University of California
Los Angeles

Learning and Inferring Perceptual Causality from Video

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

by

Amy Sue Fire

2016

Abstract of the Dissertation

Learning and Inferring Perceptual Causality from Video

by

Amy Sue Fire

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2016

Professor Song-Chun Zhu, Chair

In the physical world, cause and effect are inseparable: ambient conditions trigger humans to perform actions, thereby driving status changes of objects in the scene. Perceptual causality is the perception of causal relationships from observation. Humans, even as infants, form such models from observation of the world around them [Saxe and Carey, 2006]. For a deeper understanding, the computer must make similar models through the analogous form of observation: video.

In this dissertation, we provide a framework for the unsupervised learning of this perceptual causal structure from video. Our method takes action and object status detections as input and uses heuristics suggested by cognitive science research to produce the causal links perceived between them. We greedily modify an initial distribution featuring independence between potential causes and effects by adding dependencies that maximize information gain.

We compile the learned causal relationships into a Causal And-Or Graph, a probabilistic and-or representation of causality that adds a prior to causality.

Validated against human perception, experiments show that our method correctly learns causal relations, attributing status changes of objects to causing actions amid irrelevant actions. Our method outperforms Hellinger's χ^2-statistic

by considering hierarchical action selection, and outperforms the treatment effect by discounting coincidental relationships.

In video, triggering conditions, causing actions, and effects may be hidden due to ambiguity, occlusion, or because they are otherwise unobservable, but humans still perceive them. We build a probability model for a sequential Causal And-Or Graph to represent actions and their effects on objects over time. For inference, we apply a Viterbi algorithm, grounded on probabilistic detections from video, that fills in hidden and misdetected actions and statuses. Our results demonstrate the effectiveness of reasoning with causality over time.

The dissertation of Amy Sue Fire is approved.

Hongjing Lu

Yingnian Wu

Mark Stephen Handcock

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2016

To my family.

Table of Contents

1 Introduction **1**

1.1 Motivation: Why Study Vision and Causality 2

1.2 Our Theory 4

1.2.1 Overview of the Dissertation 6

2 Literature Review **7**

2.1 Causality 7

2.2 Computer Vision 9

2.2.1 Causality in Computer Vision 10

2.2.2 Learning in Computer Vision 11

2.3 Artificial Intelligence and Commonsense Reasoning 11

2.4 Perceptual Causality in Cognitive Science 12

2.5 How Our Work Differs 13

3 Actions, Fluents, and the Causal And-Or Graph **16**

3.1 Converting Perceptual Causality to Heuristics 16

3.2 Fluents: Time-varying states of objects 17

3.3 The Causal And-Or Graph to Represent Causality 18

4 Learning the Causal And-Or Graph **23**

4.1 Setting Up the Learning Problem 23

4.1.1 Assumptions and Structural Equation Models 23

4.1.2 Potential Effects: The Space of Fluent Changes 24

4.1.3 Potential Causes: The Space of Action Detections 25

4.2 Perceptual Causal Relations 25
4.2.1 Defining Perceptual Causal Relations 25
4.2.2 Preparing the Data: Creating Clips from the Video 26
4.2.3 Evaluating Causal Relations 27
4.3 Pursuit of the Causal Relations 28
4.3.1 Fitting the Causal Relation 30
4.3.2 Pursuing Causal Relations by Information Projection . . . 32
4.3.3 Selection of **cr** When Actions are Hierarchical 33
4.4 The Learned Causal And-Or Graph 36
4.5 Experiments 38
4.5.1 Toy Example: Simulated Vending Machine 38
4.5.2 The Office: Learning Amid Confusing Actions 40
4.5.3 Locked Door Data: Hierarchical Action Selection 45
4.5.4 Elevator Data: Delayed Effects 47
4.5.5 Reasoning in Surprising Circumstances 48
4.6 Summary and Discussion 49

5 Inferring on the Causal And-Or Graph 51

5.1 Inference of a Single Parse Graph: The Energies 51
5.2 Reasoning over Time 53
5.2.1 Consistency of Transitions between Parse Graphs 53
5.2.2 Non-Markovian Duration 53
5.2.3 Inference of the Sequential Parse Graphs 54
5.3 Experiment 1: Causal Grammar vs. Detections 57
5.3.1 Baseline: Bottom-Up Fluent and Action Detection 57

5.3.2 Baseline: Random Noise 60
5.3.3 Human Annotation 60
5.3.4 Protocol for Experiment Evaluation 61
5.3.5 Results 61
5.4 Experiment 2: Variability of Humans 65
5.4.1 Human Annotation 66
5.4.2 Baseline Estimate (Random Noise). 66
5.4.3 Computer Estimate (The Causal And-Or Graph). 66
5.4.4 Results and Discussion 67
5.5 Deriving the Viterbi Algorithm 69
5.6 Discussion and summary 71

6 Discussion and Future Work 72

List of Figures

1.1 An example of causal inference. 2

1.2 Fluent examples. 3

1.3 Causal connections. 5

2.1 Pre- and post-intervention distributions. 8

3.1 A Causal And-Or Graph for an office at time t. 19

3.2 A Causal And-Or Graph and a parse graph. 21

4.1 Bar charts of relative frequencies. 28

4.2 Causal knowledge as causal networks. 29

4.3 Node selection in a hierarchy. 35

4.4 Office Causal And-Or Graph. 36

4.5 Simulated Spatio-Temporal-Causal And-Or Graph. 38

4.6 Simulation results. 39

4.7 Office data information gains. 41

4.8 Noisy data information gains. 43

4.9 ROC curves for joint pursuit of the door and light. 44

4.10 ROC curves for the monitor. 45

4.11 ROC curves for joint pursuit of door, light, and monitor. 46

4.12 ROC curve using N randomly selected examples. 46

4.13 Hierarchical example: the locked door. 47

4.14 Confounded example: the elevator. 48

5.1 Inconsistent state transition. 54

5.2 Fluent durations. 55
5.3 Hidden semi-Markov model 55
5.4 Fluent detection. 59
5.5 Human poses and depth images. 60
5.6 Microwave results. 62
5.7 Results for the screen and light example. 63
5.8 Sample of human judgment key frames. 66
5.9 Correcting spatio-temporal detections. 67
5.10 MDS plots of fluent value estimates. 68
5.11 Hidden semi-Markov model with completion nodes 69

List of Tables

4.1 Causal relation. 26

4.2 Relative Frequencies. 28

4.3 Legend of actions for office scene. 40

4.4 Information gains for top causal relations. 41

5.1 Dataset Included Action/Fluent Relationships 58

5.2 Hit rates for actions and fluents. 64

5.3 List of fluents considered. 65

Acknowledgments

First and foremost, I would like to thank my advisor, Song-Chun Zhu, for the opportunity to work in computer vision. He has taught me many things about academia, but perhaps my treasured lesson is to keep on trying; failure is something everyone encounters, and it is important to pick yourself up and try again. I thank him for his support and guidance, as well as his encouragement and patience while I juggled motherhood with being a student.

I also thank the other members of my committee: Judea Pearl (who had to leave), Mark Handcock, Yingnian Wu, and Hongjing Lu (who stepped in for Judea).

I thank Glenda Jones and Jason Mesa for their help navigating the UCLA bureaucracy.

I appreciate having the opportunity to work with my lab mates at UCLA (in alphabetical order): Zhi Han, Wenze Hu, Jungseock Joo, Wei Liang, Bruce (Xiaohan) Nie, Seyoung Park, Mingtao Pei, Brandon Rothrock, Xiaoqiong (Phoebe) Su, Dan Xie, Zhenyu (Benjamin) Yao, Ping Wei, Tianfu Wu, Yibiao Zhao, Mingtian Zhao, and Yixin Zhu. Their help with my research, their friendships, and their insights have been invaluable.

I thank Judea Pearl and Elias Bareinboim for discussions on causality. As we came to the conversation from different sides, I found their perspective particularly valuable.

This work has been supported by the Office of Naval Research, under MURI grant N00014-10-1-0933.

This dissertation is based on the following publications, Song-Chun Zhu as PI:

A. Fire and S.-C. Zhu. Using causal induction in humans to learn and infer causality from video. In *The Annual Meeting of the Cognitive Science*

Society (CogSci), 2013.

A. Fire and S.-C. Zhu. Learning perceptual causality from video. *ACM Trans. Intell. Syst. Technol.*, 7(2):23:1–23:22, 2016.

A. Fire and S.-C. Zhu. Inferring hidden statuses and actions in video by causal reasoning. In submission, *Advances in Neural Information Processing Systems*, 2016.

Finally, I thank my family for their support. I thank them for their countless babysitting hours, errand-running, and encouragement.

Vita

2002	B.A. (Mathematics), University of California, Berkeley
2004	M.S. (Mathematics), San Francisco State University
2004–2011	Associate Professor, Mathematics, College of the Canyons.
2009–2011	Teaching Assistant. Department of Statistics, University of California, Los Angeles
2011-2013	Graduate Student Researcher. Department of Statistics, University of California, Los Angeles
2013	C.Phil. (Statistics), University of California, Los Angeles
2011-2014	Senior Vision Consultant, Blindsight Corporation, Berkeley

Publications

A. Fire and S.-C. Zhu. Learning perceptual causality from video. In *AAAI Workshop: Learning Rich Representations from Low-Level Sensors*, 2013.

A. Fire and S.-C. Zhu. Using causal induction in humans to learn and infer causality from video. In *The Annual Meeting of the Cognitive Science Society (CogSci)*, 2013.

A. Fire and S.-C. Zhu. Learning perceptual causality from video. *ACM Trans.*

Intell. Syst. Technol., 7(2):23:1–23:22, 2016.

