Semantic Description of Activities in Videos

Fillipe Dias Moreira De Souza
University of South Florida, fdms8@gmail.com

Follow this and additional works at: http://scholarcommons.usf.edu/etd
Part of the Applied Mathematics Commons, and the Artificial Intelligence and Robotics Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.
 Semantic Description of Activities in Videos 

 by 

 Fillipe Dias Moreira de Souza 

 A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy Department of Computer Science and Engineering College of Engineering University of South Florida 

 Co-Major Professor: Sudeep Sarkar, Ph.D. 
 Co-Major Professor: Anuj Srivastava, Ph.D. 
 Dmitry Goldgof, Ph.D. 
 Rangachar Kasturi, Ph.D. 
 Rajiv Dubey, Ph.D. 
 Andrew Raij, Ph.D. 

 Date of Approval: 
 March 29, 2017 

 Keywords: Pattern Theory, Video Analysis, Activity Recognition, Graphical Models, Compositional Approach 

 Copyright © 2017, Fillipe Dias Moreira de Souza
DEDICATION

I dedicate this work to my mother Luciene Dias Sacramento de Souza, my father Fernando Antonio Moreira de Souza, my sister Fernanda Dias Moreira de Souza, my godparents and all the people that in one way or another believed in my potential and supported my choices. This dedication goes specially to my mentors and friends. I also dedicate this work to the inventor of General Pattern Theory, Ulf Granander, in my humble attempt to honor his original ideas. It has been an interesting and enjoyable journey...
ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisors Professor Sudeep Sarkar and Professor Anuj Srivastava, who had the attitude and the substance of geniuses: they continually and convincingly conveyed a spirit of adventure and innovative perspectives in regard to research and entrepreneurship. I also thank them for having introduced me to flourishing perspective of solving problems using Grenander’s General Pattern Theory and Monte Carlo methods. Without their guidance, encouragement and persistent help this dissertation, awards and my accomplishments would not have been possible. I am forever grateful and indebted for their intellectual loans.

I would like to thank my committee members Professor Dmitry Goldgof and Professor Rangachar Kasturi for their continuous support and invaluable advice for the completion of a successful PhD and my future pursuits. I would also like to thank my committee members Professor Rajiv Dubey and Dr. Andrew Raij for their support, sometimes even at short notice.

Special thanks go to Matt Galvin for being supportive and my mentor during the NSF I-CORPS program. I would like to thank Richard Berman, Dr. Glenn Whichard and Valerie McDevitt for their support for my participation in the USF I-Corps site as a teaching assistant and the national NSF I-Corps program as an entrepreneurial lead. I would like to thank Dr. Sanjukta Bhanja and Dr. Srinivas Katkoori for giving me advice and support to participating in the NSF I-Corps program and becoming an IEEE CS Student Chapter president.
In addition, a great thank you to Professor Miguel Labrador who, since my first day as a PhD student in the CSE department, has been so supportive of me. I would also like to thank all other faculty members of the CSE department whose names were not mentioned here for their hard work in teaching and assisting us, students, even out of their office hours.

Special thanks to my parents Fernando Souza and Luciene Souza, my sister Fernanda Souza, my godparents Joao Carlos and Jobette and my cousins for their love, invaluable support and encouragement to help me pursue and complete my PhD program. I would also like to thank other special people that I met along the way and helped me have a balanced life throughout this program, namely, Laurie Ferguson, Linsey Ferguson, Erin Symonds, Viviane Kretli and their family members.

I shall not forget to express my gratitude to all my friends inside and outside the research lab for the stimulating conversations and the rare happy hours that actually happened to give me some relaxation time! I very much appreciate the support from Matthew Shreve, Ravi Panchumarthy and Sergiy Fefilatyev given with, not only friendship but also, job and internship referrals. I am also thanking Mauricio Pamplona, Luis Jaimes, Tiago Cunha, Sandra Avila, Rajmadhan Ekambaram, Ravikiran, Ravi Subramanian and Santosh Aditham for their invaluable support and great conversations. I would also like to thank all my friends whose names were not mentioned but in one way or another influenced me on this journey.

I would like to thank Yvette Blanchard, Kim Bluemer, Theresa Collins, Kelly Schuler, Laura Owczarek, Gabriela Franco, Shanie Lightbourne, Victoria Stuart and Catherine Burton from Computer Science and Engineering department for their patience and support throughout my Ph.D.
I specially thank my former academic advisors Professor Adriano Hoth, Professor Eduardo Valle, Professor Guillermo Camara and Professor Arnaldo Albuquerque for having always believed in my determination and skills and having provided the adequate support to pursue higher education programs. Finally, I would also like to thank the National Science Foundation (NSF) for their financial support throughout my PhD program.
# TABLE OF CONTENTS

LIST OF TABLES iii  
LIST OF FIGURES iv  
ABSTRACT vi  

CHAPTER 1 : INTRODUCTION 1  
1.1 Problem Statement 3  
1.2 Challenges and Motivation 5  
1.3 Our Approach 6  
1.4 Contributions 8  
1.5 Dissertation Overview 8  

CHAPTER 2 : PRIOR WORK ON VIDEO DESCRIPTION 10  
2.1 Implicit Models 12  
2.2 Explicit Models 14  
2.3 Limitations 15  

CHAPTER 3 : HUMAN ACTIVITY REPRESENTATION WITH PATTERN THEORY 17  
3.1 Constructing a Pattern Theory Framework 17  
3.1.1 Generators and Bonds 18  
3.1.2 Connectors and Configurations 22  
3.1.3 Regularity of Configurations 24  

CHAPTER 4 : GENERATING SEMANTIC DESCRIPTIONS FOR VIDEOS 28  
4.1 Operations on the Configuration Space $C(R)$ 29  
4.2 Bond Quantification 30  
4.3 Probabilistic Superstructure on $C(R)$ 33  
4.4 Learning 34  
4.5 Inference 35  
4.5.1 MCMC-Simulated Annealing 36  
4.5.2 Global Proposal Function 39  
4.5.3 Local Proposal Function 39  
4.6 Time and Space Complexities 40  
4.7 Differences with Other Graphical Models 41  

CHAPTER 5 : EXPERIMENTAL FRAMEWORK, FIRST RESULTS 45  
5.1 Data 46  
5.1.1 YouCook Dataset 47  
5.1.2 Breakfast Actions Dataset 47  
5.2 Training 48  
5.3 Features 48  
5.4 Baseline Algorithms 49  
5.5 Evaluation Metrics 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 6: Spatially Coherent Video Descriptions</td>
<td>58</td>
</tr>
<tr>
<td>6.1 Overview</td>
<td>58</td>
</tr>
<tr>
<td>6.2 Spatial Coherence Energy Term</td>
<td>60</td>
</tr>
<tr>
<td>6.3 Handling Clutter of Objects</td>
<td>60</td>
</tr>
<tr>
<td>6.3.1 Spatially Coherent Interpretations under Clutter of Objects</td>
<td>62</td>
</tr>
<tr>
<td>6.3.2 Experimental Results and Discussion</td>
<td>64</td>
</tr>
<tr>
<td>6.4 Handling Multiple Occurrence of Activities</td>
<td>65</td>
</tr>
<tr>
<td>6.4.1 Video Units with Multiple Occurring Activities</td>
<td>70</td>
</tr>
<tr>
<td>6.4.2 Experimental Results and Discussion</td>
<td>70</td>
</tr>
<tr>
<td>6.5 Summary and Discussion</td>
<td>73</td>
</tr>
</tbody>
</table>

| Chapter 7: Temporally Coherent Descriptions for Long Videos | 74 |
| 7.1 Overview | 75 |
| 7.2 Modeling Long Video Sequences with Temporal Bonds | 78 |
| 7.2.1 Temporal Bonds | 78 |
| 7.2.2 Interpretations for Sequence of Activities | 80 |
| 7.3 Experiments with Synthetic Classifiers | 84 |
| 7.4 Experiments with Real Classifiers | 85 |
| 7.5 Running Time | 88 |
| 7.6 Summary and Discussion | 88 |

| Chapter 8: Interpretation of Activities with Multiple Modalities | 90 |
| 8.1 Overview | 90 |
| 8.2 Integrating Data from Multiple Modalities into the Model | 91 |
| 8.3 Experimental Results and Discussion | 93 |
| 8.3.1 Video Descriptions based on Discriminant Models | 94 |
| 8.3.2 Video Descriptions as Structured Output on Deep Features | 95 |
| 8.3.3 Video Descriptions Built on Sound and Vision | 97 |
| 8.3.4 Improving Video Descriptions with Grenander’s Structure | 98 |
| 8.4 Summary and Discussion | 99 |

| Chapter 9: Conclusions | 104 |
| References | 105 |
| Appendix A: Copyright Clearance Forms | 110 |
| About the Author | END PAGE |
LIST OF TABLES

Table 2.1  Summary of related work on interpretation of activities in videos. 13
Table 3.1  Cooking domain generator space $G$ sample. 21
Table 3.2  The equivalence classes $G^\alpha$. 22
Table 3.3  The truth-valued table by $\rho(\beta^\prime_{\text{out}}, \beta^\prime_{\text{in}})$. 26
Table 6.1  Performance comparison for simultaneously occurring events. 71
Table 7.1  Average performance of interpretations for increasing temporal windows. 87
Table 7.2  Average CPU+I/O time of videos per number of feature generators $n_f$. 88
Table 8.1  Interpretation performance with sound and vision features. 98
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Example of a video description for the activity of a cooking video.</td>
<td>4</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Example of generator space for cooking domain arranged into levels.</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Example of a video interpretation using elements of pattern theory.</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Connector graphs and their configurations.</td>
<td>24</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>Illustration of how transformation change configurations.</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Illustration of the notion of spatial coherence.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>First two steps of the global proposal function.</td>
<td>43</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Final step of the global proposal function.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>These figures illustrate how the local proposal function works.</td>
<td>44</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>Overview of using the pattern theory framework for video description.</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Density of the sample of interpretations during the inference process.</td>
<td>52</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Accepted proposals at different iterations.</td>
<td>53</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>How soon until the best interpretation is found?</td>
<td>53</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Performance rates for different degradation levels.</td>
<td>56</td>
</tr>
<tr>
<td>Figure 5.6</td>
<td>Robustness to degrading classification errors of labels.</td>
<td>57</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>Performance results on the YouCook and Breakfast Actions data sets.</td>
<td>59</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>Illustration of where the spatial constraint quantification come from.</td>
<td>61</td>
</tr>
<tr>
<td>Figure 6.3</td>
<td>Robustness to different amounts of clutter of objects.</td>
<td>62</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>(a) Example video depicting the interaction “pick up meat from the plate”.</td>
<td>66</td>
</tr>
<tr>
<td>Figure 6.5</td>
<td>Example 1 of interpretation under object clutter scenario.</td>
<td>67</td>
</tr>
</tbody>
</table>
Figure 6.6  Example 2 of interpretation under object clutter scenario.  68
Figure 6.7  Example 3 of interpretation under object clutter scenario.  69
Figure 6.8  Performance rates of interpretations for simultaneously occurring events.  70
Figure 6.9  Scenarios of simultaneously occurring events.  72
Figure 7.1  Overview of the pattern theoretic framework proposed in [1].  76
Figure 7.2  Illustration of advantage in using temporal bonds.  77
Figure 7.3  Results of interpretations for sequences of two activities.  81
Figure 7.4  Result for a sequence of four consecutive activities (making salad).  82
Figure 7.5  Result for a sequence of four consecutive activities (making sandwich).  83
Figure 7.6  Interpretation performance for varying scenarios of classification error.  85
Figure 7.7  Robustness to decreasing classification score ranks for correct label.  86
Figure 7.8  Influence of bond types on the performance quality of interpretations.  87
Figure 7.9  Performance rates for the top \( k \) best interpretations.  87
Figure 8.1  Video description framework for multimodal data.  91
Figure 8.2  Overview of the proposed approach.  92
Figure 8.3  Interpretations with deep features and Grenander's structures.  96
Figure 8.4  Comparative interpretation performance with traditional and deep features.  96
Figure 8.5  Histogram of correct labels using CNN, CNNFLOW and spectrogram features.  101
Figure 8.6  Histogram of correct labels using CNN and spectrogram features.  102
Figure 8.7  Comparing interpretations based on deep features.  103
ABSTRACT

Description of human activities in videos results not only in detection of actions and objects but also in identification of their active semantic relationships in the scene. Towards this broader goal, we present a combinatorial approach that assumes availability of algorithms for detecting and labeling objects and actions, albeit with some errors. Given these uncertain labels and detected objects, we link them into interpretative structures using domain knowledge encoded with concepts of Grenander’s general pattern theory. Here a semantic video description is built using basic units, termed generators, that represent labels of objects or actions. These generators have multiple out-bonds, each associated with either a type of domain semantics, spatial constraints, temporal constraints or image/video evidence. Generators combine between each other, according to a set of pre-defined combination rules that capture domain semantics, to form larger structures known as configurations, which here will be used to represent video descriptions. Such connected structures of generators are called configurations. This framework offers a powerful representational scheme for its flexibility in spanning a space of interpretative structures (configurations) of varying sizes and structural complexity. We impose a probability distribution on the configuration space, with inferences generated using a Markov Chain Monte Carlo-based simulated annealing algorithm. The primary advantage of the approach is that it handles known computer vision challenges – appearance variability, errors in object label annotation, object clutter, simultaneous events, temporal dependency encoding, etc. – without the need for a exponentially-large (labeled) training data set.
CHAPTER 1 : INTRODUCTION

This research work addresses the problem of describing activities in videos from the point of view of computer vision, with mathematical ground on general pattern theory. Automated description of video content is an important capability for a very diverse set of interestingly useful applications [2, 3, 4, 5, 6, 7]. The authors of this work have personally done customer discovery and identified a number of potential customer segments that can greatly benefit from this technology. For instance,

1. Cable TV companies are currently looking into ways to automate and improve the customer experience with advertisements pushed between TV shows and on on-demand video applications. The video description technology would permit to push more relevant ads for specific consumers and, as a result, increase impression rates – high impression rates are strongly associated with increased sales. Additionally, cable TV companies are interested in attracting more corporate clients to advertise products and services with them; therefore, increasing the possibility of higher revenues.

2. Video editors from the media production business are seeking solutions to reduce the amount of time taken to search for content-specific video snippets from raw footage. Additionally, there is an interesting in generating first. This could potentially help them save 20-30% of their work time on a single project.
3. Video surveillance systems have had positive impact in reducing crime rates of public areas of cities like Baltimore and Chicago, even cost savings associated with crimes. Data analysis on the events captured in surveillance videos can help cities implement better security measures to ensure public safety.

The first step towards the development of solutions for the applications mentioned above is to design algorithms that can describe the content of videos by identifying events of interest that are occurring in a video scene. The goal was to design a method that is mathematically grounded and permits us to surpass a number of limitations by current solutions used to address the problem of interpretation of activities in videos. Events compose and describe the semantic content of a video scene. In the context of the present work, an event is a human activity such as “pouring water in a cup” or “adding sugar to coffee”. Different domains will typically require different semantic models; for example, in the security surveillance domain, we are concerned with human activities such as “man carrying backpack is crossing the street” or “man opening a car trunk”.

In the literature, many have attempted to simplify the problem of interpretation of activities in videos by tackling it as a video classification problem. When this is the case, one must predefine all possible categories (a fixed number of labels) of interest to label a video; for instance, “birthday party”, “making a sandwich”, “opening car door”, “cracking eggs into a pan using knife”, “pouring eggs and milk in a bowl’, to name a few. In this simplified view, one needs to train a classification model for each single category using video data – this assumes that sufficient number of training instances are available. These approaches do not generalize well for there can be a very large number of categories that need to be enumerated. Additionally, it does
not account for all possible variations in which a particular category can appear. Others have attempted to derive descriptions from key words inferred from a joint probability distribution of features and words from text. The sentences are then validated by some post semantic verification process; however, they cannot avoid errors due to clutter of information in first inference step. They implicitly assume that the underlying statistical model employed handles uncertainty due to these sorts of complexities. In this work, we devised a video content description framework that is sufficiently comprehensive to overcome all these limitations in a single formalism using the language of pattern theory.

1.1 Problem Statement

The main difference between video classification [8] and video description [9] is that in the former both the number of labels and the labels themselves are required to be known. Each label typically represents an event of a human activity [10]. If a specific event happens to occur in a form that it is not anticipated or the event has not thought of as a specific label from the predefined set of labels, then the video classification method will fail by assigning some label for which it maximizes the posterior. Video description, on the other hand, is typically thought to be domain independent and follows a free form of expressing the occurring events from a video instead of relying on a predefined set of labels. Its free form of semantic expression includes a variable number of semantic relationships allowed in a description.

