the absence of lights. The latter limitations affects the online model the most, since bad motion primitives to start off with only make the model worse over time, as more data is collected. An online sparse coding based technique [41] for learning motion primitives might be better to deal with such scenarios.

Other limitations include the failure to incorporate crosswalk location in developing the common curbside frame. Along with curbside geometries, there can be intersection scenarios across which difference in crosswalk location and angle with respect to curbsides also plays a significant role in building a cohesive, common frame.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] Georges S Aoude, Brandon D Luders, Joshua M Joseph, Nicholas Roy, and Jonathan P How. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Autonomous Robots*, 35(1):51–76, 2013.

[3] Saeed Asadi Bagloee, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of Modern Transportation*, 24(4):284–303, Dec 2016.

[4] Lamberto Ballan, Francesco Castaldo, Alexandre Alahi, Francesco Palmieri, and Silvio Savarese. Knowledge transfer for scene-specific motion prediction. In *European Conference on Computer Vision*, pages 697–713. Springer, 2016.

[5] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2903–2910. IEEE, 2012.

[6] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.

[7] Alessandro Bissacco and Stefano Soatto. Hybrid dynamical models of human motion for the recognition of human gaits. *International journal of computer vision*, 85(1):101–114, 2009.

[8] Sarah Bonnin, Thomas H Weisswange, Franz Kummert, and Jens Schmüdderich. Pedestrian crossing prediction using multiple context-based models. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 378–385. IEEE, 2014.

[9] Trevor Campbell, Miao Liu, Brian Kulis, Jonathan P How, and Lawrence Carin. Dynamic clustering via asymptotics of the dependent dirichlet process mixture. In *Advances in Neural Information Processing Systems*, pages 449–457, 2013.

[10] Ashwin Mark Carvalho. *Predictive Control under Uncertainty for Safe Autonomous Driving: Integrating Data-Driven Forecasts with Control Design*. PhD thesis, UC Berkeley, 2016.

[11] Gianluca Cesari, Georg Schildbach, Ashwin Carvalho, and Francesco Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine*, 9(3):23–35, 2017.

[12] Yu Fan Chen. Predictive Modeling and Socially Aware Motion Planning in Dynamic , Uncertain Environments by. *Thesis*, 2017.

[13] Yu Fan Chen, Miao Liu, and Jonathan P How. Augmented dictionary learning for motion prediction. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2527–2534. IEEE, 2016.

[14] Patrick Pakyan Choi and Martial Hebert. Learning and predicting moving object trajectory: a piecewise trajectory segment approach. *Robotics Institute*, page 337, 2006.

[15] Pasquale Coscia, Francesco Castaldo, Francesco AN Palmieri, Alexandre Alahi, Silvio Savarese, and Lamberto Ballan. Long-term path prediction in urban scenarios using circular distributions. *Image and Vision Computing*, 69:81–91, 2018.

[16] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

[17] Alexander G Cunningham, Enric Galceran, Ryan M Eustice, and Edwin Olson. Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1670–1677. IEEE, 2015.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[19] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009.

[20] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2012.

[21] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 566–568. IEEE, 1994.

[22] A. Ess, B. Leibe, K. Schindler, , and L. van Gool. A mobile vision system for robust multi-person tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*. IEEE Press, June 2008.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[24] Daniel J Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[25] Sarah Ferguson, Brandon Luders, Robert C Grande, and Jonathan P How. Real-time predictive modeling and robust avoidance of pedestrians with uncertain, changing intentions. In *Algorithmic Foundations of Robotics XI*, pages 161–177. Springer, 2015.

[26] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[27] Michael Goldhammer, Matthias Gerhard, Stefan Zernetsch, Konrad Doll, and Ulrich Brunsmann. Early prediction of a pedestrian's trajectory at intersections. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 237–242. IEEE, 2013.

[28] Avelino J Gonzalez, William J Gerber, Ronald F DeMara, and Michael Georgiopoulos. Context-driven near-term intention recognition. *The Journal of Defense Modeling and Simulation*, 1(3):153–170, 2004.

[29] David J Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009.

[30] Andrés Michael Levering Hasfura. *Pedestrian detection and tracking for mobility on demand*. PhD thesis, Massachusetts Institute of Technology, 2016.