CHAPTER 1

Introduction

Humans perceive causes and effects as they navigate the world. This perception of causality leads to deep-rooted expectations, for example, that hitting a light switch will turn a light on or off. Humans form these relationships from infancy [Saxe and Carey, 2006], and cognitive scientists believe that this knowledge is acquired by observation [Griffiths and Tenenbaum, 2005].

Consider the images in Figure 1.1. A man raises a phone to his head. He moves the mouse and starts typing. He grabs a cup, moves to the water dispenser, and brings the cup to his head. He moves to the trash can and bends down. He moves to the wall and raises his arm; the light goes out.

Connecting triggering conditions to actions to effects, Figure 1.1 shows an inference possible by long-term reasoning. Seeing a man raise a phone to his head, we can infer he's talking to someone on the phone, perhaps because it rang. The man moved the mouse to wake the monitor, his thirst motivated him to fill the cup and drink, and he threw something away. Knowing that the monitor is actively displaying, for example, is imperative to being able to label the action "sitting in front of the computer" as "using the computer". Without seeing the person flipping a light switch (the switch is not detectable), we still reason that he performed that action based on the observed effect. By the end of the event, we might infer that the monitor is inactive.

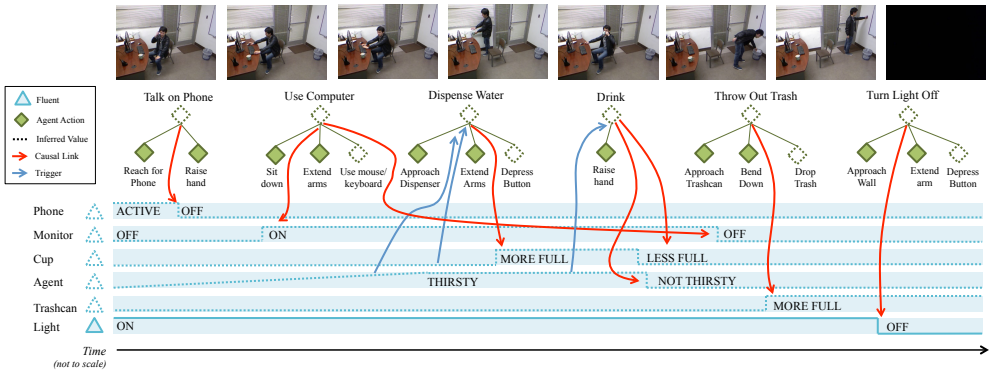


Figure 1.1: An example of causal inference. Over time, observed actions are used to infer values of hidden fluents, and values of observed fluents are similarly used to infer hidden actions.

1.1 Motivation: Why Study Vision and Causality

The observations from the office scene in Figure 1.1 illustrate the power of the causal connection between actions and fluents. A *fluent* specifically defines those statuses of objects and humans that change value over time [Mueller, 2006]. A light fluent, for example, takes the values "on" and "off" over time as the light switch is flipped.

Further examples of fluents are shown in Figure 1.2. In these examples, object fluents are connected to actions as preconditions or triggers (e.g., an empty cup gets filled by a thirsty person) or as effects (e.g., using the mouse or keyboard turns the monitor on). Because of limitations on visibility and detectability, the values of these fluents are often hidden (e.g., the fill-level of a cup).

Changes in fluent value may be caused by human action (e.g., a light turns on when a person flips the switch) or by an internal mechanism (e.g., a screensaver activates on a monitor). Non-changes are explained by inaction (e.g., a light that is on stays on until it's turned off) or by maintaining action (e.g., continued computer use keeps the monitor awake). Actions can be detectable (e.g., using a computer) or hard to detect (e.g., making a phone call). Some actions are even

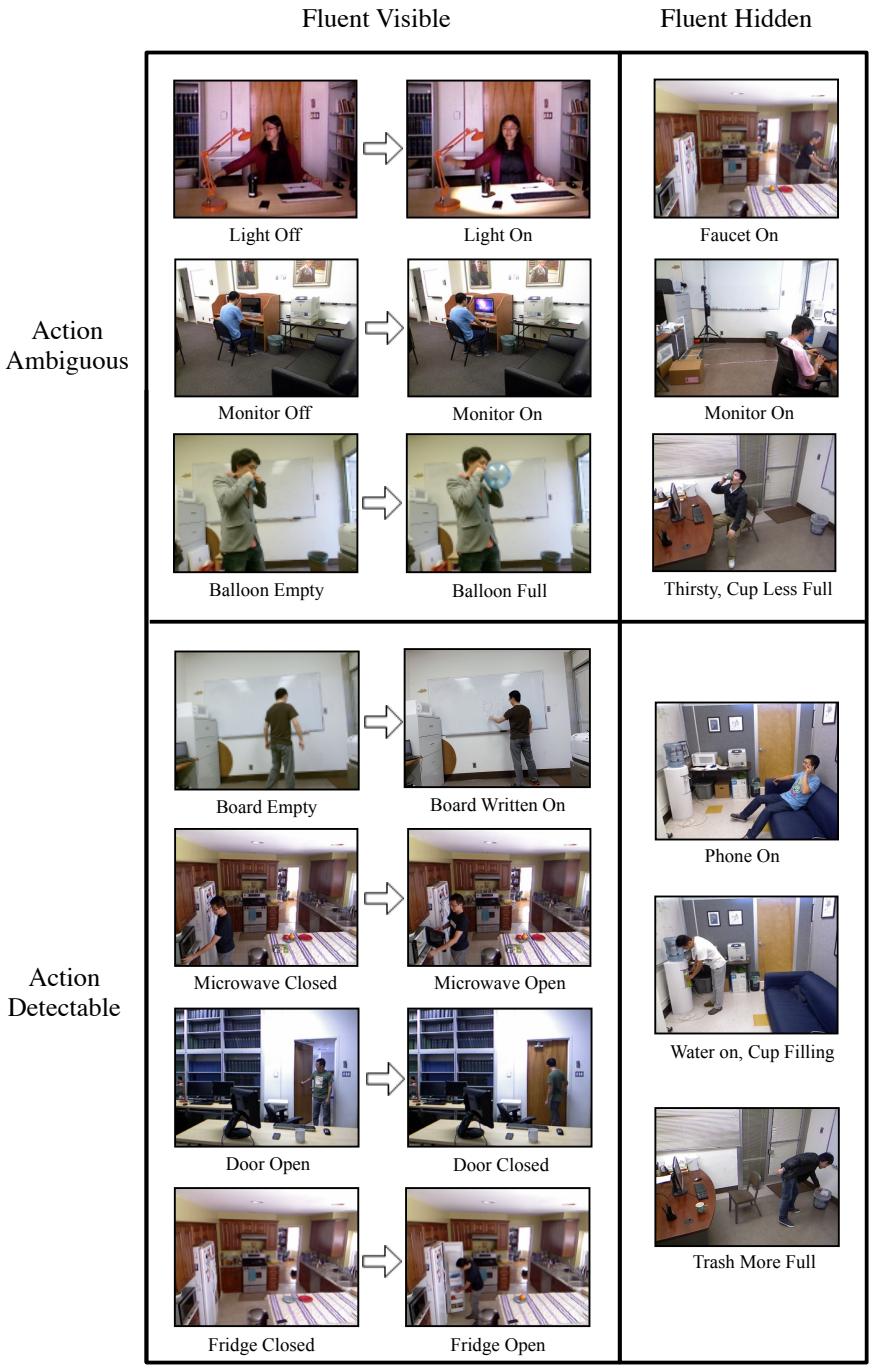


Figure 1.2: Fluent examples. Fluents are time-varying properties of objects and may be visible or hidden; they change as a result of causing actions. Some actions may be easily detectable, while others are ambiguous. Under the context of causal relationships between actions and fluents, detections improve.

defined by their causal effects: a "blowing" action is not detectable, but can be reasoned from the expanding balloon.

Interpreting actions as goal-driven, humans perform actions to cause changes in fluents, motivated by triggering conditions [Csibra and Gergely, 2007]. In this dissertation, we use causality to jointly infer actions and fluents from video, even when they are unobservable. Connecting short-term causal knowledge consistently over the course of a video enables reasoning about actions and fluents over time as preconditions, causes, or effects, thereby improving detection (what is in the scene and where it is located) and moving toward higher-level cognition to answer the questions of "why" and "how".

1.2 Our Theory

This dissertation focuses on perceived causal relationships between actions and fluents in two ways: learning these relationships through observation and inferring instances of these relationships over time.

The primary analog of observation for the computer comes from video. In this dissertation, we propose a framework for the unsupervised learning of perceptual causal structure (specifically those causal relationships that are perceived) from video taken in everyday scenes such as an office, a doorway, and an elevator [Fire and Zhu, 2013b, 2016]. We search for causes of fluent changes, learning causal relationships in the world, as illustrated by the dashed arrows in Figure 1.3.

As input, our method takes potentially noisy action and fluent detections from video. We use an information projection pursuit to incrementally learn perceptual causal structure in an unsupervised way, drawing causal links between actions and fluents. We derive analytic solutions for obtaining and fitting the causal relations, and we show that this method selects the most relevant action from an action hierarchy.