Formally, the problem of video description can be cast as follows: Given a video, provide a textual description of semantic value that describes the main relevant events in the scene (see Figure 1.1). Typically, and in this work, we refer to description of human activities. Human activities is typically characterized by a person performing an action that results in a single or multiple
human-object interaction(s). Semantic descriptions of human activities involves both: (1) detection of the ongoing actions and objects and (2) a spatially and temporally coherent explanation of the semantic relationships between the detected actions and objects in a specific scene. This is different from and harder than video activity classification where the focus is solely on labeling the video with a human activity category based on the decision of a discriminative function. A description provides us with a richer understanding of the ongoing activity, with a description of semantic relationships between the entities involved in the activity. Some example of activities are shown in Figure 1.1 which shows “cracking eggs into a pan using knife”, “pouring eggs and milk in a bowl”, “adding lettuce, ham and pickle to bagel” and “spooning flour”.

Figure 1.1: Example of a video description for the activity of a cooking video.

Picking up a bowl containing eggs.
Pouring eggs from bowl into a bowl.
1.2 Challenges and Motivation

The problem of automatically generating coherent semantic description of dynamic scenes from video data is hard for several reasons. Firstly, it is difficult to account for the large variety of potential interactions that can characterize activities. There can be variability in terms of the objects associated with an activity. For instance, different ingredients can be used in making a sandwich. Secondly, the object appearances vary due to pose and also interactions with other objects. For example, cucumber in a bowl appears different from when it is being handled by a cook. The appearance variability often leads to errors in feature extraction, segmentation, and tracking. Coping with detection errors is a key requirement in successful interpretation of activities from videos. Thirdly, not all the detected objects participate in a single activity. The non-participating objects can be considered clutter for the purpose of interpretation. For instance, many different food and utensil types can be present in the scene.

The challenge is to sort the relevant subset of detected objects from the clutter and associate them with each other to build relationships that compose a semantically coherent interpretation of the video scene. Lastly, multiple activities may be occurring simultaneously. The video might contain two cooks cooking separate recipes simultaneously, as it is typically seen a cooking competition shows.

Due to these challenges the performances of state-of-the-art algorithms for video interpretation are quite low, except in very controlled and narrow conditions. This has been highlighted in recent papers [11, 12, 1], showing low recognition performance rates for cooking video data sets. Even a small progress on trackling these challenges can impact a diverse range of application areas. Activity interpretation is useful for public safety using surveillance cameras [13], behav-
ior analysis and monitoring (e.g., of animals or elderly people) [14], human-computer interaction assistance [15], and semantic video content based indexing and retrieval [11, 1, 16].

1.3 Our Approach

The approach advocated here is combinatorial in nature and an overview of this framework is illustrated in Figure 5.1. We assume the existence of computer vision algorithms, labeled low-level feature processing and learning, that can detect and label basic items such as objects and actions in videos. This is not an unreasonable assumption given the great strides that have been made in object detection and action recognition, especially those using machine learning approaches. However, we allow for errors in both detection and recognition algorithms. In other words, not all the relevant objects may be detected perfectly and some extraneous objects may be present. Also, the labels assigned to objects and actions may not be perfect. Indeed, this is the case especially for the cooking datasets used for experiments in this work. Classification errors remain a common problem due to low discriminatory power of the selected features and the presence of overlapping and occluding objects in the scene, e.g. a bowl will usually contain either a spatula or a whisk, and will be harder to detect than when sitting alone. Even machine learning algorithms, that are considered state of the art in object and action classification, are highly susceptible to errors when encountering test cases that differ from training cases in a significant way.

Given these labels, our approach is to link them into interpretation structures using the domain knowledge. We formally express video interpretations using the language of Grenander’s general pattern theory [17], as shown in the last column of Figure 5.1. We briefly introduce some of the terminology here but will discuss it in more detail later using Figure 3.1. The basic units, which are termed generators, represents basic object and action categories. Each generator has
multiple out bonds, capturing different type of domain semantics, spatial constraints, and image support evidence. The generators combine, based on small set of pre-defined combination rules capture domain semantics, to form larger structures that represent the video interpretation. For instance, the generator for action pick up has out bonds to generators representing food, utensil, and grasper. A spoon can be a grasper for pick up but a stirrer for the action stir, changing the role of the object spoon. The resulting configurations can vary in size and structures. The representational power of this setup comes from this flexibility of the possible generator configurations. We will impose a probabilistic measures on these configuration spaces and will generate statistical inferences (most likely interpretations) using a Markov chain Monte Carlo based simulated annealing process.

The primary advantage of this approach is that it can handle the problems associated with appearance variabilities that show up as errors in object labels, object clutter, and simultaneous events, without the need for exponentially-large labeled-video training data. The requirement for training data is used here only for: (1) learning the basic object and action classifiers and (2) developing domain knowledge in form of co-occurrences of semantic labels. The first version of this framework appears in [1], which is followed by a demonstration of its potential use for description of long-hour videos using temporal semantic information in [18]. In [19], we demonstrate how the proposed framework, with the introduction of the concept of spatial coherence, handles inference both when the scene exhibits clutter of objects and when multiple events are occurring at the same time. Lastly, in [20], we show the success of the proposed framework in generating video descriptions using evidence from multiple modalities and single-label models at the same time, without particularly consider an early fusion or late fusion approach. The proposed approach is a generalization of the approaches considered by the literature that use either discriminative or
generative models, capable of handling multiple computer vision challenges associated with the problem of video description in a unified formalism.

1.4 Contributions

The main contributions brought by this research work are the following, to the computer vision scientific community on what concerns video activity analysis:

1. Scalable model for representing the structural variability of video events that can occur in a particular context without the need to specify each one of them. Structural diversity in this context means multiplicity of semantic relationships that can be encountered in a particular event.

2. A video description framework that requires linearly increasing amount of video training data for an exponentially large number of possible video descriptions;

3. A principled mechanism to integrate responses from different computer vision techniques and correlate video data from different modalities in a seamless and unified way for generating improved descriptions of video events.

1.5 Dissertation Overview

In Chapter 2, we present and discuss the limitation of the state-of-the-art and other interesting approaches proposed for addressing the problem of video description to motivate the proposed video description framework. In Chapter 3, we discuss the construction of the pattern theory-based video description framework. We present the concepts of pattern theory and concurrently show how we modeled the representation of human activities using them. In Chapter 4,
we discuss the generation of description using the proposed framework. This entails a discussion on the inference algorithm and the learning needed to make the framework operational. In the following Chapters, from 5 to 7, we discuss three different study cases demonstrating how the framework allow us overcoming the limitations of the approaches encountered in the literature for video description, which were discussed in Chapter 2. We conclude in Chapter 8 with a discussion about potential future work that could result in continued progress in the area of video description.
CHAPTER 2: PRIOR WORK ON VIDEO DESCRIPTION

Video description is a labeled summarization of the ongoing events in a scene. We explicitly represent this summarization as a connected structure, formally known as configuration. Although we explicitly define it as a textual description, here we will also present it as a graph-like structure, formally known as configuration. The problem of video description is typically modeled using one of two main general ideas. The first one ignores the fined-grained semantic relationships that define the activities and relies on learning discriminative models to find a posterior function that can differentiate statistical representations of different labels of activities. These methods rely on implicit models. The second school of thought believes in explicit modeling of the semantic relationships inherent to the description of the activity. For example, explicitly modeling the dependencies related to the occurrence of an action or object. Graphical models are the traditional tool for this line of work (explicit models).

The main difference between these methods is in the way contextual, logical and temporal dependencies are encoded. Table 2.1 [21] pinpoints the variations among works in both major categories of video interpretation approaches, i.e. explicit and implicit models. The difference is in the abstraction model used to represent the semantic structure of events and in the chosen

---

learning and inference methods. Here we only study cases of video description characterized by human activities; therefore, the literature will also include works on recognition of human activities as video classification tasks. For a comprehensive survey on recognition and description of human activities in videos, we refer the reader to [22]. Below we provide just an overview of the most recent methods in this area.

Video description has mostly been studied as a problem of classification, as can be seen in the papers [23, 24, 25, 26, 27, 7, 28]. In view of their simplicity and computational efficiency, discriminative methods have become popular, despite their limited goal of detecting only prescribed events [29]. Table 2.1 summarizes the most recent, to the date this manuscript is being written, approaches on the problem of video description. The simplest approaches, based on implicit modeling of semantic relationship, concentrate on selecting the right combination of features and machine learning algorithms to train classification model for each activity class. Histograms of optical flow and gradient orientations are among the most commonly used features, typically extracted from local neighborhood of space-time interest points. Bag of Visual Words (BoVW) appear to be a widely common statistical representations based on these features to summarize structural information. In this case, feature detectors are assumed to be noise-free and the learned classification models are assumed to be sufficiently discriminative using the chosen types of features. They rely on the robustness of learning algorithms to handle low-level feature errors. Additionally, no useful information about the structures of interest are explicitly encoded in the models – it is assumed that the chosen feature representation implicitly captures useful structural information descriptive of the target collection of semantic labels.
Note that these classification models do not account well for structural variability inherent to real semantic structures. To cope with a potentially large suites of activities, many approaches have taken an analysis-by-synthesis or generative view, and proposed compositional models that allow for flexible interpretations. These methods explicitly model interactions using different ideas: graphical models [30, 12], probabilistic description frameworks [31, 32], Markov logic networks [33], inductive logic programming [34], Petri-nets [35, 36], context free grammars [37], and AND-OR graphs [38, 39, 40, 41, 42]. They handle uncertainty by considering probabilistic measures on the representation space.

2.1 Implicit Models

Implicit structural models [56] [57] [58] attempt to capture these information implicitly through some general form of data representation [47], such as the BoVW framework [26] and Linear Dynamical Systems (LDS) [25], or a set of coefficients learned using the max-margin framework [50]. Their typical data representation scheme is based on bag-of-visual-words and the classification models are learned using traditional machine learning algorithms such as Support Vector Machines (SVM) [43, 44, 45]. These approaches attempt to bridge the gap between feature representation and video classification in one step. Others compute distributions or some statistical summaries, such as the co-occurrence of concepts, which serve to indicate or corroborate with certain probability the existence of more complex ones and also used to derive semantic description through text [46, 48, 11]. These approaches do not offer flexibility in representation because if new concepts are later added to redefine the characterization of a single complex event, the models have to be reconstructed. Moreover, it is not clear how scalable these methods are for when the structure variability and semantic complexity of the target events increase. In summary, for these
### Table 2.1: Summary of related work on interpretation of activities in videos.

<table>
<thead>
<tr>
<th>Work</th>
<th>Features</th>
<th>Representation</th>
<th>Learning</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motwani et al. 2012 [43]</td>
<td>STIP, HOG</td>
<td>BoVW</td>
<td>Obj-Act Co-occurrence prob. distr.</td>
<td>DPM, bagged REP decision, Bayes’ Rules</td>
</tr>
<tr>
<td>Guo et al. 2012 [44]</td>
<td>OpponentSIFT, STIP</td>
<td>BoVW-SVM</td>
<td>Concept score combination + SVM</td>
<td>SVM</td>
</tr>
<tr>
<td>Kantorov et al. 2014 [45]</td>
<td>MPEG-Flow</td>
<td>BoVW-SVM</td>
<td>SVM</td>
<td>SVM</td>
</tr>
<tr>
<td>Khan et al. 2011 [46]</td>
<td>Haar, Color, EOH, COH</td>
<td>High-Level Features</td>
<td>Statistical language modeling</td>
<td>Probabilistic parser</td>
</tr>
<tr>
<td>Krishnamoorthy et al. 2013 [48]</td>
<td>HOG, STIP</td>
<td>DPM, SVM</td>
<td>DPM, SVM, SVO LM</td>
<td>Linear interpolation of scores</td>
</tr>
<tr>
<td>Das et al. 2013 [11]</td>
<td>HOG3D, HOG, Color</td>
<td>MMLDA, DPM</td>
<td>MMLDA, DPM, Tripartite Template Graph</td>
<td>POS w/ NLP tools, MMLDA + Ranking</td>
</tr>
<tr>
<td>Laxton et al. 2007 [49]</td>
<td>not mentioned</td>
<td>ADBN</td>
<td>action detectors</td>
<td>approx. Viterbi</td>
</tr>
<tr>
<td>Tang et al. 2012 [50]</td>
<td>STIP-BoVW</td>
<td>variable duration HMM</td>
<td>LS-SVM</td>
<td>MAP w/ Dynamic Programming</td>
</tr>
<tr>
<td>Amer et al. 2012 [51]</td>
<td>STIP-BoVW</td>
<td>SPN</td>
<td>EM</td>
<td>MPE w/ graph parsing</td>
</tr>
<tr>
<td>Swears et al. 2014 [52]</td>
<td>not mentioned</td>
<td>GC-DBN</td>
<td>Adaboost + EM</td>
<td>Maximum Likelihood</td>
</tr>
<tr>
<td>Pirsaiavash et al. 2014 [53]</td>
<td>not mentioned</td>
<td>SRG (CFG)</td>
<td>LS-SVM</td>
<td>FSM [Viterbi]</td>
</tr>
<tr>
<td>Vo et al. 2014 [54]</td>
<td>not mentioned</td>
<td>SCFG+BN</td>
<td>manual + CFG compilation</td>
<td>Message Passing</td>
</tr>
<tr>
<td>Hilde et al. 2014 [12]</td>
<td>STIP, Trajectons-BoVW (actions as HMMs)</td>
<td>SCFG</td>
<td>manual</td>
<td>Graph Parsing w/ HTK</td>
</tr>
<tr>
<td>Si et al. 2013 [40]</td>
<td>not mentioned</td>
<td>AOG</td>
<td>IP*, MDLP*</td>
<td>Earley’s-like parser</td>
</tr>
<tr>
<td>Tu et al. 2014 [35]</td>
<td>not mentioned</td>
<td>AOG</td>
<td>manual</td>
<td>Hierarchical Cluster Sampling + Earley parser</td>
</tr>
<tr>
<td>Wei et al. 2013 [41]</td>
<td>RGBD HOG, Kinect®s 3D joint motion vectors</td>
<td>4DHOI (hierarchical graph)</td>
<td>manual</td>
<td>DN Beam Search</td>
</tr>
<tr>
<td>de Souza et al. 2014 [1]</td>
<td>HOG, HOF</td>
<td>Concept co-occurrence tables</td>
<td>MCMC-SA</td>
<td></td>
</tr>
</tbody>
</table>

*Information Projection  *Minimum Description Length Principle
models, the video interpretation task may consist of either video labeling or video description by natural language sentences. These approaches are simple to implement but they suffer limitations in handling low-level processing errors and scalability in structural representation.

2.2 Explicit Models

Probabilistic graphical models (PGMs) are the most common of all. Solutions based on PGMs encode context, logical and temporal dependencies explicitly using graph representations and knowledge about the domain to define the semantic relationships. Such models are in general parametric, such as Hidden Markov Models (HMMs) and dynamic Bayesian Networks (DBNs), that require full estimation of many parameters in high dimensional spaces. We classify these approaches as explicit models because they model the structure of events explicitly. With explicit models, video description is also commonly addressed as video labeling by maximizing some posterior probability or equivalently minimizing an energy function for input video features, e.g. dynamic Bayesian Networks (DBN) [49][52], Sum Product Networks (SPN) [51, 59], Stochastic Context-free Grammars (SCFG) [60][54][12][53], AND-OR graphs [41][42], Petri Nets [36], or general hierarchical graphical models [30] [61]. In some of these works, human activities are sought to be composed by a set of temporally ordered sequence of sub-events [54], which can also suffer certain order variations or have optional steps [12]. These works typically consist of a low-level layer in which feature observations provide evidence for concepts from the top layers, such as sub-events or composite events. For example, Hilde et al. [12] use HMMs to learn models for sub-events such as pour coffee and take cup and model more semantically complex events such as preparing coffee using a SCFG that describes their syntax in the form of occurrence of sub-events. Other works use the SVM framework to provide confidence values as evidence for the occurrence
of sub-events. Context and temporal dependence constraints are mostly supplied by coefficients learned with the max-margin framework or co-occurrence statistics of the target concepts.

2.3 Limitations

The major limitations of the approaches described above are the following:

1. Low scalability in representing the structural variability of different activities. For example, if all training instances are “crack egg”, then it will not be possible to compose a description involving other elements such as “crack egg in a bowl” or “crack egg in a bowl using a spoon”. For implicit models, the other objects, i.e. bowl and spoon, will be noise and an approach based on PGM would need to train a new model with more random variables.

2. If it was possible to enumerate all the many categories of activities, we would need an extraordinarily large amount of training data to account for all variabilities. The benefit of using probabilistic graphical models is that they provide compact representations and have the ability to handle uncertainty from the visual detectors; however, they may require large amounts of training data and parameterization may be unfeasible for more complex structures.