[31] Henry O Jacobs, Owen K Hughes, Matthew Johnson-Roberson, and Ram Vasudevan. Real-time certified probabilistic pedestrian forecasting. *IEEE Robotics and Automation Letters*, 2(4):2064–2071, 2017.

[32] Nikita Jaipuria, Golnaz Habibi, and Jonathan P. How. Casnsc: A context-based approach for accurate pedestrian motion prediction at intersections. In *NIPS Machine Learning for Intelligent Transportation Systems Workshop (MLITS)*, 2017.

[33] Joshua Joseph, Finale Doshi-Velez, Albert S Huang, and Nicholas Roy. A bayesian nonparametric approach to modeling motion patterns. *Autonomous Robots*, 31(4):383, 2011.

[34] Vasiliy Karasev, Alper Ayvaci, Bernd Heisele, and Stefano Soatto. Intent-aware long-term prediction of pedestrian motion. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2543–2549. IEEE, 2016.

[35] Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 60:416–442, 2015.

[36] Arne Kesting, Martin Treiber, and Dirk Helbing. Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1928):4585–4605, 2010.

[37] Julian Francisco Pieter Kooij, Nicolas Schneider, Fabian Flohr, and Dariu M Gavrila. Context-based pedestrian path prediction. In *European Conference on Computer Vision*, pages 618–633. Springer, 2014.

[38] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 593–604. ACM, 2007.

[39] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1(1):1, 2014.

[40] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[41] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[42] Dimitrios Makris and Tim Ellis. Spatial and probabilistic modelling of pedestrian behaviour. In *British Machine Vision Conference 2002, vol. 2*. Citeseer, 2002.

[43] Justin Miller, Andres Hasfura, Shih-Yuan Liu, and Jonathan P How. Dynamic arrival rate estimation for campus mobility on demand network graphs. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2285–2292. IEEE, 2016.

[44] Justin Miller and Jonathan P How. Predictive positioning and quality of service ridesharing for campus mobility on demand systems. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1402–1408. IEEE, 2017.

[45] Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

[46] Carl E Rasmussen and Zoubin Ghahramani. Infinite mixtures of gaussian process experts. In *Advances in neural information processing systems*, pages 881–888, 2002.

[47] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[48] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. *arXiv preprint arXiv:1711.10061*, 2017.

[49] Friederike Schneemann and Patrick Heinemann. Context-based detection of pedestrian crossing intention for autonomous driving in urban environments. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2243–2248. IEEE, 2016.

[50] Andreas T Schulz and Rainer Stiefelhagen. A controlled interactive multiple model filter for combined pedestrian intention recognition and path prediction. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 173–178. IEEE, 2015.

[51] Andreas Th Schulz and Rainer Stiefelhagen. Pedestrian intention recognition using latent-dynamic conditional random fields. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 622–627. IEEE, 2015.

[52] M. Shen, G. Habibi, and J. P. How. Transferable Pedestrian Motion Prediction Models at Intersections. *ArXiv e-prints*, March 2018.

[53] Kevin Spieser, Kyle Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone. Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore. In *Road Vehicle Automation*, pages 229–245. Springer, 2014.

[54] Cynthia Sung, Dan Feldman, and Daniela Rus. Trajectory clustering for motion prediction. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1547–1552. IEEE, 2012.

[55] Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4):429–436, 2006.

[56] Dizan Vasquez, Thierry Fraichard, and Christian Laugier. Incremental learning of statistical motion patterns with growing hidden markov models. *IEEE Transactions on Intelligent Transportation Systems*, 10(3):403–416, 2009.

[57] Christian Vollmer, Sven Hellbach, Julian Eggert, and Horst-Michael Gross. Sparse coding of human motion trajectories with non-negative matrix factorization. *Neurocomputing*, 124:22–32, 2014.

[58] Benjamin Völz, Karsten Behrendt, Holger Mielenz, Igor Gilitschenski, Roland Siegwart, and Juan Nieto. A data-driven approach for pedestrian intention estimation. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 2607–2612. IEEE, 2016.

[59] Benjamin Völz, Holger Mielenz, Gabriel Agamennoni, and Roland Siegwart. Feature relevance estimation for learning pedestrian behavior at crosswalks. In *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pages 854–860. IEEE, 2015.

[60] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *ArXiv e-prints*, May 2018.