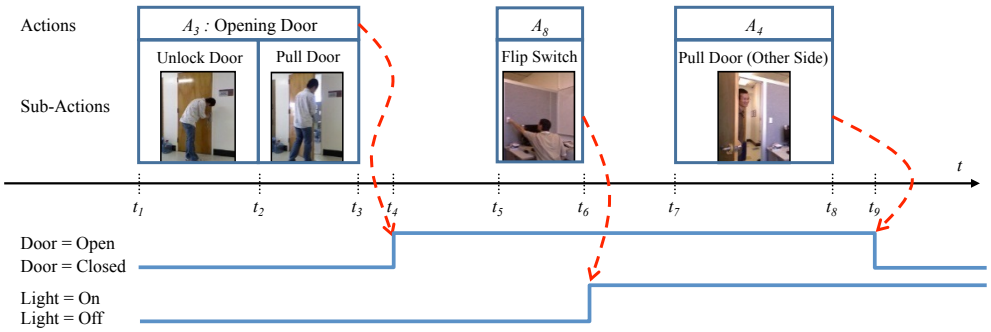


Figure 1.3: Causal connections. Key actions over time are shown at the top, and status changes below. Dashed arrows indicate perceived causal links. A link can be found between any action in an action hierarchy and its relevant fluent change. The joint actions of unlocking and pulling cause the door to open at t_4. From t_5 to t_6, a person turns a light on. From t_7 to t_8, a person pulls the door from the other side, resulting in a closed door at t_9.

These learned causal links are assembled into a Causal And-Or Graph and the learned probability model is used to encode prior information on causality.

In experiments, we study how much temporal lag to allow between actions and their fluent effects, finding it best to control both the number of actions considered as causes as well as the absolute time between the causing action and its fluent effect. We further explore the number of training examples needed. We compare our results against Hellinger's χ^2 and the treatment effect, finding that our method performs best. Finally, we examine other factors that affect the perception of causality from video: incorrect detections and confounding actions.

We develop a probability model for the C-AOG [Fire and Zhu, 2013b] that integrates with real detections. We extend the C-AOG to a sequential model, allowing long-term inference of both actions and fluents from video data, connecting triggering fluents to actions to their effects. We present a Viterbi algorithm to fill in hidden fluents and actions and correct misdetections.

1.2.1 Overview of the Dissertation

This rest of this dissertation is organized as follows: In Chapter 2, we present a literature review to ground our multidisciplinary work. In Chapter 3, we explain fluents and introduce the Causal And-Or Graph. In Chapter 4, we present our learning theory [Fire and Zhu, 2016]. In Chapter 5, we extend the Causal And-Or Graph over time and show how it can be used to improve detections.

CHAPTER 2

Literature Review

Extracting causal relationships from video is multi-disciplinary, combining current vision detection systems along with artificial and human thought (artificial intelligence and cognitive science). We begin by introducing causality.

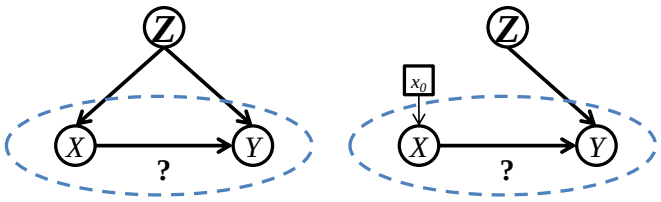
2.1 Causality

Causality has been studied extensively in social statistics where researchers have investigated, for example, whether smoking causes cancer, asking counterfactual questions such as "What would have been the cancer level for this person had the person not smoked?" [Rubin, 2007]. Other variables such as age might confound the causal effect, where age influences both preference to smoke and chance of getting cancer.

Pearl provides tools for the analysis of counterfactuals: the causal diagram provides a non-parametric graphical model for causality, and the *do*-notation separates causal quantities from statistical ones [Pearl, 2009].

Consider the causal diagram in Figure 2.1(a). Suppose X gives smoking status where $X \in \{\text{smoker, nonsmoker}\}$, and X causes Y where Y is lung cancer status, $Y \in \{\text{cancer, no cancer}\}$. Further, $\mathbf{Z}$ represents all potential confounding factors, such as age or locale, that could affect X and Y. Unlike a Bayesian network where arrows merely capture dependencies, these arrows point from causes to effects.

Observation, e.g., where there is no assignment of smoking status, allows mea-



(a) Pre-Intervention (b) Post-Intervention

Figure 2.1: Pre- and post-intervention distributions. The pre-intervention distribution in (a) gives the causal structure of the variables. Intervening to force X to take value x_0 breaks the dependence of X on $\mathbf{Z}$, yielding the post-intervention network in (b).

surement of the statistical relationship between X and Y, $P(Y = y|X = x_0)$. From the network in Figure 2.1(a),

$$P(y|x_0) = \sum_{\mathbf{z}} \left[P(y|x_0, \mathbf{z})P(\mathbf{z}|x_0)\right]. \tag{2.1}$$

Both the graph and the equation (after applying Bayes rule) show that X depends on $\mathbf{Z}$. The relationship $P(y|x_0)$ is influenced by the mechanism that causes a person to select a smoking status (here, age).

To isolate the effect of smoking on cancer, the connection between smoking and age must be broken by intervening on the mechanism that decides smoking status, thereby forcing a given smoking status. When X is forced through intervention to take a particular value, x_0, all causal links to X are removed. This produces the post-intervention network, which renders X independent of $\mathbf{Z}$, as shown in the post-intervention distribution of Figure 2.1(b). Pearl introduced notation for this intervention, $do(X = x_0)$. Further, Pearl showed it is possible to estimate post intervention quantities from the pre-intervention distribution:

$$P(y|do(x_0)) = \sum_{\mathbf{z}} P(y|x_0, \mathbf{z})P(\mathbf{z}), \tag{2.2}$$

where the dependence between X and $\mathbf{Z}$ is now broken.

Potential causes are ranked based on their causal effect. One such measure is the treatment effect, denoted by TE, of treatment x_0 over x_0':

$$TE = E(Y|do(X = x_0)) - E(Y|do(X = x_0')), \tag{2.3}$$

the difference in expected values of Y from setting X to x_0 versus x_0'. The larger $|TE|$, the stronger the causal effect.

In order to bring the context of causality to vision research, we first review current work.

2.2 Computer Vision

Researchers in computer vision focus on detecting objects, understanding scenes, and recognizing actions and events. Contextual information is increasingly sought to improve recognition performance by exploiting non-accidental relationships in space and time.

Image parsing uses the spatial context between objects and backgrounds for object recognition and scene categorization [Tu et al., 2005, Hoiem et al., 2007]. Combining multiple frames, video parsing exploits temporal context to recognize actions by using hidden Markov models [Brand et al., 1997], dynamic Bayesian networks [Al-Hames and Rigoll, 2005], logic formulas [Brendel et al., 2011, Albanese et al., 2010] and stochastic grammar models (both context-free [Ryoo and Aggarwal, 2006, Ivanov and Bobick, 2000] and context-sensitive [Pei et al., 2011]). Even though HMMs and DBNs also perform event recognition, grammar models are reconfigurable and accommodate high-level structures, both of which are needed for reasoning over the time-varying detections of actions and fluents [Pei et al., 2011]. Many of these models for actions are built atop temporal logic formulas [Allen and Ferguson, 1994].

Recent vision research has found even greater progress through joint inference

over other types of context. Recognition rates improve for small objects when taken in the context of human actions [Gupta et al., 2009a, Yao and Fei-Fei, 2010] and for pedestrians when taken in the context of the scene [Saberian et al., 2014]. Context also allows inference of the intangible, such as potential uses of objects as tools [Zhu et al., 2015], "dark matter" [Xie et al., 2013], and of forces applied to objects in human-object interaction [Pham et al., 2015].

2.2.1 Causality in Computer Vision

The context of causality is studied in computer vision research in a limited way. Using causality, many of the event recognition works based on logic representations infer actions without propagating the information to effects or over time (e.g., CASE^E [Hakeem et al., 2004]). In the storyline model [Gupta et al., 2009b], linguistic annotations of video are used to learn which actions precede other actions, such as a pitch in baseball preceding a hit. Causality has also been used in the spatial domain to aid segmentation [Taylor et al., 2015].

Vision works using formal causal models are infrequently found, with some early works using Newtonian mechanics to distinguish actions [Mann et al., 1997].

Recently, vision researchers have used causal measures such as Granger causality to learn similar patterns of repeated low-level actions, allowing the unsupervised identification of hand shaking sequences and child hand games such as pat-a-cake [Prabhakar et al., 2010].

The works studying causality in vision listed above fall into two groups: some utilize causal relationships for recognition and others use causal measures to learn similar patterns. None of these approaches formally study cause-and-effect relationships in a way that allows causal structure to be learned from video.

Further, current action datasets largely ignore cause and effect relationships, focusing instead on human motion [Kuehne et al., 2011], human interactions [Ryoo

and Aggarwal, 2010], or complex activities [Niebles et al., 2010].

Advancing in the direction of cognitive science and perceptual causality, Brand borrows from infants' perceived implications of motion to provide the "gist" of a video using detected blobs [Brand, 1997]. One of the main drawbacks to this work, however, is that the grammar is not learned.

2.2.2 Learning in Computer Vision

The works using causality listed in Section 2.2.1 do not learn their causal relationships. While researchers are making progress in the unsupervised learning of actions in video [Si et al., 2011, Brendel and Todorovic, 2011], these works are devoid of causal relationships.

2.3 Artificial Intelligence and Commonsense Reasoning

Learning causality in artificial intelligence, on the other hand, usually amounts to traditional causal induction as done by constraint-based algorithms such as IC [Pearl, 2009], PC, and FCI [Spirtes et al., 2000], or by Bayesian formulations that place a prior on graph structure [Heckerman, 1995]. These methods are intractable to ground on vision sensors, and the methods used in computer vision are far from learning this kind of causal structure. Even using these systems atop mid-level visual words is computationally infeasible when considering the vast domain of observable causal relations.