3. Clutter of objects may serve as noise for implicit models during the inference, which can lead to erroneous predictions. Similarly, explicit models do not account for extra occurrence of entities in their model. Most research work overlooks this issue by concentrating their experiments on controlled scenarios that conform with their training data sets.

4. Implicit models do not account for temporal information and explicit models have to account for order of occurrence when they do.
5. They may have to rely on additional methods, early fusion or late fusion, for prediction dependent on multiple modality data.
CHAPTER 3 : HUMAN ACTIVITY REPRESENTATION WITH PATTERN THEORY

In this chapter, we discuss how to use the language of pattern theory for modeling representation of human activities seen in videos. As we dissect the theoretical elements of representation in pattern theory, we also show their abstract concepts applied in a more concrete form, using actual examples.

3.1 Constructing a Pattern Theory Framework

We will construct a pattern theory-based framework to represent and describe human activities from videos in a precise mathematical form. Our focus is on cooking activities, such as “pour egg into plate”, “pour milk”, “crack egg”, “cut fruit”, etc. A sequence of such activities will typically describe a cooking recipe. The recipe steps, referred to here as activities, are interactions of actors and objects in fixed-length temporal intervals. Actors perform actions on objects that in turn can interact with other objects directly during the development of an activity. For example, in Figure 3.2 a woman is stirring eggs in a bowl using a spoon. In this example, we have an activity described in terms of a relation between a single action and multiple objects (e.g., stirring eggs, stirring using spoon) and objects with other objects (e.g., eggs in a bowl, spoon and eggs).

Portions of this chapter were previously published in Fillipe D M de Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. Pattern theory-based interpretation of activities. In ICPR, 2014.
Permission is included in Appendix A.
In this section, we will introduce some elements of pattern theory, starting with the generators and their bonds. We show how generators represent semantic labels of objects and actions and we show that bonds determine how generators can be used to describe activities. We define an equivalence relation (on the set of generators) to represent inter-changeability among labels. The generators are connected together by bonds to form configurations whose connections are governed by local and global regularity constraints. Configurations will be used to denote semantic interpretations of videos. Local and global regularity constraints control the cost in searching for a solutions in an exponentially large space of interpretations. Then, we show how bonds can be quantified and define a probabilistic superstructure over the configuration space. While this framework has some resemblance to other graphical models, we will discuss important differences in Section 4.7.

3.1.1 Generators and Bonds

The most basic unit of representation in pattern theory is a generator, denoted by $g$. Generators are the building blocks of more complex structures. They represent basic units of information specific to a certain domain knowledge. For example, in the cooking domain these units are actions, such as "stir", "fry", "pour", and objects, such as "egg", "plate", etc. The corresponding pattern theoretic names are stir generator, pour generator, egg generator, plate generator, etc. We will also use generators to represent features extracted from the video. These are called feature generators. In this work, feature generators will correspond to two types of features: histograms of optic flow (HOF) and histogram of oriented gradients (HOG). The complete collection of generators forms a generator space $G$. See an example of a generator space in Figure 3.1.
Figure 3.1: Example of generator space for cooking domain arranged into levels. The generators represent actions, objects and features types. Each generator has a bond structure formed by both out-bonds (white connectors) and in-bonds (black connectors). Each bond is associated with a categorical value. For example, the stir generator has bonds with bond values container, food, stirrer, and feature.
Each generator \( g \) has a set of connectors called bonds; the number of bonds \( w(g) \) is known as the \textit{arity} of a generator. Thus, each bond in a generator can be identified by a coordinate \( j = 1, 2, \ldots, w(g) \). Most importantly, it also has a \textit{bond value} \( \beta^j \) that takes values from a set \( \mathcal{B} \). The term \( \beta^j(g) \) will denote the \( j \)-th bond of generator \( g \). Bonds also have markers to indicate whether it receives connections, called \textit{in-bonds}, \( \beta_{\text{in}} \) or connects to other bonds, called \textit{out-bonds}, \( \beta_{\text{out}} \). In Figure 3.1, the in- and out-bonds are shown as links with black and white semicircles, respectively. For example, the \textit{bowl generator} has three bonds: one in-bond with bond value \textit{receptacle} and two out-bonds with bond values \textit{food} and \textit{feature}. The relationship among these bond values serves to determine when two bonds, each from a different generator, are compatible and can form a connection. The local connecting structure of a generator is made up of a bond structure \( B(g) \), with coordinates, markers and bond values.

The generator space \( G \) can be partitioned into disjoint sets \( G^\alpha \) induced by an \textit{equivalence relation} \( S \), such that \( G = \cup_{\alpha} G^\alpha \). An example of such partition is shown in Table 3.2 (right) for the cooking-domain generator space. Generators belonging to the same subset \( G^\alpha \) have the same bond structures, making them interchangeable. For example, the generators from the subset \( G^1 \) as presented in Table 3.2 (right), namely \textit{pick up}, \textit{put down}, have the same bond structure

\[
B(g) = \{\beta_{\text{out}1} = \text{feature}, \beta_{\text{out}2} = \text{food}, \beta_{\text{out}3} = \text{grasper}, \beta_{\text{out}4} = \text{utensil}\},
\]

as illustrated in Table 3.2 and Figure 3.1. Formally, this partition can be seen as the result of an equivalence relation on \( G \) induced by similarities \( s: G \to G, \forall s \in S \), such that

\[
g_i \equiv g_j \pmod{S} \iff \exists s \ni sg_i = g_j \tag{3.1}
\]
We name these subsets, which form the bond values set $B = \{\text{utensil, container, stirrer, food, feature}\}$. This is by no means the only way to define the bond value set but in our context it helps in formalizing the rules of combinations of generators. Note that \textit{container} is a category accounting for all object generators that can be used as a container and the \textit{utensil} category accounting for all kitchen tools. As an example, the \textit{pour} generator can combine with the \textit{bowl} generator by connecting its out-bond \textit{container} to the in-bond \textit{container} of the \textit{bowl} generator. A more interesting case involves the \textit{pick up} generator that, through its out-bond \textit{utensil}, can connect to any object generator that has in-bonds carrying multiple bonds values, namely \textit{utensil, container, stirrer} and \textit{grasper}. This idea will be formalized in the next section.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Level</th>
<th>$\beta_{in}$</th>
<th>$\beta_{out1}$</th>
<th>$\beta_{out2}$</th>
<th>$\beta_{out3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>pick up</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>grasper</td>
</tr>
<tr>
<td>2</td>
<td>put down</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>grasper</td>
</tr>
<tr>
<td>3</td>
<td>stir</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>stirrer</td>
</tr>
<tr>
<td>4</td>
<td>pour</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>container</td>
</tr>
<tr>
<td>5</td>
<td>crack</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>peel</td>
<td>3</td>
<td>-</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>whisk</td>
<td>3</td>
<td>stirrer</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>spatula</td>
<td>3</td>
<td>stirrer</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>9</td>
<td>spoon</td>
<td>3</td>
<td>stirrer</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>egg</td>
<td>2</td>
<td>food</td>
<td>feature</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>lettuce</td>
<td>2</td>
<td>food</td>
<td>feature</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>carrot</td>
<td>2</td>
<td>food</td>
<td>feature</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>knife</td>
<td>2</td>
<td>utensil</td>
<td>feature</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>plate</td>
<td>2</td>
<td>container</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>bowl</td>
<td>2</td>
<td>container</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>16</td>
<td>cup</td>
<td>2</td>
<td>container</td>
<td>feature</td>
<td>food</td>
<td>-</td>
</tr>
<tr>
<td>17</td>
<td>HOF</td>
<td>1</td>
<td>feature</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>18</td>
<td>HOG</td>
<td>1</td>
<td>feature</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3.2: The equivalence classes \( G^\alpha \).

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( G^\alpha )</th>
<th>( #(G^\alpha) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{pick up, put down}</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>{pour}</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>{stir}</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>{crack, peel}</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>{egg, lettuce, carrot}</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>{knife}</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>{whisk, spatula, spoon}</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>{plate, bowl, cup, pan}</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>{HOF, HOG}</td>
<td>2</td>
</tr>
</tbody>
</table>

3.1.2 Connectors and Configurations

Bonds allow generators to combine with each other to form connected structures termed configurations. These configurations denote interpretations of videos in our context. A configuration has an underlying graph topology, specified by the connector graph, \( \sigma \). The set of all feasible connector graphs \( \sigma \) is formally denoted by \( \Sigma \), also known as the connection type. We let the connection type follow the directed connections between elements of a Partially Ordered Set, i.e. \( \Sigma = \text{POSET} \). The POSET structure mirrors the hierarchical organization of the generator space \( G \) by levels \( \ell \), as depicted in Table 3.2 and Figure 3.1. Action generators are one level above the object generators, implying a natural order of connection, i.e. action generators have out-bonds that connect to in-bonds of object generators. In general, if \( g_i \) connects some out-bond \( \beta'(g_i) \) to an in-bond \( \beta''(g_j) \) of another generator \( g_j \), then \( \ell(g_i) \geq \ell(g_j) \).

Formally, a configuration \( c \) is a connector graph \( \sigma \in \Sigma \) whose sites 1, 2, \ldots, \( n \) are populated by a collection of generators \( g_1, g_2, \ldots, g_n \), thus, denoted by \( c = \sigma(g_1, g_2, \ldots, g_n) \). The collection of generators \( g_1, g_2, \ldots, g_n \) is the content of the configuration \( c \). There is a multitude of connectors \( \sigma \in \Sigma \) representing different structural patterns of configurations. These connectors vary in the
Figure 3.2: Example of a video interpretation using elements of pattern theory. The generators represent detected objects and actions. The bonds between generators express the relations and interactions found between the detected generators. Two main types of bond interactions are shown here, namely, support bonds (between feature generators and action/object generator) and semantic bonds (between action/object generators).

number of sites and in the arrangement of connections. Each site $i$ in $\sigma$ will host generators from a single or multiple subsets $G^\alpha$ so long as the structure of the connector graph $\sigma$ is preserved.

A single connector graph $\sigma$ and the generator space $G$ together span a space of feasible configurations $C(\sigma)$. The structure of the configurations $c \in C(\sigma)$ is the same, varying only in content, that is in the assignment of generators to the sites $i$. See Figure 3.3 for an illustration. The configuration space $C(\sigma)$ can be compared to a Markov random field (MRF) or a Bayesian network (BN). The graph structure $\sigma$ describing the relationship of the random variables remain the same; thus, an assignment to the random variables is analogous to a configuration $c$ with a particular collection of generators $\sigma(g_1, g_2, \ldots, g_n)$. To obtain different assignments to the sites of a connector graph $\sigma$, we apply the swapping transformation to change the generators in a configuration. These transformations exchange generators $g_i \in c$ with generators $g_j \in G^\alpha$ such that
Figure 3.3: Connector graphs and their configurations. Left: Examples of connector graphs. Center and right: Possible configurations using the connector graphs on the left, hosting independent collections of generators on their sites.

$g_i, g_j \in G^n$. This exchange of generators translates into a change of content of the configuration $c$, which results in a configuration $c'$. This can be considered a move in the configuration space $C(\sigma)$.

### 3.1.3 Regularity of Configurations

We have mentioned that generators connect to other generators through bonds that are compatible. We formalize this idea by introducing the bond relation function

$$\rho : B \times B \to \{TRUE, FALSE\}. \quad (3.2)$$

This function determines whether two bonds $\beta'(g_i)$ and $\beta''(g_j)$ between two generators, $g_i$ and $g_j$, are compatible and is denoted by $\rho[\beta'(g_i), \beta''(g_j)]$ or simply as $\rho[\beta', \beta'']$. The type of bond relation $\rho$ will vary according to the application; it could be of type ‘EQUAL’, ‘UNEQUAL’, ‘INCLUSION’, ‘EQUAL’.
Figure 3.4: Illustration of how transformation change configurations. (a) Left: A configuration $c$ whose interpretation is “picking up bowl and stirring eggs in the bowl using a spoon”. Center: A configuration $c' = T_s c$ resulted from applying the transformation $T_s$ on a subset of generators that forms $c$; its interpretation is “putting down cup and stirring carrot in the cup using a whisk”. Right: A configuration $c'' = T_s c$ resulted from applying the transformation $T_s$ on $c$; its interpretation is “putting down pan and stirring lettuce in the pan using a spatula”. (b) Interpretations after applying the transformation $T_c$ that results in removing one generator from the configurations shown in a). (c) Interpretations after applying the transformation $T_c$ that results in adding feature generators to the configurations shown in b).
and so on. For all the study cases presented here, we choose $\rho =$ INCLUSION. One can think of
the bond values as categories of things. Some categories are more specific cases of more generic
categories. For example, stirrer, grasper, and containers are specific cases of the category utensil.
Thus, we establish that an out-bond $\beta'_\text{out}$ carrying bond value utensil connects to in-bonds $\beta''_\text{in}$
carrying either of the bond values stirrer, grasper, or containers. See Table 3.3 for an example of
the bond relations used in this work.

Table 3.3: The truth-valued table by $\rho(\beta'_\text{out},\beta''_\text{in})$.

<table>
<thead>
<tr>
<th>$\beta'_\text{out}$</th>
<th>utensil</th>
<th>container</th>
<th>stirrer</th>
<th>grasper</th>
<th>food</th>
<th>feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>utensil</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>container</td>
<td>FALSE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>stirrer</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>grasper</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>food</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>feature</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

With these concepts and definitions, we may now formalize the notion of regular configu-

rations. A configuration $c$ is called locally regular if

$$\forall (\beta'_i, \beta''_j) \in c \quad \rho[\beta'_i(g_i), \beta''_j(g_j)] = \text{TRUE}. \quad (3.3)$$

Equation 3.3 is known as the first structure formula. A configuration $c$ is said to be globally regular
if $\sigma \in \Sigma$. A configuration is then called regular if it is both locally and globally regular. The
set of all regular configurations is denoted by $C(R)$. This formal notion helps us design inference
algorithms that search the regular configuration space $C(R)$ in an efficient and smart fashion. Note
also that $C(R)$ represents the union of all subspaces $C(\sigma)$, $\forall \sigma \in \Sigma$. 

26
This completes the formal specification of a *regularity*, denoted by \( R(G,S,\rho,\Sigma) \). The regularity \( R \) specifies in mathematically precise terms the principles that govern the construction of regular structures to represent patterns of interest. Additionally, the regularity help define efficient operations on the regular configuration space \( C(R) \) that does not violate the domain-specific characteristics of the patterns it is meant to formalize. This will be beneficial in the design of efficient algorithms to perform probabilistic analysis of these structures.
CHAPTER 4: GENERATING SEMANTIC DESCRIPTIONS FOR VIDEOS

In previous chapter, we introduced the basic units of representation in pattern theory. In this chapter, we devote attention to leveraging this representation scheme to perform inference on video data and generate descriptions about human activities appearing in it. We know that a set of generators span a space of allowable configurations by connecting with each other through their local structure (bonds). Configurations are the result of performing inference and each encodes a description of the ongoing activities in a video. The goal is to devise an algorithmic process that takes as input video features and the representation scheme discussed earlier to build configurations that explain the activities happening in a video.

The proposed inference process relies on a set of operations on the configuration space that we formalize in Section 4.1. These operations ensure that the bond relations are preserved as new configurations are inferred and proposed as solutions. Once a solution, i.e. a promising configuration, is found, we must be able to decide whether or not it provides a good description for the target video scene. The quality of a video description is based on a quantification of the energies of the bonds that form the inferred configuration. Thus, in Section 4.2, we explain how the energy of closed bonds are quantified, which is then followed by a discussion, in Section 4.3, on the probabilistic superstructure imposed on the configuration space. In Section 4.5 presents discuss and describe the algorithms designed for the inference process.

Portions of this chapter were previously published in Fillipe D M de Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. Spatially coherent interpretations of videos using pattern theory. International Journal of Computer Vision, 2016. Permission is included in Appendix A.
4.1 Operations on the Configuration Space $C(R)$

One can operate on and modify regular configurations in several ways. We use two types of transformations for this purpose: (1) swapping of generators and (2) changing of configuration structure. The former applies similarity transformations from $S$ to interchange objects and actions and the latter applies some simple transformations that change the structure of the configuration. We will denote them as $T_s : G \rightarrow G$ and $T_c : C(R) \rightarrow C(R)$, respectively. These transformations allow us to traverse the space of regular configurations $C(R)$ in a structured manner.

The generator swapping transformation $T_s$ ensures regularity by design. Given a configuration $c \in C(\sigma)$ and equivalence classes $G^a \subset G$, $c$ can be changed by applying $T_s$ to any nonempty subset of generators $g_i \in c$ (see Figure 3.4 (a)). Such transformations consist of replacing a generator $g_i$ by another generator $g_j$ such that $g_i, g_j \in G^a$. Thus the bond structure $B(g_i)$ is preserved, i.e. $T_s$ replaces each target $g_i \in \sigma(g_1, g_2, \ldots, g_n)$ with some $g_j$ in the same equivalence class $G^a$, where $B(g_i) = B(sg_j)$. As a result, the connector graph $\sigma$ remains unchanged and the operation results in a configuration $c'$ from the local neighborhood of $c$ (i.e., $c' \in C(\sigma)$). Figure 3.4 (a) illustrates this idea: Columns 2 and 3 show two possible different results from applying $T_s$ to $c$ shown in the left.

We use $T_s$ to change just one generator $g_i$ of a configuration $c$ but a sequence of these transformations could result in changing multiple generators. Transformations that change the structure of the configurations allow us to reach configurations from a different $C(\sigma) \subset C(R)$. This means that the operation changes the connector $\sigma$; thus, resulting in a configuration $c'$ from a neighboring subspace $C(\sigma')$. Such operations are performed on the connector graph $\sigma$, such as i) removing a site $i$ from $\sigma$ to produce $\sigma'$ or ii) adding a new site $j$ to $\sigma$ to produce $\sigma''$. These transformations are defined as $T_c : C(R) \rightarrow C(R)$ and are illustrated in Figures 3.4 (b) and (c), respectively.
These transformations are the building blocks for designing inference proposal functions as described in Subsections 4.5.2 and 4.5.3. We use $T_c$ to assemble configurations with arbitrary connector shapes in the global proposal function. For the local proposal function, $T_s$ is used to propose new configurations that vary in content but preserve the structure whereas $T_c$ is used to replace existing generators by others that result in changes to the connectivity of the configuration.

### 4.2 Bond Quantification

We consider two main types of bond interactions. Semantic bond interactions are those between action generators and object generators or between object generators. These bonds measure the semantic interaction compatibility among actions and objects. Support bond interactions are those bond connections formed by feature generators and either action generators or object generators; these bond interactions provide video data support to the action and object generators forming the video interpretations. See Figure 3.2 for an illustration for these two different types of bond interactions.

Before we can derive a quantity to evaluation a configuration $c$, we must define how to quantify the bond interactions. The quality of a bond interaction $(\beta', \beta'')$ is measured by the acceptor function $a(\beta', \beta'')$, which is defined as

$$w(\beta', \beta'') \tanh(\gamma(\beta', \beta'') f(\beta', \beta'')).$$

The acceptor function $a$ expresses the degree of compatibility between a pair of connected generators. The function $f(\beta'(g_i), \beta''(g_j))$ outputs classification confidence score produced by ac-
tion/object classifiers when \(g_i\) is an action/object generator and \(g_j\) is a feature generator, where \(\beta'\) and \(\beta''\) are the bonds that connect \(g_i\) and \(g_j\). \(\gamma()\) is a weight function for the scores output by \(f\).

If the generators \(g_i\) and \(g_j\) are from the set of action and object generators, then \(f\) is the product of two functions \(g\) and \(h\). The function \(g\) takes values on a frequency table whose values inform how often actions and objects occur together. The frequency tables are constructed by counting the number of occurrence actions and objects appear together in the activities from the data set annotation. The function \(h\) outputs the overlap ratio between the spatial locations of the features that support the generators \(g_i\) and \(g_j\) that connected by the bond \(\beta'\) and \(\beta''\) (see Figure 6.2 for an illustration). This way, \(f(\beta'(g_i), \beta''(g_j)) = g(\beta'(g_i), \beta''(g_j))h(\phi(g_i), \phi(g_j))\), where \(\phi(g)\) represents the spatial location of the generator \(g\). This relationship \(g(\beta'(g_i), \beta''(g_j))h(\phi(g_i), \phi(g_j))\) encourages semantic bond interactions between action and object generators whose associated explaining features overlap in space and time. This function \(h(.)\) introduces the notion of spatial coherence to the interpretations. We define spatial coherence as the grouping of semantic elements (action and object categories) that are spatially correlated due to the location proximity of their associated features.

Because \(f\) outputs values from different sources (classification scores and frequency tables) that have different ranges, we weigh the \(f\) responses using the weight function \(\gamma(\beta', \beta'')\) (this is appropriate since the type of bond interaction is directly related to the output source of \(f\)). This allows us not to favor certain types of bonds over others because of the data source. The raw score value of a bond interaction produced by the product \(\gamma(\beta', \beta'')f(\beta', \beta'')\) is rescaled using the hyperbolic function \(\tanh\) to fall into the range -1 to 1. In practice, we let \(\gamma(\beta', \beta'') = 2\) when the score value produced by \(f\) is a classification confidence score and let \(\gamma(\beta', \beta'') = 0.015\) if it comes
Figure 4.1: Illustration of the notion of spatial coherence. Regions A-F are examples of area overlaps between object bounding boxes or an object bounding box and the motion-salient area where an action occurs. 

a) An example video showing the interaction “picking up spreader”. Just a few bounding boxes appear to show detected objects in the scene. 

b) Only the bounding box for the spreader object is completely overlapping with the area where the action occurs. 

c) An example video showing several detected objects are detected in a scenario where only one of them is participating in the interaction. 

d) Region C provides evidence through overlap ratio-based spatial constraint that the bread object is the only object participating in the predominant activity.
from the frequency table. These parameter values are defined so that the minimum, median and maximum values defining the ranges of the different data sources (the action/object classifiers and the frequency table) are mapped to the same values when input to the hyperbolic function \(\text{tanh}\). In addition to this, we may desire to weigh the rescaled compatibility scores of the bond interactions differently so as to guide the inference process in a specific direction. To this end, we introduce a variable \(w\) as a function of the bond interaction \(\beta', \beta''\) to the bond interaction quantification that results in the Equation 4.1. \(w(\beta', \beta'')\) can be either learned automatically or manually defined. In practice, we found that weighing the bond interactions equally produces good results.

4.3 Probabilistic Superstructure on \(C(R)\)

Now we introduce probability measures on configuration spaces. Given a set of video features, our goal is to find an interpretation that obey the regularity \(R(G, S, \rho, \Sigma)\) and best describe the ongoing events in a video. Such semantic patterns are regular configurations \(c \in C(R)\) whose regularity is verified with the first structure formula described in Equation 3.3. This regularity measure simply states whether the configuration is regular or not. A probability superstructure on the regular configuration space will measure the probability, or degree of regularity, of the configurations, conditioned upon the input features (which appear as feature generators in the configurations). This notion is formalized by quantifying the first structure formula with a probability density function \(p\) on the configuration space \(C(R)\),

\[
p(c) = \frac{1}{Z} \prod_{(\beta', \beta'') \in \sigma} \exp(a(\beta'(g_i), \beta''(g_j))),
\]  \hspace{1cm} (4.2)
where \( Z \) is the normalizing constant or the partition function, and the acceptor function \( a \) expresses the degree of compatibility between a pair of connected generators, as defined in previous section. This probability density function \( p \) on the regular configuration space \( C(R) \) takes the form of a grand Gibbs ensemble since the connector graph \( \sigma \) is variable; thus, the partition function is given by

\[
Z = \sum_{\sigma \in \Sigma} \sum_{c \in C(\sigma)} \prod_{(\beta', \beta'') \in \sigma} \exp(a(\beta'(g_i), \beta''(g_j))).
\] (4.3)

Equation 4.2 is also termed the second structure formula. It measures the amount (degree) of regularity in a configuration \( c \), or in other terms, the probability of \( c \) for a particular set of input features. To perform inference, we must maximize the probability \( p \) or equivalently minimize the energy function \( E(c) = -\log p(c)Z \), which results in

\[
E(c) = -\sum_{(\beta', \beta'') \in \sigma} w(\beta', \beta'') \tanh(\gamma(\beta', \beta'')f(\beta', \beta'')).
\] (4.4)

4.4 Learning

The elements to be learned are the generator space, the action and object classifiers and the frequency tables quantifying the co-occurrence of actions and objects. The generator space can be learned in several ways. For example, it can be learned by parsing video annotations or documents about the domain knowledge (e.g., recipe books). Ideally, it may also be designed using a more formal structure of knowledge as an ontology. For simplicity and practical reasons, we learned the generator space by parsing the video annotation files from the selected data sets. In short, we do not provide a principled mechanism to learn the generator space. We also relied on the provided video annotations to learn classification models for the individual categories of actions.
and objects. There is no requirement regarding the number of variations in which these categories should appear to learn strong classification models. In fact, we assume that these classification models are mostly weak predictors. Two dimensional frequency tables were learned by counting the number of occurrences in which actions and object appear together in the video annotations; however, note that these parameter values may also be provided by a domain knowledge expert.

4.5 Inference

The inference process consists of finding the most probable interpretation of the contents of a video. This amounts to minimizing the energy cost function $E(c)$. The likeness to MRF’s energy function is superficial. Both use a pairwise energy model but the search spaces for the solutions are very different. Unlike other models such as MRF whose solution space is spanned using a single connector graph, with the pattern theory framework the inference algorithm works to simultaneously find the structure (i.e., the connector graph) as well as the “labels” of the nodes of that structure that together best explain the data. We are allowed to leave some features without explanation; thus, both the number of closed bonds and the number of generators are variable. Given the facts above, this problem cannot be solved exactly without taking an unfeasible computational amount of time. In fact, we may consider an infinite number of possible connector graphs that can be used to construct interpretations. This motivates the use of a sampling strategy for inference, following in spirit the Heuristic Search algorithm.

Our inference process is a Monte Carlo Markov Chain (MCMC) algorithm coupled with a simulated annealing scheduling. This algorithm comprises of two proposal functions designed using the family of transformations $T : C(R) \rightarrow C(R)$ described in previous section. The quality of the proposal functions is critical for the success of the search. These proposal functions are
generic inference engines that drive the search through the configuration space for the best solution. Each proposal has a role in the search. The global proposal function operates on $C(R)$ to propose configurations that may come from different subspaces $C(\sigma) \subset C(R)$. The local proposal function makes simple, local changes in the configurations by replacing existing generators. This substitution could result in new configurations whose structures are most of the time preserved (using transformation $T_s$). In those cases, the replacement candidates are drawn from the same $G^\alpha$ to which the generator $g_i$ to be removed belongs. Other times, we will let the replacement to come from different $G^\alpha$’s; thus, changing the structure of the configuration (using the transformation $T_c$). As follows, we present in detail all algorithms used for inference, including the MCMC-based simulated annealing procedure and all proposal functions.

### 4.5.1 MCMC-Simulated Annealing

The steps of the MCMC-simulated annealing algorithm are presented in Algorithm 1. First, in line 2, the algorithm creates a configuration $c''$ with feature generators connected to their top $k$ action/object generators given classification confidence scores. Each feature generator corresponds to a set of features extracted from either a spatiotemporal region or an object tracklet of the input test video. This results in a configuration containing feature generators connected to multiple action/object generators as depicted in Figure 4.3 a). This configuration is a start point for constructing new configurations. We use it as input to the global proposal function for sampling new configurations (or video interpretations). The algorithm initializes the inference process by calling the global proposal function. The global proposal function samples a new configuration $c$ using $c''$ as a reference (line 3). The newly proposed configuration is then kept as the best seen so far (line 4).
The sampling process starts at line 5 and finishes when the maximum number of iterations \( k_{max} \) is reached. The first step is to sample a number \( t \) between 0 and 1 according to a uniform distribution. Then, a new configuration is sampled using the global proposal function if \( t \) falls in the range 0 to \( p \), which means that the global proposal function is called with probability \( p \). As a result, the local proposal function is called with probability \( 1 - p \), which is invoked every time \( t \geq p \) (line 10). In line 11, the annealing temperature \( T \) is updated. The temperature \( T \) starts high and cools down as the sampling process approaches the end. \( T \) changes by a factor of \( \alpha^k \), where \( \alpha \) is a predefined constant that dictates how fast the temperature drops at each iteration. The factor of change is updated according to each iteration \( k \) so that the temperature drop is higher by a factor of \( \alpha \) each time.

Given a fixed parameter values for \( k_{max} \) and \( \alpha \), higher initial temperatures \( T_0 \) will allow the sampling process to spend more time in the exploratory phase and take longer to converge. Lower initial temperatures will spend less time in the exploratory phase but converge faster. Typically, we seek an initial temperature \( T_0 \) that lets it spend equal amount of time in both the exploratory and exploitation phases. In practice, we set the initial temperature \( T_0 \) to 2500 and the maximum number of iteration \( k_{max} \) was set to 12000 (also used as the main stop criteria). \( E(c) \) is the energy of a configuration \( c \) and \( \alpha \) was empirically chosen to be 0.9967. Every newly proposed configuration (lines 6-10) is accepted if it passes the test at line 12, which is either true for new configurations with energy lower than the current configuration’s or true with a certain probability that is proportional to the energy difference between the recent and the old configurations.
**Algorithm 1: The Inference Algorithm: MCMC Simulated Annealing**

| **Input:** | $G, \alpha, p, k_{\text{max}}$ |
| **Output:** | $K$ best configurations (descriptions) that explain the input video |

Let $\sigma(c’’)$ be a configuration where each feature generator $g_f \in \sigma(c’’)$ is explained by the top $k$ most likely ontological predictions.

$\sigma(c) \leftarrow \text{GLOBALPROPOSAL}(\sigma(c’’));$

$\sigma(b) \leftarrow \sigma(c);$  

for $k \leftarrow 1, k_{\text{max}}$ do

$\quad t \sim U(0,1);$  

$\quad \text{if } t < p \text{ then }$  

$\quad \quad \sigma(c’) \leftarrow \text{GLOBALPROPOSAL}(\sigma(c’’));$

$\quad \text{else }$  

$\quad \quad \sigma(c’) \leftarrow \text{LOCALPROPOSAL}(G, \sigma(c));$

$\quad T \leftarrow T_0 \times \alpha^k;$  

$\quad \text{if } P(E(\sigma(c’)), E(\sigma(c)), T) \text{ then }$  

$\quad \quad \sigma(c) \leftarrow \sigma(c’);$  

$\quad \quad \text{if } E(\sigma(c’)) < E(\sigma(b)) \text{ then }$  

$\quad \quad \quad \sigma(b) \leftarrow \sigma(c’);$  

$\text{Input: } P(e’, e, T)$

$\text{if } e’ < e \text{ then }$

$\quad \text{return } True;$  

$\text{else }$

$\quad z \sim U(0,1);$  

$\quad \text{if } z < \exp(-(e’ - e)/T) \text{ then }$

$\quad \quad \text{return } True;$  

$\quad \text{else }$

$\quad \quad \text{return } False;$

---

38
4.5.2 Global Proposal Function

The global proposal function serves to diversify the search process by replacing a current interpretation by a completely new one. The resultant interpretation will most likely have different structure and completely new collection of generators. The global proposal function receives a configuration $c''$ formed by feature generators connected to their $k$ strongest action/object generators given the classification prediction scores. In line 2, Algorithm 2, a feature generator $g_f$ is randomly selected from $c''$ according to a uniform distribution. Then, in line 3, one of the top $k$ action/object generators explaining the selected feature generator is randomly chosen, also according to a uniform distribution, to explain $g_f$ in the new interpretation $c'$. Steps 3 through 5 are repeated until no feature generator is left to be explained. Finally, the collection of selected action and object generators are bonded among themselves so that the number of closed out-bonds in the interpretation is maximized. An illustration depicting the steps of the global proposal function is shown in Figures 4.3 a)-c).

Algorithm 2: Global proposal function algorithm.

\begin{verbatim}
1: procedure Global Proposal ($c''$)
2: Create empty configuration $c'$;
3: Select a feature generator $g_f$ from $c''$ and add it to $c'$;
4: Select one of the action/object generator $g_k$ that explains $g_f$ in $c''$;
5: Connect $g_k$ to $g_f$ in $c'$;
6: Repeat until no feature generator $g_f \in c''$ is left to be explained in $c'$;
7: Connect all generators $g_k \in c'$ so as to maximize the number of closed out-bonds;
8: return $c'$.
\end{verbatim}

4.5.3 Local Proposal Function

We implemented a local proposal function whose steps can be summarized as follows. First, it selects an existing action or object generator $g_i$ from the current configuration $c$ to be re-
placed by another generator $g_k \in G$ that does not belong to $c$. Then, it proposes $m$ replacement candidates $g_k$ for $g_i$ such that $g_i, g_k \in G^\alpha$. Finally, it selects the candidate $g_k$ that forms a configuration $c'$ whose total energy is the minimum out of those formed by the other candidates. This is done by simply computing the energy in the local neighborhood of the generator $g_k$, which consists of summing up the energies of all its closed in-bonds and out-bonds. The local proposal function is described in Algorithm 3. It starts in line 2 by randomly selecting an action or object generator from the current interpretation. Then, from line 3 to line 7, $m$ replacement candidates $g_k$ are sampled either i) from the subset of generators $G^\alpha$ to which $g_i$ belongs or ii) from more general set of generators $G$; there is 50% of chance for each sampling option. In line 8, the generator $g_i$ is removed from $c$. The local energy contributed to the total energy of the interpretation is measured for each candidate $g_k$ individually. Such local energy is computed by summing the energies contributed by each closed in-bond $\beta'' \in B_{in}(g_k)$ and out-bond $\beta' \in B_{out}(g_k)$ of the candidate $g_k$ (line 9). The candidate that contributes with the lowest amount of energy to the total energy of the interpretation is selected as the best surrogate and used to form the new interpretation (line 10). All random selections follow a uniform distribution. An illustration depicting the steps of the local proposal function is shown in Figure 4.4 d-f).