While Bayesian networks are commonly used to represent causality [Pearl, 2009], reconfigurations within a grammar model represent a greater breadth of possibilities than a single instance of a Bayesian network with pre-defined structure [Griffiths and Tenenbaum, 2007], making it more suitable for vision applications. The And-Or Graph graphically embodies grammar models and has been used for objects, scenes, and actions [Zhu and Mumford, 2006]. Even though HMMs and

DBNs also perform event recognition [Brand et al., 1997, Al-Hames and Rigoll, 2005], grammar models are reconfigurable and accommodate high-level structure, both of which are needed for reasoning over time-varying detections of actions and fluents.

Further, artificial intelligence research strives to generate the causal conclusions as would be drawn under well-designed experiments. However, this notion of causality does not necessarily align with human perceptions.

Researchers in commonsense reasoning usually apply first-order logic to causal reasoning tasks [Mueller, 2006]. Learning these models is disjoint from vision sensors and features. Solution methods follow constraint satisfaction techniques and deduction, and tend not to admit probabilistic solutions, which are crucial in vision to allow for ambiguity of unreliable detections.

Markov logic networks [Richardson and Domingos, 2006] relax the strictness of first-order logic by wrapping them in a Markov random field, but while they have been applied to the task of action detection [Tran and Davis, 2008], the knowledge base was not learned. The network structure in Markov logic networks is pre-defined (not reconfigurable) and slow inference hinders their widespread use. As Brendel et al. point out [2011], these networks are not suitable for vision due to tractability issues.

2.4 Perceptual Causality in Cognitive Science

As humans observe their world, they form conclusions about causal relationships, linking states of the world to perceived causing conditions. Causal connections are so strong in humans that they can even override spatial perceptions [Scholl and Nakayama, 2004]. Cognitive scientists recognize that even infants are equipped with a notion of *perceptual causality*, able to draw causal conclusions from observations based on temporal spacing and an innate understanding of agency [Carey,

2009]. It is this type of causality that is learnable from video.

Perceptual causality as studied by cognitive scientists fills in the gaps that make traditional causal discovery methods insufficient for computer vision tasks. The artificial-intelligence methods for causal induction leave many questions: they do not inform which detection variables humans would indicate as causes or effects (from pixels over time, to features aggregating pixels, to object classifiers using features, to action detectors, to hierarchies thereof); they do not indicate how to divide the video stream to create examples (too small a clip might omit causes; too large of one introduces noise); and they do not encode a prior understanding of likely causes that could be used in detections. Cognitive science research of infants provides the following answers.

Humans link a change in an object status with the action of an agent [Saxe et al., 2005]. Humans award the "cause" distinction to the agent's action of opening the door (decomposed at a high level into unlocking and pulling open the door), ahead of individual pixels, the door, and the lock (part of the door). Philosophers agree—humans use a simplified causal model when answering questions. When asked what caused a light to turn on, humans will identify the agent's action alone—ignoring all the other necessary conditions for the effect such as working electrical power and the switch being connected to the light [Mackie, 1965].

Humans consider cause and effect relationships when the temporal lag between the two is short and cause precedes effect [Schlottmann and Shanks, 1992]. Finally, humans learn perceptual causality through daily observation by internally measuring co-occurrence of events and effects [Griffiths and Tenenbaum, 2005].

2.5 How Our Work Differs

In order to bring causality and vision together, we use perceptual causality.

The theories for learning and inferring causality that have been developed in

artificial intelligence are insufficient for the task of learning from video. Constraint satisfaction algorithms do not represent perceptual causality, and while Bayesian formulations have been used in cognitive science [Griffiths and Tenenbaum, 2005], they have not been grounded on action detections from video. Even though perceptual causality lacks the accuracy of causal induction, it still provides valuable—and more human—information.

The learning process incrementally builds a probability model, and then the acquired causal knowledge is used for reasoning. Observation for a computer comes through video, and to begin learning perceptual causality, the computer must examine co-occurrence, similarly restricted. Beginning with a vision system that detects fluents and actions from video, our method learns perceptual causality from video in an unsupervised manner, attributing an action as the cause of a fluent change. Further, by using the same measure for co-occurrence to learn perceptual causality as used for learning objects and actions from low-level sensors (information projection), we provide a principled approach to learning, integrating the spatial, temporal, and causal domains.

We limit ourselves to agentive actions as potential causes of fluent changes. We construct examples from the video that only consider actions occurring within a small window preceding a given effect. We measure co-occurrence between actions and fluents.

While Bayesian networks are commonly used to represent causality [Pearl, 2009], reconfigurations within a grammar model represent a greater breadth of possibilities than a single instance of a Bayesian network with pre-defined structure [Griffiths and Tenenbaum, 2007], making it more suitable for vision applications. The And-Or Graph (AOG) [Pearl, 1984] graphically embodies grammar models and is used for object, scene, and temporal representation [Zhu and Mumford, 2006, Pei et al., 2011]. In Section 3.3, we adapt the And-Or Graph to represent causality, providing a representation for causality that grounds on pixels through

its consistency with current spatio- and temporal-models.

CHAPTER 3

Actions, Fluents, and the Causal And-Or Graph

In this chapter, we convert perceptual causality to heuristics, introduce the two types of fluents studied here, and introduce a causal grammar model for them.

3.1 Converting Perceptual Causality to Heuristics

Perceptual causality offers solutions for causal discovery from video, and distinguishes itself from the causal induction typically done. We now summarize the ideas behind perceptual causality presented in Section 2.4 and convert them to heuristics.

Heuristic 1 Agentive actions are causes,

$$\text{Action} \rightarrow \text{Effect}.$$

This heuristic informs the set of potential causes: It's not the pixels we see or the human that we detect, but it's the human *doing something* that causes a fluent to change.

Heuristic 2 The temporal lag between cause and effect is short, with cause preceding effect,

$$0 < \text{Time(Effect)} - \text{Time(Causing Action)} < \delta.$$

This provides a method for breaking the video stream into clips to create examples. Determining δ is challenging: taking it too small

might exclude the cause, and taking it too large creates too much noise. We examine various temporal lags, as well as different ways of measuring the temporal lag, in Experiment 4.5.2.3.

Heuristic 3 Strength of a perceptual causal relationship is obtained by measuring the co-occurrence between actions and effects.

In learning causal relations, we examine co-occurrence while simultaneously building our model following an information projection pursuit. In experiments, we find our method outperforms Hellinger's χ^2 measure for co-occurrence and the treatment effect.

In this paper, we restrict causal relations to be specifically between an agent's action and a fluent change, where the action precedes the fluent change within some small time window. When the computer examines the co-occurrence of Heuristic 3, restricted by Heuristic 1 and Heuristic 2, then we assume the model determined represents perceptual causality: linking a fluent to its causing action.

It is important to briefly note that perceptual causality is a loose form of causality: at times, a human will perceive incorrect causal relations. By using the heuristics, we run the same risk of error. Despite this, the heuristics provide a useful model connecting vision and causality.

3.2 Fluents: Time-varying states of objects

In contrast to constant attributes such as gender or color [Farhadi et al., 2009], the concept of fluents was introduced to stress the time-varying properties of objects, including the continuous position and velocity in Newtonian mechanics [Newton, 1736] and discrete states in event calculus [Mueller, 2006]. We examine two types of fluents:

Object fluents, e.g., whether a monitor is on, or a cup has water. Object

fluents are connected to actions as preconditions (an empty cup gets filled by a thirsty person) or as effects (using the mouse or keyboard turns the monitor on, and filling or drinking from the cup changes its fill-level). Because of limitations on visibility and detectability, the values of these fluents are often hidden.

Human fluents, e.g., whether a person is thirsty. The basic state of a human triggers that person's actions. These fluents are never directly observable.

Changes in fluent value are caused by human action (e.g., the light turns on when a person flips the switch) or may be spontaneous due to an internal state change (e.g., a screensaver activates on a monitor, or a person becomes thirsty over time). Non-changes are explained by non-action (e.g., a light that is on stays on until it's turned off) or by a maintaining action (e.g., continued computer use keeps the monitor awake).

3.3 The Causal And-Or Graph to Represent Causality

The Causal And-Or Graph adds a causal layer on And-Or Graph representations for objects and actions, identifying human actions as causes for fluent changes and providing a stochastic grammar representation of perceptual causality [Fire and Zhu, 2013b, 2016]. The And-Or Graph naturally lends itself to represent actions as causes for fluent changes: And-nodes group sub-actions (e.g., the sub-actions used to detect "use keyboard"), while Or-nodes represent the alternative causes (e.g., a monitor can be woken by someone using a mouse *or* a keyboard). Examples are shown in Figures 3.1 and 3.2.

Since the hierarchical structure of the And-Or Graph has been used to represent spatial knowledge in image parsing [Zhu and Mumford, 2006] and temporal knowledge in event parsing [Pei et al., 2011], using the And-Or Graph to represent causal information provides a uniform spatial-temporal-causal representation that adds another layer of hierarchy atop spatial and temporal grammar mod-

els, grounding the Causal And-Or Graph on raw sensors. Allowing for multiple configurations and high-level structures, the hierarchical structure of the And-Or Graph gives maximum flexibility in selecting potential causing actions.

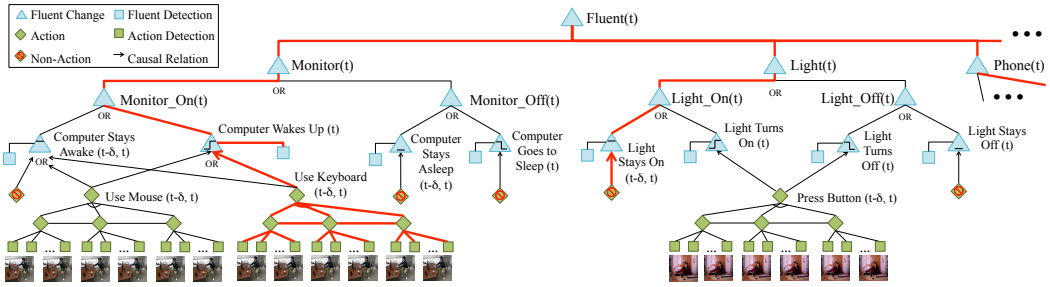


Figure 3.1: A Causal And-Or Graph for an office at time t. Fluent values are consequences of their children. Arcs connect children of And-nodes. A single selection at the Or-nodes (red, bold lines here) provides a parse graph, explaining the current instance of time. Terminal leaf nodes ground the Causal And-Or Graph on video, linking input from detected features. Step functions indicate types of fluent changes: step up for turning "on", step down for "off".