4.6 Time and Space Complexities

The search space size is exponential with respect to the number of feature generators. However, the space of all feasible configurations, constrained by the bond structures, is not exponential. All possible connections of generators are not allowed for it is constrained by the bond relation function $\rho(.)$. Given an interpretation, adding a new feature makes the search space grows according to $O(n)$, where $n$ is the number of generators in the current interpretation.
Algorithm 3: Local proposal function that replaces some generator g from c that minimizes $E(c)$.

**Input:** A configuration $c$ and a positive integer $m$.  
**Output:** New configuration $c'$ derived from replacing some generator of $c$.

Randomly select $g_i \in c$, $\ell(g_i) > 1$

$z \leftarrow \text{random\_number}(0,1)$

if $z \leq 0.5$ then
  Form a set $G' \subset G^a$ of $m$ generators $g_k$, such that $g_i \in G^a$ and $g_k \neq g_i$
else
  Form a set $G' \subset G$ of $m < n - 1$ generators $g_k$, $\forall g_k \neq g_i$

Remove $g_i$ from $c$, forming $c'$

Select the generator $g_k \in G'$ that minimizes $E(\sigma(c', g_k))$

Add $g_k$ to $c'$ to form $c''$

return $c''$

Our global and local proposal functions explore the bond-constrained space of feasible solutions in an efficient manner. The time computational complexity was worked out to be $O(k \ast m_c \ast m_o + k(n_f \ast m_o))$, where $k$ is the total number of sampling iterations, $m_c$ is the number of bonds from a candidate generator for replacement, $m_o$ is the total number of open bonds in a current interpretation and $n_f$ is the number of feature generators.

### 4.7 Differences with Other Graphical Models

Competitive models such as Markov random fields (MRF), Conditional random fields (CRF), Bayesian networks (BN), Markov Logic networks (MLN) and And-Or graphs (AOG) rely on representations that have *fixed graph structures*. They typically cannot accommodate topological changes in the graphs based on the data. CRF, MRF, MLN and BN are defined over a finite number of random variables whose relationships are predefined, or learned, and the goal is to find the assignment that maximizes the posterior probability. All these models can be absorbed in our framework, by assuming that the connectivity pattern is fixed. For example, an MRF can be
modeled using a single type of connector graph $\sigma$ (see Section 6.4 in [62] for more details). This results in a probability distribution on the regular configuration space $C(\sigma)$, which is a subspace of the space of solutions spanned by our generator space. In pattern theory, instead of random variables, we have generators. In theory there is no limit in the number of generators that can be used to construct interpretative structures and their forms are as general as it can be. Variations in the form in which an event can occur are accounted for by the probabilistic nature of the connections (bonds) and allowance for arbitrary number of connections among compatible generators.
(a) Each observation (highlighted in bold red) from the video can be explained by $k$ best labels.

(b) Explaining each observation with one of the $k$ best labels associated with them (uniform sampling of the label for each observation).

Figure 4.2: First two steps of the global proposal function. (a) First, it constructs a configuration whose feature generators are connected to their top $k = 3$ action/object generators given the classification prediction scores; no bonds between action and object generators. (b) Secondly, it forms a configuration with each feature generator connected to a single action/object generator (all generators highlighted in bold red).
Figure 4.3: Final step of the global proposal function. It finds the bond relationships between generators that maximizes the number of closed bonds in the configuration (bonds highlighted in bold red). This also amount at finding a set of relationships between those labels of nouns and verbs that form a semantically coherent interpretation.

Figure 4.4: These figures illustrate how the local proposal function works. First (bottom left), an action/object generator in the current interpretation is selected for removal. Then (top left), \( m \) generator candidates \( g_k \) are sampled from some subset \( G^{\alpha} \), such that \( g_i, g_k \in G^{\alpha} \) (in this example, \( g_i \)). Finally (right), it selects the candidate that minimizes the energy of the configuration. An interpretation structure with the newly selected surrogate is proposed (configuration highlighted with a red bounding box).
CHAPTER 5: EXPERIMENTAL FRAMEWORK, FIRST RESULTS

In this section, we present and discuss the qualitative advantages of modeling video content analysis using pattern theory. We discuss how elements of the pattern theoretic approach allows us to overcome common issues in computer vision tasks, such as object and action classification errors, and why. We also demonstrate how the inference process of the proposed approach work in cases of multiple occurrence of events and multiple presence of objects appearing in the ongoing events. Unlike nonstructural approaches such as discriminant models, there is always a concern on whether structural models such as probabilistic graphical models can span a sufficient number of variations representative of a phenomena or not. We show that our pattern theory based approach overcome this challenging while maintaining two keep properties: Both the number of representation units and required amount of training video data grow linearly with the complexity of the phenomena. Note that the space of solutions for describing a phenomena of interest (that is, an event) still increases exponentially. However, the search is constrained by the semantic structure imposed by the ontology.

In this section, we describe the datasets and the computer vision challenges associated with them. We also discuss the computed features and the training procedure for learning ob-

---

4 Portions of this chapter were previously published in Fillipe D M de Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. Pattern theory-based interpretation of activities. In ICPR, 2014.
Permission is included in Appendix A.
j ect and action classification models. This is followed by a description of the baseline algorithms considered for comparative analysis and, subsequently, definition of the evaluation metrics. This section ends with a discussion on the setup for the scenarios with simultaneously occurring activities.

5.1 Data

Two video datasets of human activities in cooking scenarios have been recently published: YouCook dataset [11] and the Breakfast Actions dataset [12]. Their videos exhibit many computer vision challenges. They show variability in background, conditions of illumination, the number of objects in the scenes, different performing actors, camera motion, to name a few.
5.1.1 YouCook Dataset

For the experiments on the YouCook dataset, we use the original training set (44 videos) split evenly into new smaller sets, one for training and another for test. We chose to only work with the original training videos because of the availability of object bounding box annotations. The YouCook dataset video shots depict a variety of steps of cooking recipes of various cooking styles, such as baking, grilling, frying, etc. These shots form our units for interpretation. There are 309 training and 359 test video shots, respectively. Each shot exhibits one of the 4 studied actions and displays some of the 19 objects that might be participating in the action or just appearing in the scene. For example, a shot could depict a cook picking up a spoon to stir ingredients in a bowl while a slice of meat and a knife are on the table. The action categories are stir, pickup, putdown, and pour. The object categories are bowl, cup, spatula, knife, pan, tongs, plate, oil, pepper, tomato, butter, spreader, bread, spoon, lemon, carrot, meat, and egg; thus, there are more than 2400 combinations of these categories that form activity interpretations to describe the events in a video shot. Despite the large number of interactions, we only need to train classification models for each individual action and object, which consists of a much smaller set (23 categories).

5.1.2 Breakfast Actions Dataset

The Breakfast Actions dataset consists of more than 1000 recipe videos, comprising of diverse scenarios depicting a combination of 10 recipes, 52 subjects and up to 5 cameras (viewpoints). The units of interpretation are temporal video segments of these videos, given by the video annotation provided along with the dataset. These units are recipe steps, for instance, pour coffee, peel fruit and fry egg; thus, we consider more than 5000 units of interpretation for evaluation purpose. The possible activity interpretations are spanned by the combination of > 10 action
categories (e.g., peel, crack, squeeze, fry, butter, stir, smear) and > 25 object categories (e.g. plate, fruit, cereals, egg, orange, bun, knife, coffee, glass).

5.2 Training

The video annotations from the training sets were used to learn classification models for the studied action and object categories. The training video annotations were also used to learn the parameters of the frequency tables describing the co-occurrence of actions and objects. We trained classification models for objects and actions using linear support vector machines (SVM). To this end, we used the well-known LibSVM tool [63]. We noticed that training with unbalanced data resulted in biased classification models for the majority categories. This issue was attenuated using SMOTE. Synthetic samples were generated using SMOTE [64] to account for the imbalance in the number of instances across categories.

In addition to imbalance in the dataset, training instances of certain objects typically appear occluded by other objects, contributing to learning of weak classification models since the features extracted from these instances describe more than one class label. For instance, spatulas are commonly used to stir ingredients in a bowl; therefore, spatulas frequently appear in training instances of bowls. Because of that, the learned classification models are subject to confuse instances of one class for another. This is the kind of confusion that we expect to be alleviated using prior information about the domain.

5.3 Features

An action is represented by a sequence of three stages of motion pattern captured by histograms of optic flow (HOF). Dense optic flow frames are computed for pairs of consecutive frames
of a video segment. The compute sequence of dense optic flow frames is divided in three temporal smaller, consecutive sequences: The first depicts the start of the action, the second represents the development of the action and the third represents the end of the action. An HOF weighed by the magnitude is then assembled for each of these smaller temporal segments to characterize the motion patterns of the action’s start, development, and ending. The action is then represented by the ordered concatenation of the three computed HOFs. As for the objects, we use the histograms of oriented gradients (HOG) computed from bounding boxes of object candidates. Other more sophisticated features are possible; however, these suffice for now to demonstrate the power of using the pattern-theoretic framework.

### 5.4 Baseline Algorithms

For comparative analysis on the YouCook dataset, we implemented a variant of our algorithm that does not use prior information about the co-occurrence of objects and actions nor does it consider bond interactions between action and object generators. Because of the latter restriction, its interpretation does express exactly how the detected actions and objects interrelate with each other in the ongoing events. The former restriction indicates that finding an interpretation is built solely based on connecting action and object generators to feature generators given their classification prediction scores.

This variant follows the dominant machine learning-based paradigm widely used in computer vision tasks. That is, it only uses purely data-driven classification models to create interpretations. It forms interpretations whose bonds link both action and objects to feature generators. Features are the only evidence supporting the interpretation content. No relationships about what action is performed on which object or which objects interact with other objects are identified.
For the experiments on the Breakfast Actions dataset, we used the performance rates of algorithms reported in [12] as our baselines. This allowed us to directly compare our approach with other competitive models for activity analysis, for instance, hidden Markov models (HMM) and Context-Free Grammar (CFG).

5.5 Evaluation Metrics

An interpretation constitutes of a network of action, object and feature generators that are connected by their compatible bonds. Evaluating the quality of an interpretation consists of computing the recall and precision associated with its collection of generators.

1. Recall: is the number of correct generators divided by the total number of generators in the ground-truth interpretation, excluding feature generators.

2. Precision: is the number of correct generators divided by the total number of generators in the output interpretation, excluding feature generators.

Using the example depicted in Figure 6.5 c), the recall is the number of correct action and object generators (highlighted in red) divided by the total number of action and object generators in the ground-truth interpretation shown in Figure 6.5 d), which results in saying that the output interpretation is 75% correct. For this example, the precision is coincidentally of 75% as well.

5.6 Inference Performance

We evaluate the inference algorithm's performance by measuring the precision and recall values on the set of generators that form an interpretation. Figure 5.2 shows the form that the dynamics in searching for the optimal solution takes during the inference iterative process. The
beginning of the exploratory phase is mostly characterized by configurations with high energy and low recall and precision. As it transitions to a more aggressive exploratory phase, we notice more findings of interpretations with lower energy and better recall and precision. The algorithm ends its search converging to proposals with lower energy and higher recall and precision as expected.

We analyze the iterative procedure performance and verify that the algorithm behaves as expected. The iterative process is illustrated in Figure 5.3, showing how the interpretation structures change using the proposal functions described above. We also show that the average performance at each iteration across all test videos increases rapidly in the beginning (the exploratory phase). In the remainder of the iteration, interpretations are adjusted for slightly better or worse performance rates until convergence is achieved – which is characterized by a steady average performance rate.

5.7 Robustness to Classification Errors

The underlying inference engine of the proposed framework depends on two key elements: i) computer vision and machine learning classification models for multiple categories of objects and actions and ii) prior knowledge data in the form of actions and objects co-occurrence tables (these tables can be thought of as potentials). It is well known that classification errors remain a challenge in computer vision tasks. This issue can be caused by i) the choice of features, ii) the limitations of the learning algorithm, iii) dataset annotation errors, iv) noise in training examples (e.g., image examples of a bowl occluded by hands). Thus, it is instructive to understand how the performance of the proposed pattern theory framework is affected by these design factors. We analyze the robustness of the framework for two important scenarios of error: 1) increasing error rates of classifiers in labeling evidence (features), and, 2) given a classification error rate, we verify
Figure 5.2: Density of the sample of interpretations during the inference process. This figure illustrates the density of interpretation structures with a certain characteristic (energy value and recall value ranges) that are proposed at different iteration intervals of the inference using the MCMC based simulated annealing. Note that the first iterations (from 1 to 1000) are characterized by interpretations typically of low energy and low recall values (first column). Iterations going from 1001 to 1500 explore interpretation structures yielding higher recalls while having highly variable energy values – this is still the exploratory phase. The exploratory phase is more pronounced from iteration 1501 to 2500. The last iterations, ranging from 2501 to 3400, focus the search on configurations of lower energy, higher recall and precision. This shows two things: the energy formulation makes sense and the algorithm converges as it approaches the end of the inference.
Figure 5.3: Accepted proposals at different iterations. This is a sample of the MCMC inference for the video clip 0081-4. Significant changes in the configuration structure happen from one iteration to a farther away one (first row). Small structure changes occur from one iteration to a consecutive one (second row).

Figure 5.4: How soon until the best interpretation is found? This graph shows that the best interpretation is on average found earlier than it would be needed. The graph shows the average of precision and recall over all video shots at each iteration. The performance rates tend to go up at each iteration of the inference.
to what extent it recover severe classification errors (for example, when the classification score of the correct label of a feature is not among the 3 best scores). The pattern theory framework deals with the uncertainty of classifiers by introducing semantic bonds (i.e., bonds between action and object generators, which carry prior information) and by considering multiple interpretations for a single video unit (just like a list of results output by an information retrieval system).

5.7.1 Varying Levels of Classification Error Rate

In this section we investigate the robustness of the proposed inference formulation to gradual degradation of the underlying classification models used to measure the energy of support bonds (connections between feature generators and action/object generators). For each classification error scenario, we also verify how the performance changes as the semantic bond weights gets more and more influence over the total energy of the configuration in comparison to the support bonds – recall that the total energy of a configuration is a numeric quantity that determines the semantic quality of an interpretation (or video description).

Each degradation level is denoted by a classification error percentage; for instance, 10%, 20%, etc. The percentage indicates how often the classifier makes a mistake in labeling features. This idea is simulated using synthetic classifiers that output artificial classification scores for the features. A good classification score, which leads to the correct label of the feature, is generated with certain probability. For example, the classification model is wrong 10% of the time, then 90% of the time the feature labeling will be correct. Winning classification scores fall in the range 0.7 to 1 whereas weak classification scores belong to the range 0.0 to 0.4.

The obtained results are summarized in Figure 5.5. With the increase of the classification error rate, the performance (in this case the recall) gradually decreases by at most 10%. Such
performance reduction is no surprise but it is important to note that no abrupt changes happen. In fact, specially on the cases where the semantic bond weights are relatively higher than the support bond’s, the performance suffers not significant changes. The prior information carried out by semantic bonds is a key factor for the stability of the inference algorithm under undesirable conditions of classification errors.

Additionally, in this particular scenario, we observed that a support bond weight higher than the semantic bond’s favors better performance. However, this finding is for a specific case and cannot be generalized. The interested reader shall find the optimal choice of weights for the different types of bonds (semantic and support) using a training dataset. Such training dataset should be ideally small, yet representative of the data, for the optimal parameter search may have to follow an exhaustive mechanism.