The Causal And-Or Graph is comprised of the following parts:

Or-nodes. Or-nodes represent fluent values, whose children are the alternate causes for that fluent value (e.g., a monitor can be woken by someone using a mouse or a keyboard).

For example, the door fluent value of closed, shown in Figure 3.2 as an Or-node, could be caused by any of the alternative causes:

$$\text{Door is Closed} \leftarrow \text{Non-Action} \vee \text{Push Door} \vee \text{Pull Door}. \tag{3.1}$$

These Or-nodes represent a choice in the causing condition. Here, actions cause fluent values to change. Similarly, non-actions maintain a fluent's value.

And-nodes. And-nodes group sub-actions, conditions, and relations thereof for the cause (e.g., the sub-actions used to detect "use keyboard").

Action recognition works by detecting spatio-temporal relationships in the video (e.g., detecting computer use through relative positions of skeleton joints and proximity to the computer [Wei et al., 2013]). These spatio-temporal relationships are really compositions of fluents (as ambient conditions or as the visual decomposition of actions). In the Causal And-Or Graph, these compositions are represented with And-nodes, e.g.,

$$\text{Open Door with Key} \triangleq \text{Unlock Door} \wedge \text{Pull Door} \tag{3.2}$$

where $\triangleq$ represents definition.

Leaf nodes. Terminal leaf nodes at the lowest level represent features for detecting fluent changes and actions in video by bottom-up methods, such as GentleBoost for fluent changes and SVM for actions. These nodes connect the Causal And-Or Graph to the video at the pixel level.

Temporal Relations. Links connect nodes with temporal relationships (e.g., a person nears the computer before using it).

Arrows. Arrows point from causes to effects.

The causality describing these fluent changes is nearly instantaneous—pushing a button immediately turns a light on; moving a mouse immediately wakes a monitor. Given a short video sequence $V[t-\delta, t]$, the Causal And-Or Graph represents causal explanations for fluents at time t where causing actions occur within a δ time window (e.g., modeling that using the keyboard causes the monitor to display and the light remains on at t, as shown with thick red in Figure 3.1).

Considering single actions alone is not enough. Actions come hierarchically defined, where, for example, the person opening the door performs the actions unlock and pull. The method we present for learning can correctly select from a hierarchy, as shown in Section 4.3.3.

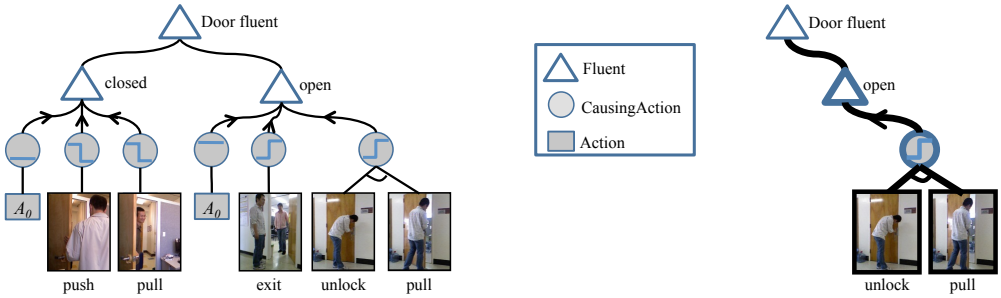


Figure 3.2: The Causal And-Or Graph (left) and a parse graph (right). Each causing action node shows an action from a high level of the hierarchy. Arrows point from these actions (causes) to the fluent (effect). Children of And-nodes are grouped by arcs. A_0 represents non-action, causing a fluent to maintain status.

A parse graph (pg) from the Causal And-Or Graph is formed by making a selection at the Or-nodes (e.g., the thick red lines in Figure 3.1, or the left side of Figure 3.2) and captures the causal reason that the fluent changed value at t (causes indicated with arrows). A parse graph provides a causal explanation for the video clip. For example, the parse graph in Figure 3.2 shows that the door is open because an agent unlocked and pulled.

The best parse graph at t is given by selecting the best children per

$$P(pg_t|V[t-\delta,t]) \propto P(pg_t;\Theta) \prod_{l \in L(pg_t)} P(l|pg_t) \tag{3.3}$$

where $L(pg)$ is the set of included terminal leaf nodes, including both actions and fluents. This posterior (explained below) is a product of the prior defined over the Causal And-Or Graph (with parameter vector Θ) and the likelihood of all leaf nodes for fluent and action detectors.

Further, the Or-nodes encode *prior* information on the different causes. Humans have an intuitive understanding of causation that they use to answer questions amid missing or hidden information. Without seeing what happened or knowing what the circumstances are in the room, they can answer: Why is the

door closed? (Because no one opened it.) Why did the light turn on? (Because someone toggled the switch.) A prior on causality is important for computer vision as it enables guesses on the particular causal relationship (both the cause and effect together) in play when only partial information is available and thus can fill in detections.

We present theory to learn the Causal And-Or Graph in Chapter 4 and how to infer instances in Chapter 5.

CHAPTER 4

Learning the Causal And-Or Graph

While event parsing describes what happens in a video, it does not generally explain *why*. In event parsing, changes in fluents occur independently of actions, leaving causal information unassigned. Augmenting event parsing with causal semantics, this chapter presents methods to learn and model the causal connections from actions to fluents, $A \rightarrow \Delta F$.

This chapter is structured as follows. We set up the hardware to learn causal relations in Section 4.1. We develop theory to sequentially learn which actions cause which fluents to change in Sections 4.2-4.3 [Fire and Zhu, 2016]. In Section 4.4, we assemble the pursued causal relations into a Causal And-Or Graph. In Section 4.5, experiments validate the learning framework.

4.1 Setting Up the Learning Problem

In this section, we write our assumptions, and we define the sets of fluents and actions.

4.1.1 Assumptions and Structural Equation Models

In addition to Heuristics 1-3, we also make some assumptions standard to traditional causal discovery.

We assume that our detections (and the hierarchies used for such) are sufficient. In particular, the set of pre-specified actions is sufficient, and the computer is able

to generally detect these elements in the scene when they occur. We assume that there are no confounders.

We assume causal faithfulness: multiple causes do not exactly cancel. When we detect no correlation, we match this to the perception of no causal connection.

We assume each effect is a function of its immediate causes and an independent error. Each action, A_i, depends on its own exogenous variable, u_{A_i}. Using ΔF_j to denote fluent change j, we notate in terms of structural equations:

$$A_i = g_{A_i}(u_{A_i}) \text{ for } i = 1, \ldots, n_A \tag{4.1}$$

$$\Delta F_j = g_{\Delta F_j}(\mathbf{A}_j, u_{\Delta F_j}) \text{ for } j = 1, \ldots, n_{\Delta F} \tag{4.2}$$

where $\mathbf{A}_j$ denotes specifically those actions that are in a causal relationship with ΔF_j. $u_{\Delta F_j}$ are exogenous.

4.1.2 Potential Effects: The Space of Fluent Changes

Given a fluent, F, that can take n_F values, there are $n_{\Delta F} = n_F^2$ possible transitions from time t to $t+1$. With the door, for example, where the fluent could be "open" or "closed", there are four possible sequences: the door changes from "open" to "closed", changes from "closed" to "open", remains "open", or remains "closed". We notate the fluent change for a clip with ΔF.

Per the commonsense reasoning literature [Mueller, 2006], a lack of change-inducing action (referred to here as a non-action) causes the fluent to maintain its status, denoted $\Delta F = 0$; for example, a door that is closed will remain closed until some action changes that status. Figure 1.3 showed the door and the light maintaining their statuses for varied durations, punctuated by periods of change due to action.

The space of possible fluent changes for an object in the video is pre-specified and denoted by

$$\Omega_{\Delta F} = \{\Delta F\}.$$

4.1.3 Potential Causes: The Space of Action Detections

Action parsing provides Ω_A, the space of actions. Ω_A contains actions at high levels of an action hierarchy. An action detection hierarchy (e.g., [Pei et al., 2011]) aggregates pixels into objects, relates these objects spatially and temporally to define atomic actions, groups those into sub-actions (such as pushing or pulling the door), and hierarchically combines them even further (for example, unlocking and pulling the door). Figure 1.3 showed actions from different levels of the hierarchy.

Ω_A is limited to top-level actions or sub-actions from a pre-designed action hierarchy. Following Heuristic 1, these agentive actions form the potential causes.

In order to make detections, these sets of actions and fluents must be pre-specified so appropriate detectors can be trained. These definitions influence the final Causal And-Or Graph learned.

4.2 Perceptual Causal Relations

In this section, we formalize our key building block for causal structure: the notion of a perceptual causal relation between an action and a fluent change.

4.2.1 Defining Perceptual Causal Relations

Combining the fluent changes with the actions, we define the space of potential causal relations as a Cartesian product that pairs an action with a fluent change.

Definition 1 (Space of Causal Relations)**.** *The space of causal relations is given by*

$$\Omega_{CR} = \Omega_A \times \Omega_{\Delta F}. \tag{4.3}$$

The space, Ω_{CR}, provides the basic units for learning. Elements $\mathbf{cr} \in \Omega_{CR}$ specify an action and fluent change, and provide the framework for the 2×2

tables shown in Table 4.1.

Table 4.1: Causal relation.

cr :		¬Action	Action
	¬Effect	c_0	c_1
	Effect	c_2	c_3

Labeling the individual cells of the table, $\mathbf{cr} = (c_0, c_1, c_2, c_3)$ where c_i functions as a binary indicator. When applied to a short video clip, the elements of Ω_{CR} identify whether or not the clip has the action and/or fluent change.

When a collection of these video clips shows strong evidence for $\mathbf{cr} \in \Omega_{CR}$, we award perceptual causal status and add the element to our model.

4.2.2 Preparing the Data: Creating Clips from the Video

We evaluate the elements from Ω_{CR} using video. A long video sequence, $\mathbf{V}$, is first decomposed into shorter video clips, $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. Following Heuristic 2 for limiting temporal lag, only actions occurring within a pre-specified $\delta_{\max}$ of the fluent change are included in $\mathbf{v}_i$, to be considered as potential causes. The function $d(t_A, t_F)$ measures time between the action completion, t_A, and the fluent change, t_F. Some ways to compute $d(t_A, t_F)$ considered in this chapter include:

1. Counting the number of frames between t_A and t_F. In experiments, we consider $\delta_{\max}$ between 15 and 90 seconds.

2. Counting the number of action detections between t_A and t_F. In experiments, we consider $\delta_{\max}$ ranging from 1 to 6 recent actions.

3. Combinations of the first two. For example, taking the maximum of 15 seconds and 2 actions ensures clips last *at least* 15 seconds long and with *at least* 2 action detections. Taking the minimum of 15 seconds and 2 actions

creates clips of *at most* 15 seconds or 2 action detections. In experiments, we consider $\delta_{\max}$ to be the maximum or minimum over combinations of $15, 45$ seconds and $1, 2, 3$ actions.

These are explored in experiments in Section 4.5.2.3. It is intuitive to expect a dependence between clip length definition and performance. If the clip is not long enough to include the causing action, then the ability to detect causes diminishes. However, if clip length is too long, the noise hides the causal relations.

4.2.3 Evaluating Causal Relations

Tallying the values from $\mathbf{cr} \in \Omega_{CR}$ across the clips, $\mathbf{v}_i$, we obtain relative frequencies for the particular action and fluent change:

Definition 2 (Relative Frequencies of a Causal Relation)**.** *Given a causal relation* $\mathbf{cr}$ *and video* $\mathbf{V}$ *that has been broken into clips* $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$*, the relative frequencies of* $\mathbf{cr}$ *are given by*

$$RF(\mathbf{cr}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{cr}(\mathbf{v}_i). \tag{4.4}$$

The relative frequencies from the video's action and fluent detections are denoted by $\mathbf{f} = (f_0, f_1, f_2, f_3)$ and represent the percentages that a video clip exhibited both a given action and/or fluent change.

Our causal model is built by greedily augmenting the action and fluent distributions with causal relations, linking actions to fluent changes. At any iteration, there is the model that has been built so far (the "current model"), and the observed data from the video. The limiting relative frequencies under the current model are denoted by $\mathbf{h} = (h_0, h_1, h_2, h_3)$. Table 4.2 summarizes these statistics.

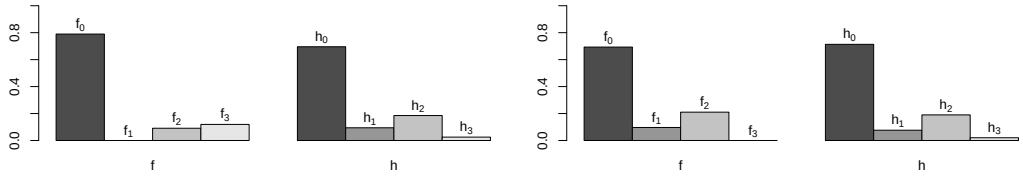
We construct our model by electing the most informative causal relations sequentially in terms of maximizing the information gain. Intuitively, this information gain is linked to the difference between $\mathbf{f}$ and $\mathbf{h}$.

Table 4.2: Relative Frequencies.

ΔF	A	cr	Current Model	Observed Data
0	0	$\mathbf{cr}_0$	h_0	f_0
0	1	$\mathbf{cr}_1$	h_1	f_1
1	0	$\mathbf{cr}_2$	h_2	f_2
1	1	$\mathbf{cr}_3$	h_3	f_3

For a causing action, **f** is shown in Figure 4.1(a), together with the relative frequencies of **cr** under a probability model assuming independence, **h**. The greatest difference between these histograms occurs in the f_1/h_1 and f_3/h_3 components. The relative frequencies **f** and **h** for a non-causing action in (b) look equivalent, indicating independence between the fluent and action.

We select the relations that show the greatest difference between **f** and **h**, as measured by the KL-divergence, thereby adding perceptual causal semantics to the model.



(a) Causing Action (b) Non-Causing Action

Figure 4.1: Bar charts of relative frequencies. Relative frequencies of **cr** for the observations are shown on the left of each pair, and for the model of independence on the right.

4.3 Pursuit of the Causal Relations

In this section we develop the theoretical framework for our learning theory. From the space of all possible relations, Ω_{CR}, we now show how to sequentially select

cr and incorporate it into the joint model.

We assume the video clips, $\mathbf{v}_i$, are drawn from an unknown distribution of perceptual causality, $f(\mathbf{v})$. We incrementally build a series of models approximating f,

$$p_0(\mathbf{v}) \to p_1(\mathbf{v}) \to \ldots \to p(\mathbf{v}) \to p_+(\mathbf{v}) \to \ldots \to p_k(\mathbf{v}) \approx f(\mathbf{v}), \tag{4.5}$$

where each new model incorporates a new causal relation as illustrated in Figure 4.2. We use an information projection approach (see, e.g., [Csiszár and Shields, 2004]).

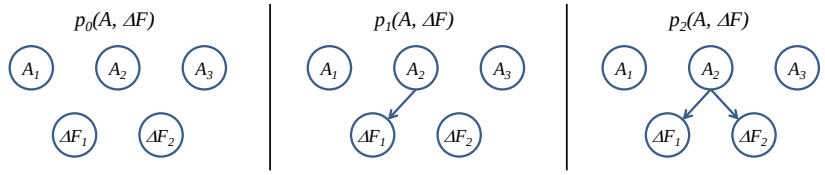


Figure 4.2: Causal knowledge as causal networks. The perceptual causal structure is incrementally constructed. Here, the action is flipping the light switch, which can turn the light on or off.

As shown in the first panel of Figure 4.2, learning initializes by independently considering action and fluent distributions, p_A and $p_{\Delta F}$, respectively:

$$p_0(\mathbf{v}) = p_A(\mathbf{v})p_{\Delta F}(\mathbf{v}). \tag{4.6}$$

In this dissertation, we initialize $p_A(\mathbf{v})$ with the proportion of clips, $\mathbf{v}$, that contain action A; similarly for $p_{\Delta F}(\mathbf{v})$.

In a single iteration, we fix the current model, p, and augment to a new model, p_+. Under the minimax information projection framework, learning proceeds in two steps. In the first step, we select which causal relation to add to the model by maximizing the information gain, the KL-divergence between p_+ and p. In step two, we minimize the KL-divergence between p_+ and p, subject to matching the causal relation to the observed data.

Any model over the video clips that considers fluent changes independently from causing actions, such as p_0, will fail to match f on causal relations. However, given a selected relation, step two requires that the new model match the observed data on the newly selected causal relation

$$E_{p_+}[\mathbf{cr}_+] = E_f[\mathbf{cr}_+] \approx \mathbf{f}. \tag{4.7}$$

The probability distribution with minimum KL-divergence, $\mathrm{KL}(p_+||p)$, subject to that constraint is

$$p_+(\mathbf{v}) = \frac{1}{z_+} p(\mathbf{v}) \exp\left(-\langle \lambda_+, \mathbf{cr}_+ \rangle(\mathbf{v})\right) \tag{4.8}$$

where $\lambda_+ = (\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ is a scalar vector corresponding to the components of $\mathbf{cr}_+(\mathbf{v}) = (c_0(\mathbf{v}), c_1(\mathbf{v}), c_2(\mathbf{v}), c_3(\mathbf{v}))$ shown in Table 4.1 and described in Section 4.2.1 and z_+ is a normalizing constant. When p_0 is uniform, Equation 4.8 yields the maximum entropy distribution.

4.3.1 Fitting the Causal Relation

Unlike other information projection applications to vision (e.g., [Della Pietra et al., 1997] [Zhu et al., 1997]), λ_+ can be computed analytically thanks to the binary nature of the causal relation.

Proposition 1. *To add the causal relation* $\mathbf{cr}_+$ *to the model in Equation 4.8, the parameters are given by:*

$$\lambda_i = \log\left[\frac{h_i}{h_0} \cdot \frac{f_0}{f_i}\right] \tag{4.9}$$

for $i = 0, \ldots, 3$*, where* h_i *and* f_i *are as found in Table 4.2.*

Proof of Prop. 1. Consider adding a single causal relation to the probability distribution, $p(\mathbf{v}) = \frac{1}{Z} exp(-\mathcal{E}(\mathbf{v}))$. This gives a new probability distribution

$$p_+(\mathbf{v}) = \frac{1}{z_+} p(\mathbf{v}) \exp\left(-\left\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \right\rangle\right). \tag{4.10}$$

Since $\sum_{i=0}^{3} c_i = 1$, there is 1 degree of freedom in λ_+; without loss of generality, set $\lambda_0 = 0$.