5.7.2 Varying Levels of Weakness of the Classifiers

In the classification error scenario above, we only emphasized how often the classifiers are wrong in their prediction. Now we discuss a specific, yet important, case within such scenarios. How wrong are the classifiers? That is, if the classifier makes a mistake, is its prediction too far from the truth or did nearly make the right decision? Similar questions are: is its prediction biased towards a particular class or is it because the data point lied close to the decision boundary and other categories are nearly as likely. To test the framework under these scenarios, we simulate the cases in which the correct label scores appear as the second best, third best, fourth best and fifth best compared to the other label’s classification scores – each being treated as an independent scenario. For the experiments, we fix the overall classification error rate to 50% and let the semantic and support bond weights to be 0.3 and 0.7, respectively.
Figure 5.5: Performance rates for different degradation levels. The horizontal axis represents the percentage of feature classification errors. The performance rates in the vertical axis are rates of recall. 0% Ontology Influence means that semantic bond weights are discarded for the total energy of an interpretation. This means that the energy computation relies solely on support bond weights (which come from classification scores). 100% means that only semantic bond weights are considered for computing the interpretation’s energy. Any percentage between 0% and 100% indicates that both semantic and support bond weights are utilized.
We verify that the inference process behaves positively as the classification score rank of the correct label drops to lower ones. The results presented in Figure 5.6 show that performance improvements are achievable so long as the rank does not go lower than the third best. In the experiments, the highest improvement rate is of 47% when the score of the correct label is the second best among all other label's scores. Nonetheless, in cases where the correct label score is below the third score rank, the information carried out by semantic bonds compensate for the information error embedded in the support bonds, allowing for slightly better overall performance rates (see Figure 5.6).
CHAPTER 6: SPATIALLY COHERENT VIDEO DESCRIPTIONS

In this chapter, we study the performance of the proposed pattern theory framework in generating video descriptions in three main settings: 1) handling clutter of objects, 2) handling inference of descriptions when more than one event is happening, and 3) robustness to uncertainty in observations. An example of case 1) is shown in Figure 6.5 a).

6.1 Overview

The main activity occurs in the presence of multiple objects, where only a subset of objects participate in the main ongoing activity. The challenge is to be able to identify what object interactions form the predominant activity. In that example, the activity is "stirring carrots in a bowl using spatula", while other two objects, namely, a bowl and a cup remained unused. Note that, the resultant pattern-theoretic interpretation (Figure 6.5 c)) is formed by exactly the expected number of objects as encountered in the ground-truth interpretation (Figure 6.5 d)) with only a single incorrect generator (the tomato generator).

We evaluate the quality of the descriptions with two separate cooking scenarios data sets: YouCook and Breakfast Actions. The YouCook data set was used to analyze the performance of the framework for object clutter case and the Breakfast Actions data set was used to evaluate the multiple occurrence of events case. Overall, on the YouCook data set, the pattern-theoretic

---

5 Portions of this chapter were previously published in Fillipe D M de Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. Spatially coherent interpretations of videos using pattern theory. International Journal of Computer Vision, 2016. Permission is included in Appendix A.
interpretations turned out to be interpretations with higher quality than the baseline’s, both in recall and precision. There was a 63% improvement in recall and 150% improvement in precision (see Figure 6.1). One comparative example of the baseline-generated interpretations and the pattern-theoretic’s is shown in Figure 6.7.

![Figure 6.1: Performance results on the YouCook and Breakfast Actions data sets. (a) Recall and precision of the video interpretations on the YouCook data set. (b) Recall rate comparison with competitive methods on the Breakfast Actions data set [12]. Legend: “top 10” means that we consider the best interpretation out of the 10 best constructed interpretations (the 10 lowest energy interpretations found). “spatial coherence” means that it considers spatial coherence in the formulation. “DM” means discriminative method; in particular, the best performance rate reported in [12] using Random Forest and Support Vector Machines.](image)

On the Breakfast Actions data set, the pattern theory-based approach was successful at improving the recall rates of interpretations by more than 200% over the HMM-based approach without context-free grammar (CFG) presented in [12] (see Figure 6.1). There was also an improvement of about 34% when compared to [12]’s CFG-based approach, even though this is not a fair comparison with ours since that CFG model uses temporal information. The pattern theory approach without the notion of spatial coherence, as described in [1], also outperformed both the HMM-based and CFG-based approaches proposed in [12], increasing the recall rate by 97% and 21.4%, respectively.
6.2 Spatial Coherence Energy Term

We demonstrate how to handle video interpretation in the presence of clutter of objects and multiple events. We add a new energy term to the energy cost function described in Equation 4.1. This spatial coherence term takes into account spatial proximity of the region where the action is taking place and objects locations. Such term imposes spatial coherence to the relationship between actions and objects found in the interpretation.

We define spatial coherence as the grouping of actions and objects that are spatially correlated due to the location proximity of the features that explain them. This notion is enforced by measuring the energy between actions and objects as

\[
a'(\beta', \beta'') = \phi(g'_i, g''_i)a(\beta', \beta''),
\]

where \(\phi(g'_i, g''_i)\) corresponds to the spatial overlap ratio between the feature locations that \(g'_i, g''_i\) explain. The range of \(\phi(g'_i, g''_i)\) is \([0, 1]\) and \(a(.)\) accounts for the semantic constraint Equation 4.1 encourages connections between ontological generators whose features have high space-time overlap through \(\phi(.)\) and whose relationship has any semantic value through \(a(.)\).

6.3 Handling Clutter of Objects

We define clutter of objects as the multiple occurrence of objects in a scene, where only a few of these objects are involved in the ongoing main activity. It is common to observe clutter of objects in videos with unconstrained scenarios. In most cases, the predominant ongoing activity involve a subset of the visible objects at each point in time. For example, in the video clip depicted in Figure 6.6a) most ingredients and objects used to make the sandwich are visible but only a few
Figure 6.2: Illustration of where the spatial constraint quantification come from. Regions A-F are examples of area overlap between object bounding boxes or an object bounding box and the area where an action occurs. a) An example video showing the interaction “picking up spreader”. Just a few bounding boxes appear to show detected objects in the scene. b) Only the bounding box for the spreader object is completely overlapping with the area where the action occurs. c) An example video showing several detected objects are detected in a scenario where only one of them is participating in the interaction. d) Region C provides evidence through overlap ratio-based spatial constraint that the bread object is the only object participating in the predominant activity.
Figure 6.3: Robustness to different amounts of clutter of objects. This graph shows how well the algorithm handles recognition under different amounts of clutter of objects. The clutter related graphs show that the algorithm can handle well the cases in which clutter is high despite low precision. For low values of clutter, the algorithm does not disappoint. The amount of clutter of objects is given by the number of objects that do not participate in the interaction divided by the total number of objects that appear in the scene.

of them are active participants of the depicted activity (“picking up lettuce from a bowl”). The challenge is to be able to identify what object interactions form the predominant activity or, in order words, to be able to filter out the objects that can be discarded without loosing information needed for completeness of the description. In this particular clip, the cook picks up a handful of lettuce from the bowl containing lettuce and adds it to the bread that rests on the table. For this video, features from several objects that appear in the scene (marked with bounding boxes) were extracted and served as input to the inference algorithm; however, the goal is to identify which of these features provide support to represent the interaction characterizing the predominant event.

6.3.1 Spatially Coherent Interpretations under Clutter of Objects

At first, our inference approach considers all input features together to build a global descriptive activity interpretation of the target video. Then it decomposes the interpretation into
smaller, locally meaningful interpretations. These smaller interpretations could belong to different spatial regions in the scene and they may or may not have any relevant semantics for interpretation of the video scene. The decomposition happens by extracting the connected components from the global interpretation. In particular, we select the largest connected component containing at least one action generator to describe the main relevant activity in the scene – this is a reasonable assumption for our case because we know that each video unit used for interpretation involves at least one action, which is always the one expressing the semantics of the main ongoing activity. This simple observation makes the connected component a fair solution for the problem of filtering out undesired elements for interpretation of the scene. On the other hand, if multiple separate events are happening and are important for the overall description of the scene, then clearly multiple interpretations must be considered – we study this case in the next section of this chapter.

The largest connected component containing at least one action generator is cast as the final interpretation of the main ongoing activity in the video. The assumption is that the predominant activity involves an action and the largest number of objects that spatially overlaps the region where the action is occurring. In that sense, the final interpretation describes action-object interactions whose connectivity is induced by both semantic correlation and spatial proximity constraints. The spatial proximity constraint is captured by the spatial coherence term embedded in the energy cost function (see Section 6.2). Action and object generators whose feature generators do not overlap in space are less likely to be found connected in an interpretation for the spatial coherence term will have a negative impact to the overall energy of the configuration (or interpretation).
6.3.2 Experimental Results and Discussion

Figures 6.3 a) and b) show the performance rates of the quality of descriptions under different conditions of clutter found among the test videos – the x axis indicates that videos with up to a certain amount of clutter was considered to compute the overall performance rate. As videos with larger amount of clutter are considered, the recall rates does not necessarily drop but remain steadily in the range 0.5 and 0.6 despite the fact that a smaller number of generators composing the final interpretation. We notice that for the approaches not leveraging the spatial coherence term, the recall rates were improved. This is expected to happen since the number of objects in the scene also increase, so does the number and diversity of the generators composing the final interpretation; therefore increasing the chances of including the correct content of the target interpretation. However, the weakness of the approaches not relying on the spatial coherent term is revealed in the precision rate plot. Essentially, showing an improvement in the recall rate does mean an overall improvement in the quality of the interpretation.

In Figure 6.3b), we notice a gradual (not abrupt) decrease of the precision rate by the proposed approach that uses the spatial coherence term. This is a natural phenomena since the amount of clutter is larger and the spatial coherence term is more selective. But the important observation here is that the spatial coherence term allow for improvements of the overall quality of the interpretations despite the larger amount of clutter. This means that the content of the interpretations is not considerably affected by the increase in clutter. We can confirm this by noticing that the overall recall rate does not drop consistently with an increase in the number of videos with more clutter (see Figure 6.3 a)).

64
A successful case under large amount of clutter is shown in Figure 6.6. The ground-truth interpretation consists of three generators only, namely: bowl, lettuce, and pickup. The final pattern-theoretic interpretation, as depicted in Figure 6.6 c), carried the ground-truth interpretation (content highlighted in red) and filtered out more than half the amount of existing clutter (six other generators).

The case in which the amount of clutter is below 40% does not exhibit any performance pattern. By looking into a sample of output interpretations from these cases we noticed that the cases of failure occurred because the support bonds were more influential than the semantic bonds in reaching a low-energy configuration, meaning that the prior knowledge was not sufficient to successfully correct the interpretation. An example of this case is shown in Figure 6.4. Because of the classification bias in favor of the class plate, the baseline algorithm produces an interpretation with two correct generators; however, its interpretation does not carry the expected semantics. Contrarily, the PT inference algorithm attempts to correct that bias with semantic bonds but ends up outputting an interpretation semantics that resembles the ground-truth's but with incorrect object generators.

6.4 Handling Multiple Occurrence of Activities

One important case is handling inference for description of multiple, independent, ongoing activities in a video. As follows, we describe the experimental setting for simulating simultaneous occurrence of activities (using the Breakfast Action dataset) and then a discussion on the performance of the inference process when describing videos where multiple activities are occurring simultaneously.
Figure 6.4: (a) Example video depicting the interaction “pick up meat from the plate”. A test video for which the PT-based inference algorithm fails to output the correct interpretation. (b) Graph structure representing the ground-truth interpretation for the test video depicted in (a). (c) The best interpretation output by the PT inference algorithm: “pick up lettuce from bowl”. (d) The second best interpretation output by the PT inference algorithm: “put down tomato in a bowl”. (e) The second best interpretation output by the PT inference algorithm. (f) The interpretation output by the baseline algorithm: “pick up plate”.
Figure 6.5: Example 1 of interpretation under object clutter scenario. (a) Illustration of a video showing an interaction performed for the recipe “making a carrot cake”. (b) Pattern-theoretic interpretation using all feature generators: “stirring tomatoes in a bowl using spatula”; “tongs”; “egg”. (c) Final pattern-theoretic interpretation as the largest connected component from the configuration shown in (b). That is, the subconfiguration more likely to be spatially coherent and describe the predominant ongoing interaction. The interpretation is “stirring tomatoes in a bowl using spatula”. (d) Ground-truth interpretation of the video depicted in (a): “stirring carrots in a bowl using spatula”. The generators highlighted in red indicate correctly detected actions and objects.
Figure 6.6: Example 2 of interpretation under object clutter scenario. (a) Illustration of a video showing an interaction performed for the recipe “making a sandwich”. (b) Pattern-theoretic interpretation using all feature generators: “picking up lettuce from two bowls using tongs”; “bowl”; “cup”; “spreader”; “tomato”; “pan”; “bread” (c) Final pattern-theoretic interpretation as the largest connected component from the configuration shown in (b). That is, the subconfiguration more likely to be spatially coherent and describe the predominant ongoing interaction. The interpretation is “picking up lettuce from two bowls using tongs”. (d) Ground-truth interpretation of the video depicted in (a): “picking up lettuce from a bowl”.

68
Figure 6.7: Example 3 of interpretation under object clutter scenario. (a) Illustration of a video showing an interaction performed for the recipe “making cookie dough”. (b) The baseline interpretation is fragmented such that no relation between the detected actions and objects are established. It simply list them: “pouring”; ‘pan”; “pan”; “pan”; “pan”. (c) Final interpretation by the pattern theory approach with spatial coherence: “pouring tomatoes in a bowl”. (d) Ground-truth interpretation for the video depicted in (a): “pouring something from cup into a bowl”.

6.4.1 Video Units with Multiple Occurring Activities

We synthesized videos with simultaneous occurrence of activities by joining side by side two video units from the original temporal segmentation provided by the Breakfast Action dataset. For each original video unit, we randomly select a different video unit to form a pair of activities occurring simultaneously in a scene. To compute the performance rate, we evaluated the interpretation output for each activity separately and average the performance over all unique video units. If an event occurs in more than one multiple-activity video unit, we either compute the average recall across all given interpretations or select the highest recall rate among the given interpretations for the same unit (see Table 6.1).

6.4.2 Experimental Results and Discussion

We evaluate the effectiveness of two mechanisms that use the idea of spatial coherence to handle cases of simultaneous activities. One uses an additional energy term (Section 6.2) and the other extends the former with a new proposal function that makes corrections to the interpreta-
tion based on violations of spatial location constraints. We named them PT+SC and PT+SC+SP, respectively. In Table 6.1, we verify that the general pattern theory model PT+SC yields better results than the original formulation that does not include the spatial coherence term [1] and than its more elaborate counterpart PT+SC+SP, which makes use of a spatial coherence-aware proposal function. We speculate that the latter has failed for adding a new proposal may require a larger number of iterations in order to achieve a significant beneficial effect. PT+SC+SP produced better results than [1] but not sufficiently higher (see Figure 6.8). Figure 6.8 presents the performance curves when considering up to the top 10 interpretations for each video unit. It is clear that improvements to the energy cost function can have a high impact on the quality of the interpretations.

Qualitative illustrations of this effect is shown in Figure 6.9 (b), cases in which the proposed approach is able to cluster the spatially correlated feature generators and find independent semantic interpretations for each group. Despite the presence of another event, it successfully recovered the correct semantic interpretation for the event on the left, for which [1] failed. This same effect can be verified in Figures 6.9 (a), (c) and (d). The interpretations on the top of Figure 6.9 shows the results by PT+SC and at the bottom the ones by [1]. When confronted with the scenario of simultaneously occurring events, interpretations output by [1] have closed bonds (highlighted with dashed lines) between action and object generators that are supported by feature generators from different events. This type of confusion is successfully attenuated by our reformulation of the general pattern theory model.

Table 6.1: Performance comparison for simultaneously occurring events.

<table>
<thead>
<tr>
<th></th>
<th>[1]</th>
<th>PT+SC</th>
<th>PT+SC+SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>27.8%</td>
<td>30.6%</td>
<td>28.3%</td>
</tr>
<tr>
<td>Max</td>
<td>38.6%</td>
<td>42.6%</td>
<td>40.3%</td>
</tr>
</tbody>
</table>
Figure 6.9: Scenarios of simultaneously occurring events. Each case (a), (b), (c) and (d) depicts a different scenario of simultaneously occurring events. The interpretations on the top were generated using the pattern theory framework that takes into account spatial coherence and the interpretations on the bottom were output by [1].
6.5 Summary and Discussion

In this chapter, we demonstrated how the proposed pattern theory framework for video description can handle object clutter and simultaneous activities by introducing the concept of spatial coherence. The evaluation is conducted using two contemporary datasets of the cooking scenario. This includes one with more 5000 videos of activities and more than 40 categories of actions and objects combined. The obtained results demonstrate that the proposed approach is consistently better than other competitive approaches using hidden Markov models and Context-Free Grammars. The experiments showed substantial numerical performance improvement under challenging scenarios exhibiting clutter of objects and multiple occurrence of activities. We have also evaluated the robustness of the proposed approach to varying degrees of classification error and demonstrated that significant performance boost is achievable, specially when the scores ranks of the correct feature labels are not so low. With the activity pattern-theoretic framework, we are able to i) provide a principled mechanism for generating semantic interpretation of activities from real-world scenarios, ii) handle a massive amount of data without the need for structure learning, and iii) incorporate domain-specific knowledge for representation and inference, all in one formalism.
CHAPTER 7: TEMPORALLY COHERENT DESCRIPTIONS FOR LONG VIDEOS

So far we have focused on analyzing short videos involving isolated actions. In this chapter we demonstrate how we can use the pattern theory framework to extend the inference process to compute a description for a sequence of activities. A sequence of activities is defined as a chain of consecutive interactions, where an interaction is a human performing action with and/or using a single or multiple objects; thus, an example of a sequence of activities is “pick up spatula, pick up bowl, put down eggs into bowl, stir eggs in a bowl using a spatula”. To this end, we introduce the concept of temporal bonds, a new type of bond to take into account the temporal structure/relationship across activities. Temporal bonds connect across individual action generators to enable semantic interpretations of longer videos. Longer temporal connections improve scene interpretations as they help discard (temporally) local solutions in favor of globally superior ones. Using this extension, we demonstrate improvements in understanding longer videos, compared to individual interpretations of non-overlapping time segments. We verified the success of our approach by generating interpretations for more than 700 video segments from the YouCook data set, with intricate videos that exhibit cluttered background, scenarios of occlusion, viewpoint variations and changing conditions of illumination. Interpretations from long sequences of video segments were able to improve the quality of the video descriptions by about 70% and, in addition, proved to be more robust to different severe scenarios of classification errors.