From the observed data, the expected value under the true distribution, f, is best estimated by the quantity from the data,

$$E_f(c_i(\mathbf{v})) = f_i. \tag{4.11}$$

Further, $E_p(c_i(\mathbf{v})) = h_i$.

$$\begin{aligned}
E_{p_+(\mathbf{v})}(c_i(\mathbf{v})) &= \int p_+(\mathbf{v}) c_i(\mathbf{v}) d\mathbf{v} && (4.12)\\
&= \int \frac{1}{z_+} p(\mathbf{v}) \exp(-\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) c_i(\mathbf{v}) d\mathbf{v} && (4.13)\\
&= E_p\left(\frac{1}{z_+} \exp(-\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) c_i(\mathbf{v}) \right) && (4.14)\\
&= \frac{1}{z_+} h_i \exp(-\lambda_i) && (4.15)
\end{aligned}$$

The last equation holds because the $c_i(\mathbf{v})$ are binary indicators and only one will be nonzero at a time.

Equating the matched statistics,

$$f_i = \frac{1}{z_+} h_i \exp(-\lambda_i). \tag{4.16}$$

Since $\lambda_0 = 0$, $f_0 = \frac{h_0}{z_+}$, or

$$z_+ = \frac{h_0}{f_0}. \tag{4.17}$$

Hence,

$$\lambda_i = \log\left[\frac{h_i}{h_0} \cdot \frac{f_0}{f_i} \right]. \tag{4.18}$$

□

Intuitively, the h_i/h_0 component "undoes" the independent consideration under the current model, and the f_0/f_i component inserts the new information joining the action and fluent change.

In experiments, $p_0(\mathbf{v})$ is defined over a finite set, and $\mathbf{h}$ is computable.

4.3.2 Pursuing Causal Relations by Information Projection

While Proposition 1 provides a formula to add a causal relation to a model, the best causal relation, $\mathbf{cr}_+$, is selected at each step through a greedy pursuit which leads to the maximum reduction of the KL divergence [Della Pietra et al., 1997], [Zhu et al., 1997]:

$$\mathbf{cr}_+ = \underset{\mathbf{cr}}{\operatorname{argmax}} \left(\mathrm{KL}(f||p) - \mathrm{KL}(f||p_+) \right). \tag{4.19}$$

Equivalently, $\mathbf{cr}_+$ is added to maximize the information gain:

$$\mathbf{cr}_+ = \underset{\mathbf{cr}}{\operatorname{argmax}}\, IG_+ \triangleq \underset{\mathbf{cr}}{\operatorname{argmax}}\, \mathrm{KL}(p_+||p) \geq 0, \tag{4.20}$$

moving the model closer to the true distribution f with each new causal relation.

An analytic formula provides the best causal relation:

Proposition 2. *The next best relation, $\mathbf{cr}_+$, to add to the model is given by*

$$\mathbf{cr}_+ = \underset{\mathbf{cr}}{\operatorname{argmax}}\, \mathrm{KL}(p_+||p) = \underset{\mathbf{cr}}{\operatorname{argmax}}\, \mathrm{KL}(\mathbf{f}||\mathbf{h}) \tag{4.21}$$

where $\mathbf{f}$ and $\mathbf{h}$ are as found in Section 4.2.3.

Proof of Prop. 2.

$$\begin{aligned}
\mathrm{KL}(p_+||p) &= \int p_+(\mathbf{v}) \log \frac{p_+(\mathbf{v})}{p(\mathbf{v})} d\mathbf{v} && (4.22)\\
&= \int p_+(\mathbf{v}) \log \left(\frac{1}{z_+} \exp(-\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) \right) d\mathbf{v} && (4.23)\\
&= \int p_+(\mathbf{v}) \log \frac{1}{z_+} d\mathbf{v} - \int p_+(\mathbf{v}) (\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) d\mathbf{v} && (4.24)\\
&= \log \frac{1}{z_+} - E_{p_+}(\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) && (4.25)\\
&= \log \frac{1}{z_+} - E_f(\langle \lambda_+, \mathbf{cr}_+(\mathbf{v}) \rangle) && (4.26)\\
&= \log \frac{1}{z_+} - \langle \lambda_+, \mathbf{f} \rangle. && (4.27)
\end{aligned}$$

Applying the formula for λ_i,

$$\lambda_i f_i = f_i \log \left[\frac{h_i}{h_0} \cdot \frac{f_0}{f_i} \right] \tag{4.28}$$

$$= f_i \log \frac{f_0}{h_0} + f_i \log \frac{h_i}{f_i}. \tag{4.29}$$

Continuing from Equation 4.27 and substituting Equations 4.17 and 4.29,

$$\mathrm{KL}(p_+||p) = \log \frac{f_0}{h_0} - \sum_{i=0}^{3} \left(f_i \log \frac{f_0}{h_0} + f_i \log \frac{h_i}{f_i} \right) \tag{4.30}$$

$$= (1 - f_1 - f_2 - f_3) \log \frac{f_0}{h_0} + \sum_{i=1}^{3} f_i \log \frac{f_i}{h_i} \tag{4.31}$$

$$= f_0 \log \frac{f_0}{h_0} + \sum_{i=1}^{3} f_i \log \frac{f_i}{h_i} \tag{4.32}$$

$$= \mathrm{KL}(\mathbf{f}||\mathbf{h}). \tag{4.33}$$

□

Therefore, in order to determine which causal relation is best to add to the model, we calculate the KL-divergence between the current model and the data for each potential causal relation, selecting the one that maximizes the information gain.

Once the relation $(A, \Delta F)$ is selected, perceptual causal arrows can be assigned, $A \rightarrow \Delta F$, attributing the fluent change to the action as proposed by Heuristic 1.

Algorithm 1 summarizes Propositions 1 and 2.

4.3.3 Selection of cr When Actions are Hierarchical

In recent computer vision literature, human actions are organized into hierarchical representations, such as a stochastic event grammar [Ivanov and Bobick, 2000] or the Temporal And-Or Graph [Pei et al., 2011]. In such representations, actions can be decomposed into sub-actions (where all parts compose the action) and

Algorithm 1: Learning the causal relations.

Input : Action and fluent change detections from a video, $d(t_A, t_F)$, and $\delta_{\max}$

Output : Probability distribution over a learned structure of perceptual causality

Create video clips according to $d()$ and $\delta_{\max}$;
Tally observations, giving $\mathbf{f}$;
Initialize model estimates, $\mathbf{h}$ (e.g., with proportions of action/fluent change occurrence);
repeat
    foreach *candidate causal relation* **do**
        Compute KL($\mathbf{f}||\mathbf{h}$) (Proposition 2);
    Select $\mathbf{cr}_+$ by selecting $\mathbf{cr}$ that maximizes the computed KL($\mathbf{f}||\mathbf{h}$);
    Calculate λ_+ by $\lambda_i = \log\left[\frac{h_i}{h_0} \cdot \frac{f_0}{f_i}\right]$ (Proposition 1);
    Update model estimates using λ_+;
until *information gain is smaller than a threshold*;

alternative actions. The Temporal And-Or Graph represents these as And-nodes and Or-nodes, respectively.

As instances of a parent and its children often compete, our learning method must have the precision to select the correct node as the cause of the fluent change. Fortunately, as the information gain for each action node in the action hierarchy is tested, these parent/child interactions are automatically taken into account. We examine single-cause situations below.

Selection on an Or-node. We first consider a parent action, A, that is a choice between two children, A_1 or A_2, as shown with the Or-node on the left of Figure 4.3. Intuitively, if A_1 is a cause, but not A_2, then A_1 will exhibit the strongest relationship with the fluent change. A will have the second highest, as some of the time it is activated when A_1 occurs and some of the time it is activated when A_2 occurs.

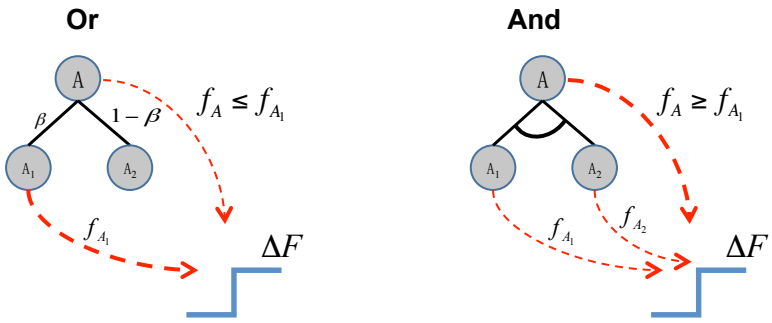


Figure 4.3: Node selection in a hierarchy. Graphical demonstration for when the algorithm encounters an Or-node or an And-node in the action hierarchy. When encountering an Or-node, where action A is identified through one of the child actions A_1 or A_2 with prior probability of A_1 of β, the pursuit process prefers the child node showing the strongest causal relation. For an And-node, where the action A is identified as a composition of A_1 and A_2, the parent is preferred.

For the case with a single causing action, A_1, the information gain is dominated by the $f_3 \log f_3/h_3$ contribution. Let f_A, f_{A_1}, and f_{A_2} be f_3 from Table 4.2 for A, A_1, and A_2, respectively. Further, let β be the Or-probability of selecting A_1. In this case,

$$\min(f_{A_1}, f_{A_2}) \leq f_A \leq \max(f_{A_1}, f_{A_2}). \tag{4.34}$$

Further, let h_A, h_{A_1}, and h_{A_2} be defined similarly. Since A happens if A_1 or A_2 happen, $h_A > h_{A_1}$.

Finally, if $h_3 < f_3$ as is the case on a distribution considering A and ΔF independently, then

$$h_{A_1} < h_A < f_A \leq f_{A_1}, \tag{4.35}$$

and the contribution on the information gain for A_1 will be larger than for A. In the case of an Or-node, the causing child node will be selected over the parent under pursuit by information gain.