6Portions of this chapter were previously published in Fillipe Souza, Sudeep Sarkar, Anuj Srivastava, and Jingyong Su. Temporally coherent interpretations for long videos using pattern theory, In CVPR, 2015 [18] Permission is included in Appendix A.
7.1 Overview

The problem of understanding activities in video data and providing meaningful semantic interpretations is very important. In recent years, a variety of solutions have been proposed and, among other ideas, the techniques based on encoding scene structure using graphs have shown promise in this problem area. These approaches represent items of interest – objects, actors, actions, etc. – as nodes in graphs and ascertain their interactions through graph edges. The main advantage of this framework is that one can naturally associate probability models with such graphs, thus providing statistical interpretations to solutions. Also, one can use both prior knowledge and the current data to deduce optimal interpretations in a coherent way. The main limitation in the current graph-theoretical solutions has been the rigidity of graph structures. In most cases, the graph geometries (connectivities, neighborhoods, etc) are pre-determined and only the node values are allowed to be variable. Even when the edges are allowed to change, they are usually based on a simple thresholding, or decisions that are spatiotemporally local, i.e. isolated from other nodes.

In Souza et al. [1], as described in Chapters 3 and 4, we introduced a flexible graph-theoretical approach for describing videos based on Grenander’s pattern theory [62]. The flexibility of the approach comes from the fact that both nodes and edges are allowed to be variables and are inferred from available knowledge. There are two dominant sources of knowledge: (1) the prior in form of frequency tables of concept co-occurrences, contextual knowledge about actions represented by the underlying ontology extracted from previous annotated videos, and (2) objects and actions detected using machine learning techniques, and their detection scores, in the current video. In [1], we only studied short videos containing individual actions (pick up, put
Figure 7.1: Overview of the pattern theoretic framework proposed in [1]. (a) shows basic elements of this framework: a generator space containing basic ontological elements of representation called generators, machine learning-based concept classifiers and prior knowledge in terms of frequency tables of concept co-occurrences. (b) shows a pattern theoretic video interpretation that is a combination of generators. Connections between generators that represent ontological concepts indicate occurrence of certain interactions. Features are connected to ontological generators to support their semantic value in the interpretation.

down, pour, stir, etc) and demonstrated the strength of this pattern theoretic approach and the flexibility of its representation. With this approach, one does not need to explicitly model each of the variants for an activity (interactions of objects over the course of an action). This approach is capable of discovering hidden events or events not previously considered during annotation phase. An illustration of this pattern theoretic framework is shown in Figure 7.1. The original formulation [1] of our pattern theory-based video description framework does not perform well on videos depicting a sequence of activities and the temporal relations between them. If those videos are split into smaller, disjoint video segments, where each shows only a single activity, then the original formulation can perform individual inferences on each segment and the over-
Figure 7.2: Illustration of advantage in using temporal bonds. Top rows shows frames from two consecutive segments of a video. The first segment depicts the interaction *put bowl down* (the small one with the left hand) and second segment depicts *stir ingredients in a bowl using spatula*. (e) shows [1]'s interpretations for both segments. (f) shows our approach's interpretation for both segments. Shaded circles denote correctly identified generators.

All interpretation can be both inconsistent and sub-optimal. In this chapter we pursue a more comprehensive approach by introducing temporal bonds across sub-configurations that represent individual activities (actions performed on/with objects). This additional structure enables us to discard (temporally) local solutions in favor of globally optimal and temporally consistent configurations, as illustrated in Figure 7.2. We demonstrate these ideas on a recent challenging data set of cooking scenarios, the YouCook dataset. YouCook's videos depict high-level cooking activities in unconstrained scenarios, with cluttered background, clutter of objects, variable conditions of illumination, different viewpoints and camera motion.
7.2 Modeling Long Video Sequences with Temporal Bonds

In this section, we analyzed the numerical performance and qualitative advantages of using the pattern theoretic approach with temporal bonds. First, we evaluated the quality of output interpretations by analyzing samples taken from the experiments. We discussed the effects of adding temporal bonds to the bond structure of generators and in which scenarios temporal bonds can lead to more interesting (desirable) interpretations. We also analyzed how critical the inclusion of temporal bonds to the model is when interpretations are based on multiple segments of videos. Then, we evaluated the performance in controlled scenarios of classification errors stemming from synthetic concept classifiers. We finalized our discussion with a comparative performance analysis on the YouCook data set when using real machine learning based concept classifiers. For comparative analysis, we contrasted the performance profile of the proposed approach with [1]'s and a baseline algorithm that generates interpretations exclusively based on the best classification scores using linear-SVM classification models (i.e. a purely machine learning-based method). The performance metric consisted of counting the number of correct ontological generators found in the interpretation given the ground-truth's. The highest performance rate is 1 and lowest is 0. For example, the performance rate of the interpretation in Figure 7.5j is 0.86.

7.2.1 Temporal Bonds

Temporal bonds allow the pattern theoretic process to take into account temporal dependence information between consecutive actions, accordingly, interpretations. We found several cases in which temporal bonds helped identifying the correct actions across multiple consecutive video segment interpretations. Four of these cases of success are illustrated in Figures 7.3–7.5.
Figures 7.3c and 7.3d show two descriptions generated by applying the original formulation presented in [1] on each of two consecutive video segments. Recall that each video segment depicts a single action that results in an interaction with one or multiple objects. These descriptions were generated by optimizing the energy cost function for each video segment, separately. The method without temporal bonds fails to find the correct action for the first video segment, interpreting it as *pick up* instead of *put down*. Contrarily, the pattern theoretic description using temporal bonds, shown in Figure 7.3i, successfully identifies the action *put down*, while maintaining the proper semantic bonds captured by [1]. Simply applying the inference without the addition of temporal bonds to the models produces locally optimal descriptions that do not capture the temporal structural constraints. In fact, the inference without temporal bonds for the description illustrated in Figure 7.3g introduced more errors, confusing *pick up* by *put down* in both segments.

Another example of success with the use of temporal bonds is presented in Figure 7.3j, where the action *stir* was correctly inferred as a result of correctly inferring the interaction occurring in the second segment. This can be verified by observing that without temporal bonds the individual inference for a single segment resulted in a description with the action *pick up* and the inference for multiple segments without temporal bonds resulted in a description with the action *season*. In both illustrated cases, not only was the approach with temporal bonds able to generate improved semantic descriptions but also able to preserve relevant bond connections and object generators correctly identified in the single-segment based inference not using temporal bonds [1]. This same effect has been observed for a longer time window of consecutive segments of activities, as illustrated in Figures 7.4 and 7.5. We discuss these cases in the next section to make an argument in favor of generating descriptions for larger temporal windows containing
multiple video segments. Since more degrees of freedom are available when considering temporal bonds, this permitted our approach to explore other possibilities of interpretations with more confidence than when they were not present.

7.2.2 Interpretations for Sequence of Activities

We identified several cases in which interpretations generated for multiple-segment temporal windows using our approach helped determine the correct interpretations of actions for video segments that are misinterpreted by [1]'s approach. Temporal bonds allow our approach to not only search for coherent local interactions but also naturally focus on identifying the correct temporal ordering of actions in adjacent video segments. For instance in Figures 7.4 and 7.5, our approach's interpretation (Figures 7.4j and 7.5j) was able to preserve the correctly detected objects found in interpretations generated by [1] (Figures 7.4e-7.4i and Figures 7.5e-7.5i) while fixing the action interpretations of consecutive video segments. More interestingly, the case depicted in Figure 7.4 shows that the temporal bond-based approach's video interpretation leveraged the confidence of the action interpretation in the third segment, put down, to propagate multiple corrections in the two past segments and the video segment ahead; the action interpreted sequence was put down → pick up → put down → pick up (Figure 7.4j). Nonetheless, the same effect was not observed when using [1]'s method, which produced the sequence pick up → put down → put down → put down (Figure 7.4i). We only explored the use of temporal bonds at the level of actions, under the assumption that the semantic coherence with respect to the participating objects in the interactions is mostly dependent on correctly identifying the ongoing action. The focus then revolves around finding temporal coherence between consecutive actions. Identifying the correct sequence of actions indirectly influences on the quality of the overall video description.
Figure 7.3: Results of interpretations for sequences of two activities. Illustrations in (a) shows a sequence of two consecutive activities, steps for making french toast. (b) illustrates a sequence of two consecutive activities, describing steps for making dough. The pairs (c)-(d) and (f)-(g) show the corresponding interpretations by [1] based on single-segment windows, while (e) and (h) are derived from two-segment windows. (i) and (j) present corresponding interpretations generated by our approach.
Figure 7.4: Result for a sequence of four consecutive activities (making salad). (a)-(d) illustrate four consecutive video segments describing steps for making salad. (e)-(h) show interpretations by [1] based on single-segment windows, and (i) for the four segments at once. (j) depicts the interpretation by our approach.
Figure 7.5: Result for a sequence of four consecutive activities (making sandwich). (a)-(d) illustrate four consecutive video segments describing steps for making sandwich. (e)-(h) show interpretations by [1] based on single-segment windows, and (i) for the four segments at once. (j) present the interpretation by our approach.
7.3 Experiments with Synthetic Classifiers

Classification scores help measure the quality of an interpretation by quantifying the support bonds; therefore, they are essential for ascertaining the global optimal interpretation. We organize two controlled experiments for simulating two main scenarios in which the classification models of objects and action exhibit poor prediction performance. The goal is to evaluate the performance robustness of the proposed approach with respect to the quality of the classification models. In one first scenario, we varied the classification error rates of the classifiers from 10% to 60%. In these cases, the classification score ranks of the correct labels for the affected features are the second best. Then, we fixed the classification error rate to 50% and vary the score rank of the feature's correct labels from 2 to 5.

Figure 7.6 shows the performance profile of the approaches for increasing rates of classification error. In Figure 7.6a, where interpretations are generated for each individual video segment, our approach and [1]'s were superior to the baseline's, but only for high rates of classification error (>20%). In this same case, our approach and [1]'s had comparable performance rates, since no temporal data could be explored. Our approach produced performance improvement increase of more than 7% over [1]'s for larger temporal windows, multiple video segments at once (Figure 7.6b). In summary, if the concept classifiers are not sufficiently good to be used alone, these results indicate that ontology-based approaches like ours and [1]'s are imperative in order to achieve reasonably sufficient performance.

Figure 7.7 shows the performance profiles in which approximately 50% of the features had their correct label's classification scores as the kth best classification scores, where k varies from 2 to 5. Overall, Figures 7.7a-7.7b show that our approach was consistently capable of correcting
the feature labeling, even when the feature correct label had the fifth best classification score. Figure 7.7b shows that for multiple-segment based inference our approach is consistently superior to [1]'s, with up to 12% increase. This suggests that under uncertain scenarios, our approach would be more advantageous because it improves performance by generating video interpretations based on multiple segments.

Figure 7.6: Interpretation performance for varying scenarios of classification error. Error rates ranged from 10% to 60%.

7.4 Experiments with Real Classifiers

Overall the temporal bond-based approach improved the total average video description performance by approximately 5 times the baseline’s and by ~30% the approach not using temporal bonds [1] (Table 7.1). Three types of bonds contribute to measure the quality of the video descriptions. Figure 7.8 shows how the overall average video interpretation performance varied as certain types of bonds were given more participation weight than others. In all cases, overweighing support bonds dropped the interpretation performance rate to the baseline’s, which was low because of the weak concept classifiers. More weight on the participation of temporal bonds was
Figure 7.7: Robustness to decreasing classification score ranks for correct label. The classifiers have 50% chance of misclassifying the video features.

sufficient to achieve higher performance rates for all multiple-segment cases (Figure 7.8b). This emphasizes our assumption that correct action interpretations should naturally lead to correct identification of the true involved objects. Overweighting semantic bonds was influential mostly in the single-segment case (Figure 7.8a), where temporal bonds were not relevant.

We also observed the average performance when considering the top $k$ interpretations for describing each video. Figure 7.9a shows that our approach and [1]'s had comparable performance for the single-segment case when no temporal information is available. When generating multiple interpretations for temporal windows containing multiple consecutive segments, our approach provided the best overall performance. In comparison to [1]'s original idea of analyzing individual segments, it nearly doubling the performance, with improvements ranging from about 67% to 73%, depending on the number $k$ (compare Figure 7.9b with Figure 7.9b).
Table 7.1: Average performance of interpretations for increasing temporal windows. The window size is given by the number of video segments that composes the whole sequence of activities.

<table>
<thead>
<tr>
<th>Method</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>0.40</td>
<td>0.52</td>
<td>0.50</td>
<td>0.52</td>
</tr>
<tr>
<td>Souza et al.</td>
<td>0.41</td>
<td>0.44</td>
<td>0.41</td>
<td>0.46</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.11</td>
<td>0.12</td>
<td>0.13</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Figure 7.8: Influence of bond types on the performance quality of interpretations. Video interpretations vary when different bond types have more participation than others.

Figure 7.9: Performance rates for the top \( k \) best interpretations.
7.5 Running Time

For all the experiments we fixed $k = 3000$. The running time grows linearly with the number of feature generators $n_f$, consistent with our analysis (see Table 7.2).

Table 7.2: Average CPU+I/O time of videos per number of feature generators $n_f$. Machine spec: 4 16-core 2.3 GHz CPUs (AMD Opteron 6376), 16 16GB RAM units.

<table>
<thead>
<tr>
<th>$n_f$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec</td>
<td>16</td>
<td>26</td>
<td>36</td>
<td>46</td>
<td>56</td>
<td>67</td>
<td>78</td>
<td>89</td>
<td>101</td>
<td>113</td>
<td>126</td>
<td>169</td>
<td>178</td>
<td>226</td>
<td>206</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n_f$</th>
<th>24</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>33</th>
<th>34</th>
<th>36</th>
<th>37</th>
<th>44</th>
<th>46</th>
<th>48</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec</td>
<td>235</td>
<td>229</td>
<td>257</td>
<td>252</td>
<td>301</td>
<td>279</td>
<td>342</td>
<td>382</td>
<td>399</td>
<td>375</td>
<td>477</td>
<td>513</td>
<td>550</td>
</tr>
</tbody>
</table>

7.6 Summary and Discussion

In this chapter we described and evaluated a modification of the video description framework proposed in [1] that introduces a new bond structure to the model to capture temporal dependence across consecutive actions. This allowed us to generate temporally coherent semantic descriptions of videos. Similar to [1], the basic units of interest (i.e., actions and objects) are denoted by generators that combine to each other to form graphical structures, which represent video interpretations. The quality of an interpretation is governed by the energies of its bonds. These bond energies are defined using classification scores and frequency tables of concept co-occurrences, which help define and seek optimal configurations. While previous applications have been restricted to analyzing short videos containing isolated actions, we have extended this idea to longer videos using additional bond structures, which allow interactions between actions that are adjacent in time. The aforementioned experiments, involving more than 700 video segments from the YouCook data set, demonstrated the power of adding action temporal bonds in the configurations. Not only did we improve the performance in detection of generators but we
also improved the overall scene interpretations. In addition, our approach was more robust to
degradation in feature-level classification performance than its counterparts.

The use of additional bond structures clearly helped interpret more complex scenes and
allowed for enhanced inferences. In view of the flexibility of this pattern theoretic framework in
representing complex systems, in future work, we plan to include additional (types of) gener-
tors and bond structures. In this future extension, configurations that represent interpretations of
small video segments can be turned into composite generators, naturally augmenting the repre-
sentation hierarchical system, which can be used to help understand even longer videos depicting
more complex activities.
CHAPTER 8: INTERPRETATION OF ACTIVITIES WITH MULTIPLE MODALITIES

In this chapter we demonstrate how to construct structured semantic descriptions of activities in videos leveraging both auditory and visual features. We use the formalism of Grenander’s structures [65] to seamlessly integrate both modalities in the inference process; thus, allowing for inference on multiple sensory data in a unified framework.

8.1 Overview

The proposed structured model performs probabilistic reasoning on multiple decisions from classification models of individual modalities (audio and vision) and different classes of objects and actions. Classification decisions are prediction scores, typically ranging from 0 to 1. These prediction scores are available for each pair of modality (or type of feature) and action or object label. For the inference, we use a greedy approach such that, for each feature in consideration, only the action or object labels with the $k$ highest prediction scores are considered for deriving the video description. Grenander’s structures use structural semantic information of the domain to weigh the feature support provided by the classification predictions of object and action models when evaluating the quality of a description. Such weighing mechanism imposes semantic consistency on video description. An illustration of our contribution is shown in Figure 8.1. The main contribution of this chapter is to exploit Grenander’s structures as a means to facilitate and enrich the video description model described in previous chapters with multimodal sensory data. In particular, we consider the integration of auditory features with visual features.
Here we also present the first results on structured semantic understanding of audio-video events using the CMU Kitchen dataset published in [66].

8.2 Integrating Data from Multiple Modalities into the Model

We consider that every data modality can be cast a new type of feature. This way, the multimodal capability is introduced to the model by simply adding new generators to the generator space, generators that account for new types of features. To test the ability of our video
description framework in handling inference with multiple data modalities, we introduced spectrogram features (auditory features), histogram of optic flow, convolution neural network features for modeling actions. Note that in previous chapter experiments, each class of action could only be explained by a single type of feature.

To enable the multimodal capability to the model, we update the model presented in previous chapter with new types of feature generators. The SPECT generator was added to represent auditory features and its bond structure formed by a single out-bond of bond value action (see Figure 8.2). We also added two other new types of feature generators to the generator space $G$ to account for CNN features of objects and actions, namely, the CNN and CNNFLOW generators. The CNN generator has an out-bond of bond value object and CNNFLOW generator has an out-bond of bond value action (see Figure 8.2). Each CNN feature generator is associated with a deep classification model, either for classification of actions or for classification of objects. These classification models are multi-class linear-SVM classifiers trained on CNN features.

![Figure 8.2: Overview of the proposed approach. Semantic reasoning on the top $k$ labels scored on CNN and auditory features for generating semantically consistent interpretations of audio-video events using Grenander’s structure.](image)

Two generators $g_i$ and $g_j$ connect through compatible bonds. The meaning of such connection is determined by their bond values; for example, the generator stir has an out-bond of bond value stirrer, such that any other generator that connects to that out-bond will serve the role of a stirrer in the event. The strength of compatibility between bonds is quantified by the acceptor...
function

\[ A(\beta'(g_i), \beta''(g_j)) = \exp(q(g_i, g_j)\tanh(f(g_i, g_j))) \] (8.1)

where \( f(.) \) is a scoring function that measures the compatibility of connecting the labels by \( g_i \) and \( g_j \) through their respective bonds \( \beta' \) and \( \beta'' \). If \( g_i \) is a feature generator and \( g_j \) is a generator representing an action or object label, then \( f(.) \) responds as the classification score associated with the classifier of \( g_i \) for the label represented by \( g_j \). If both \( g_i \) and \( g_j \) represent action/object labels, then \( f(.) \) represents the entry value of the frequency table that counts the co-occurrence of labels describing events of the target domain. \( q(.) \) weighs the rescaled score output by \( f(.) \) depending on what type of bond is formed between \( g_i \) and \( g_j \). If \( (\beta'(g_i), \beta''(g_j)) \) forms a support bond, then we let \( q(.) = 1.5 \), otherwise, if a semantic bond, \( q(.) = 1.0 \). This means we are emphasizing the support given by the classification scores more than the prior.

### 8.3 Experimental Results and Discussion

The Carnegie Mellon University Multimodal Activity database [66] contains multimodal measures of human activity, performing tasks that involve cooking and food preparation. The dataset contains five different recipes: brownies, pizza, sandwich, salad and scrambled eggs. The following modalities were recorded: high and low resolution videos and five microphones. We carried out our experiments using brownie recipe videos only since only those videos had their fine-grained annotation of events available. Spriggs et al. [?] generated the ground truth for some videos and recipes. In total, there are 13 event brownie labeled videos. For training, we use videos identified by numbers 7, 8, 13, 14, 17 and 19. The test set was formed by videos numbered 9, 12, 16, 20, 22 and 24. In brownie recipe dataset, we can find 12 action labels, namely, \textit{stir},
crack, spray, twist, etc. and 14 object labels, including baking pan, bowl, brownie box, oil, fridge, etc. The test set consisted of 233 event video segments. We evaluated and compared different combinations of features with different inference approaches (pure machine learning - ML and pattern theory - PT). For actions, we chose histograms of optical flow (HOF), convolution neural network based on motion (CNN Flow) and histograms of audio features based on spectograms (SPECT). As for object, we chose appearance convolution neural network (CNN) and histograms of oriented gradient (HOG). This way, the combination PT cnn-cnnflow means that the inference and representation were modeled with pattern theory using object models based on CNN features and action models based on CNN Flow features.

8.3.1 Video Descriptions based on Discriminant Models

Discriminant models such as support vector machines (SVM) [10] and neural network (NN) [?] are widely used to label actions [67] and objects [68] directly from video features. We implemented a strategy (which we called ML-based labels) based on linear-SVM classification models to generate semantic interpretations of events based on auditory and visual features. Given a set of auditory and visual features from a video, each feature is labeled according to the best classification score. The resulting set of labels represents the semantic understanding of the video. The best semantic interpretation is the one formed with all labels retrieved from best classification scores of the features. Thus, the $k$th best interpretation is formed by labels retrieved from the $k$th best classification scores of the features. This strategy ignores the structural semantic information of the domain, relying solely on the confidence of the classification scores to build the interpretations.
8.3.2 Video Descriptions as Structured Output on Deep Features

Grenander’s structures jointly with priors leverage the evidence and confidence provided by the classification scores of deep models better than its counterpart that use no structural information and rely on support of deep models decisions alone. Figure 8.3 shows that the performance of interpretations by the deep models with Grenander’s structures are often superior over the top 10 interpretations. Grenander’s structures help the inference algorithm to exchange highly confident classifier’s choices of labels by less confident ones that improve the semantic consistency of the interpretations; thus, improving the overall quality of the interpretations. Examples of these are illustrated in Figure 8.7, for example, the exchanges of put by crack, bowl by brownie box and oil by egg.

Although the methods employing HOF and HOG features generally show lower performance when compared to those built on CNN features, Figure 8.4 shows that the method with HOF and HOG features combined with Grenander’s structures can achieve performance rates comparable to methods employing CNN features without Grenander’s structures. This suggests that a structured model based on pattern theory can be potentially used to boost the performance of models using traditional features and have comparable performance to the state-of-the-art models using CNN features only; therefore, serving as possibly a less costly alternative if training and using deep models are computationally demanding for a specific task. In summary, the interpretations supported by deep features and Grenander’s structures had the highest performance rates, leading both recall and precision rates. Table 8.1 shows the overall performance rate of each method, considering up to the top 10 interpretations. Once more CNN features were proven to be superior to the traditional combination of feature histograms such as HOFs and HOGs.
Figure 8.3: Interpretations with deep features and Grenander’s structures.

Figure 8.4: Comparative interpretation performance with traditional and deep features. Semantic description with traditional features (HOFs and HOGs) coupled with Grenander’s structures (PT hog-hof) has achievable performance rates comparable to the semantic description models relying solely on classification score support of deep models (ML cnn-cnnflow).
8.3.3 Video Descriptions Built on Sound and Vision

The methods employing only auditory features for the recognition of actions were the most positively sensitive to the presence of domain knowledge imposed by Grenander’s structures. For example, when ignoring motion features, there was a performance rate improvement of 11.5% and 12.3% in precision and recall (see Table 8.1), respectively. Grenander’s structures allowed the latent discriminating power of auditory features to become visible, which was reflected by having PT cnn-spect outperform ML cnn-spect by more than 10% in all performance metrics. We also observed that this method (PT cnn-spect) achieved comparable performance rates to more computationally heavier methods that depend on motion features, namely, PT cnn-cnnflow and PT cnn-cnnflow-spect. This suggests that audio features could be potential surrogates for the discriminating power offered by motion features while requiring less computational power; thus, allowing for implementation strategies of the low-level video processing layer that are computationally less expensive.

Qualitatively, this improvement was reflected mostly on selecting the right action to describe the event, correcting 47.8% of all test events. The most corrected actions were pour (26%), take (23.3%) and stir (21.9%). On the other hand, the sound feature support was not as positively complimentary to deep visual features (cnnflow) as we expected in building the semantic understanding of events. The method combining deep visual features with auditory features (PT cnn-cnnflow-spect) corrected just as many cases of wrongly labeled actions as did the method with deep features only (PT cnn-cnnflow) when contrasted with their counterparts supported with HOF and HOG features. Additionally, PT cnn-cnnflow-spect did not improve any interpretation case missed by PT cnn-cnnflow. In Table 8.1, overall performance rates were equivalent.
Table 8.1: Interpretation performance with sound and vision features.

<table>
<thead>
<tr>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML hog-hof</td>
<td>0.583</td>
<td>0.651</td>
</tr>
<tr>
<td>PT hog-hof</td>
<td>0.625</td>
<td>0.692</td>
</tr>
<tr>
<td>ML cnn-cnnflow</td>
<td>0.631</td>
<td>0.694</td>
</tr>
<tr>
<td>PT cnn-cnnflow</td>
<td><strong>0.667</strong></td>
<td><strong>0.734</strong></td>
</tr>
<tr>
<td>ML hog-spect</td>
<td>0.583</td>
<td>0.649</td>
</tr>
<tr>
<td>PT hog-spect</td>
<td>0.63</td>
<td>0.696</td>
</tr>
<tr>
<td>ML cnn-spect</td>
<td>0.585</td>
<td>0.651</td>
</tr>
<tr>
<td>PT cnn-spect</td>
<td><strong>0.657</strong></td>
<td><strong>0.726</strong></td>
</tr>
<tr>
<td>ML hog-hof-spect</td>
<td>0.593</td>
<td>0.659</td>
</tr>
<tr>
<td>PT hog-hof-spect</td>
<td>0.631</td>
<td>0.696</td>
</tr>
<tr>
<td>ML cnn-cnnflow-spect</td>
<td>0.626</td>
<td>0.689</td>
</tr>
<tr>
<td>PT cnn-cnnflow-spect</td>
<td><strong>0.667</strong></td>
<td><strong>0.734</strong></td>
</tr>
</tbody>
</table>

8.3.4 Improving Video Descriptions with Grenander’s Structure

Figure 8.5 shows how often labels of certain actions and objects are fixed in the interpretations due to the semantic consistency imposed by Grenander’s structures. The gray bars indicate how often labels are missed by Grenander’s structures but correctly retrieved by the method without priors. The graph on the top shows that the most likely actions to be corrected by Grenander’s structure-based methods are *stir, pour, take* and *open*. This also dictates what object labels are most likely to be correctly selected to build the interpretations, namely, *bowl, fridge, measuring cup* and *brownie box*. Note that these objects are semantically compatible with the most likely actions to be often correctly selected by the Grenander’s structure methods. For example, interpretations likely to be proposed with combination of these actions and objects include *open fridge, stir bowl, pour oil into measuring cup, take brownie box, open brownie box*, etc. The graph at the bottom, in Figure 8.5, shows that in fact these labels are the most likely labels to be corrected by methods using Grenander’s structures. On the other hand, other object labels more likely to be corrected by methods without the structural influence, for instance, *cap* and *egg.*
In a nutshell, the graphs in Figure 8.5 show that the methods based on Grenander’s structure are more likely to generate semantically consistent interpretations than the methods based solely on feature support. Figure 8.7 illustrates three interpretation cases depicted at different rows. On the first row, the correct interpretation is generated by the method based on Grenander’s structures because of the structural connections (bonds) between the action and object labels, namely, crack → egg and bowl → egg; thus changing the interpretation from putting egg in a brownie box to cracking egg in a bowl. Another good example of semantic consistency is illustrated in the second row of Figure 8.7, where the interpretation is changed from open bowl (which even by common sense may not be semantically coherent) to open brownie box.

8.4 Summary and Discussion

In this chapter, we demonstrated how to seamlessly account for the multimodality of the data source in a unified way using the proposed pattern theory-based video description model. Other observations are the following: the experiments show that the predictive power of CNN features were improved by considering the structural semantic dependencies of events encoded in terms of Grenander’s structures (generators, bonds and configurations). These structures carry complimentary data that encourage rectification of erroneously highly confident prediction by deep neural network classifiers of actions and objects. Auditory features were verified to be a potentially sufficient source of data for modeling actions. The semantic interpretations generated by the method built on auditory features for actions and CNN features for objects, i.e. PT cnn-spect, were qualitatively comparable to the ones generated by its counterparts that model actions with CNN features. This indicates that we could potentially reduce the feature pre-processing computational cost by skipping the motion analysis step. Finally, we verified that even when
using features not as discriminative as CNN features, Grenander’s structures can be sufficiently strong to achieve performance rates comparable to when using CNN features-based models alone.
Figure 8.5: Histogram of correct labels using CNN, CNNFLOW and spectrogram features. Number of action (first row) and object (second row) labels that were correctly selected to build the best interpretations by the Grenander’s structures-based methods and were missed by the methods (black bars). The gray bars show the opposite statistics.
Figure 8.6: Histogram of correct labels using CNN and spectrogram features. Number of action (first row) and object (second row) labels that were correctly selected to build the best interpretations by the Grenander’s structures-based methods and were missed by the methods (green bars). The gold bars show the opposite statistics.
Figure 8.7: Comparing interpretations based on deep features. This comparison is between a version with and without Grenander’s structures. Comparative illustration of video interpretations generated by the method based on deep models without structural information (second column) and the method with deep models using Grenander’s structures (third column). For each case (each row), the interpretations were corrected by the method that uses Grenander’s structures.
CHAPTER 9 : CONCLUSIONS

In this work, we introduced a novel framework for generating compact descriptions of video content. This framework underlying formalism is grounded on the mathematics of general pattern theory. We have demonstrated that by modeling the problem of video content description using pattern theory, in particular for interpretation of human activities, offers several advantages: i) it allows us to easily integrate and perform inference based on multiple modalities of data and other computer vision/machine learning algorithms, ii) prior knowledge with different levels of formality rigor can be easily leveraged by the framework using the constructs of pattern theory, iii) it can infer highly complex structures by learning from much simpler units of information, which incurs in needing a lot less training data then it is typically needed by competitive approaches, iv) it is capable to represent a very large space of solutions with a compact representation model. This latter, in some cases, may incur in an expected increase of the inference computation cost but that can be alleviated by implementing sophisticated inference algorithms. We have studied and experimented the robustness and performance of this framework in several challenging, contemporary datasets of human activities. Challenging scenarios included scenes containing clutter of objects, occurrence of simultaneous events and longer sequences of consecutive activities.
REFERENCES


APPENDIX A: COPYRIGHT CLEARANCE FORMS

Find below the permissions for chapters 3, 4 and 5.

Thesis / Dissertation Reuse
The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of (university/educational entity’s name goes here)’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
INTRODUCTION
Find below the permission for chapter 6.
Find below the permission for chapter 7.

**Thesis / Dissertation Reuse**

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.

2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [Year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]

2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.

3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
ABOUT THE AUTHOR

Fillipe Souza was born and raised in Alagoinhas-BA, a small town in Brazil. He received his Bachelors of Science in Computer Science from the Universidade Estadual de Santa Cruz (UESC), in 2009, located in Ilheus, BA Brazil. It was during his time as an undergraduate student of Computer Science that he was exposed to scientific research by Prof. Adriano Hoth, when he worked as a research assistant at the Laboratorio de Astrofisica Teorica e Observacional da UESC. Following that, he received a Master’s degree in Computer Science from the Universidade Federal de Minas Gerais (UFMG), Brazil in 2011, when he was advised by Prof. Arnaldo de Albuquerque, Prof. Eduardo Valle and Prof. Guillermo Camara-Chavez. During that time, he initiated his studies on subjects of Computer Vision and Machine Learning. In September of 2010, he had the opportunity to present his first published work on computer vision in a national graphics conference knowns as SIBGRAPI. In the conference, he was impressed by the keynote speaker Dr. Sudeep Sarkar and became interested in pursuing his PhD degree with him. In August 2011, he moved to Tampa, FL to pursue a doctoral degree in Computer Science & Engineering under advisory of Prof. Sudeep Sarkar and Prof. Anuj Srivastava. Fillipe was the Entrepreneurial Lead for NSF Innovation Corps grant (USD $50,000) which was awarded to validate the commercialization opportunity of the video technology developed in his PhD work. He also served as the President for IEEE Computer Society USF student chapter from 2016 to 2017.