Selection on an And-node. Next, let A be a parent that groups its children A_1 and A_2 as in the right side of Figure 4.3. In this case, A happens if both children

A_1 and A_2 happen and so $h_A < h_{A_1}$ and

$$f_A \geq f_{A_1}, f_{A_2} \tag{4.36}$$

and

$$h_A < h_{A_1} < f_{A_1} \leq f_A. \tag{4.37}$$

Therefore, for an And-node where both children must happen in order for the parent node to happen, our method selects the parent node.

4.4 The Learned Causal And-Or Graph

The learned causal relations are assembled into a Causal And-Or Graph, which serves to graphically represent the joint probability distribution learned in Section 4.3, conditioned on the fluent value. An example learned in experiments is shown in Figure 4.4. Causes are shown as children of fluents, with arrows indicating the direction of causality.

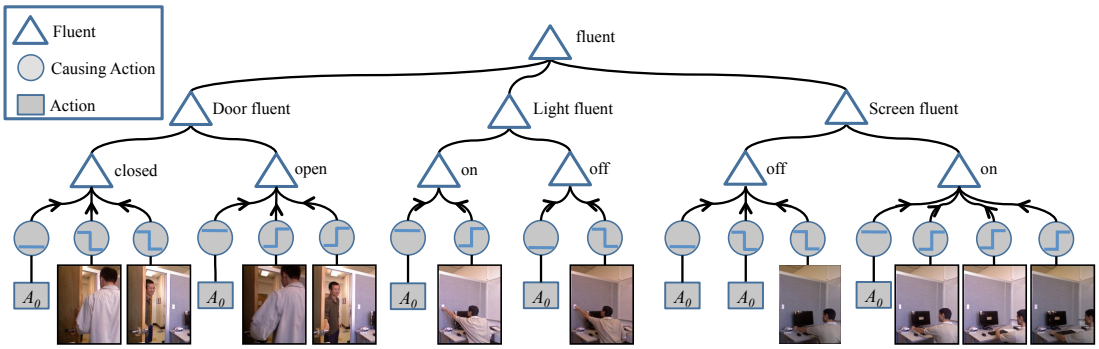


Figure 4.4: Office Causal And-Or Graph for door status, light status, and screen status. Action A_0 represents non-action (a lack of state-changing agent action). Non-action is also used to explain the change of the monitor status to off when the screensaver activates. Arrows point from causes to effects, and undirected lines show deterministic definition.

More concretely, probability is defined over the parse graphs, pg, in the Causal And-Or Graph, and is formed by conditioning on the fluent value in the jointly

pursued model:

$$p_{\mathrm{C}}(pg) = p(pg|F) \propto \exp\left(-\mathcal{E}_{\mathrm{C}}(pg)\right) \tag{4.38}$$

where

$$\mathcal{E}_{\mathrm{C}}(pg) \ = \mathcal{E}_0(pg) + \sum_{a \in CR(pg)} \lambda_a(w(a)). \tag{4.39}$$

$\mathcal{E}_0(pg)$ is the energy from the model p_0 in Equation 4.6, limited to the actions and fluents relevant to the included causal relations. $CR(pg)$ is the set of all non-empty, causal relations included in the parse graph (Or-nodes). $w(a)$ is the choice of causing action a (the selection of the child from the Or-node). λ_a comes from Equation 1 and represents the switch probability on the Or-nodes for $\mathbf{cr}_a$, providing a measure for how frequently an action causes the fluent status.

This prior on causality, $p_{\mathrm{C}}(pg)$, allows common knowledge to overcome ambiguous or missing action and fluent detections. When this prior distribution over the parse graphs is combined with a likelihood model, MAP inference provides instances of perceptual causality in video.

This probability on the Causal And-Or Graph can be thought of as a scoring mechanism for detection purposes. In particular, detections of fluents and actions contribute to the score, and the prior on causality contributes a favorable amount to the score if the actions and fluents detected are linked.

Note that the learned Causal And-Or Graph depends on both the pre-specified fluents of interest and the action recognition hierarchy used. For example, here we learned the joint actions of unlock and push open the door. This could more accurately be represented by changing the lock's fluent, coupled with the pushing action. Regardless, the learning method still produces a graph structure that is useful.

4.5 Experiments

In this section, we evaluate the learned perceptual causality relationships against human perception.

4.5.1 Toy Example: Simulated Vending Machine

To test the learning process amid incorrect action detections, we simulated a vending machine with the joint Spatio-Temporal-Causal And-Or Graph shown in Figure 4.5. An agent can use the vending machine or perform a confusing action. Using the vending machine correctly causes the machine to vend various confections. Some example sequences synthesized from the graph are:

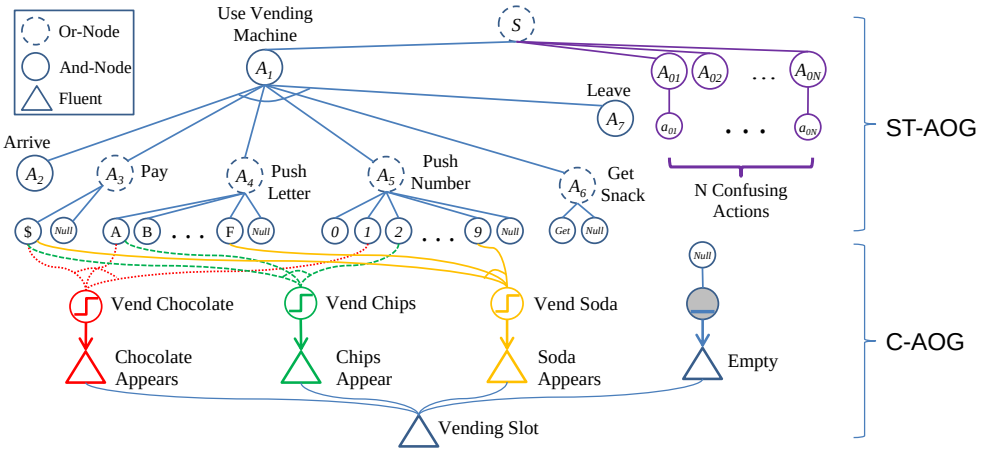
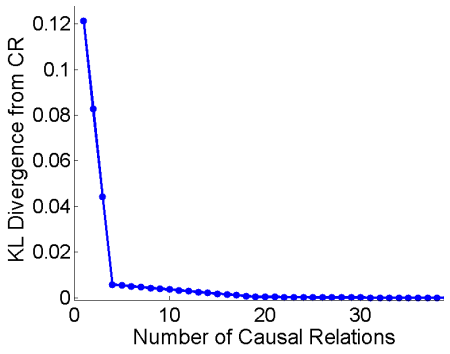


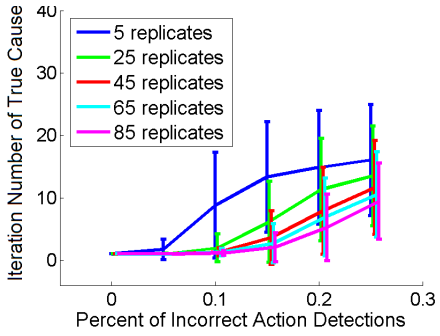
Figure 4.5: Simulated Spatio-Temporal-Causal And-Or Graph for vending machine. To use the vending machine, a code must be entered using an alphanumeric pad. With payment, the correct combination will cause the machine to vend one of three snacks: chips, chocolate, or soda. With an incorrect code or no payment, the vending slot remains empty.

1	Arrive, Push D, Push 1, Leave.
2	Arrive, Pay, Push A, Push 1, Get Snack, Leave.
	Machine Vends Chocolate.

Individual nodes, including 10 confusing actions and combinations thereof, are considered as potential causes for the machine to vend the various confections. The KL-divergence between the true data and the learned model that is attributable to causal relations is shown in Figure 4.6(a). After learning the true causal relations, the model learns noise, but these causal relations contribute minimally to the reduction in KL-divergence and are not generalizable.



(a) KL-divergence as causal relations are added to the model.



(b) Iteration in which true cause is selected when varying the number of misdetections.

Figure 4.6: Simulation results.

We randomly change a fraction ($p = 0, 0.05, 0.1, 0.15, 0.2, 0.25$) of simulated actions and fluents to provide noise that would occur with detection algorithms.

Possibilities from the And-Or Graph are sampled $N = 5, 25, 45, 65, 85$ times, creating replicates. The number of iterations to detect the true cause is calculated. Results are shown in Figure 4.6(b) where error bars are estimated using 500 different samples of each replicate. Under replication of the experiment design, our methods are able to overcome faulty action detection, ranking the true cause appropriately.

4.5.2 The Office: Learning Amid Confusing Actions

A video was recorded with a Kinect sensor in an office scene. Actions in the scene are listed in Table 4.3. Fluents include door open/closed, light on/off, and computer monitor on/off. The Causal And-Or Graph of Figure 4.4 shows some screenshots of the video. The video contains 8 to 20 (sometimes simultaneous) instances of each action category. There are a total of 66 possible action-fluent relations, with 10 true causal relationships among them.

Table 4.3: Legend of actions for office scene.

A_i	Description
A_0	Non-action, no explaining action
A_1	Open the door from the inside
A_2	Close the door from the inside
A_3	Open the door from the outside
A_4	Close the door from the outside
A_5	Touch the power button on the monitor
A_6	Touch the mouse
A_7	Touch the keyboard
A_8	Touch the light switch
A_9	Confusing action: pick something up
A_{10}	Confusing action: have a conversation
A_{11}	Confusing action: walk by

In this office scene experiment, we start with perfect action and fluent detections to demonstrate learning. We compare these results to those obtained with noisy detections.

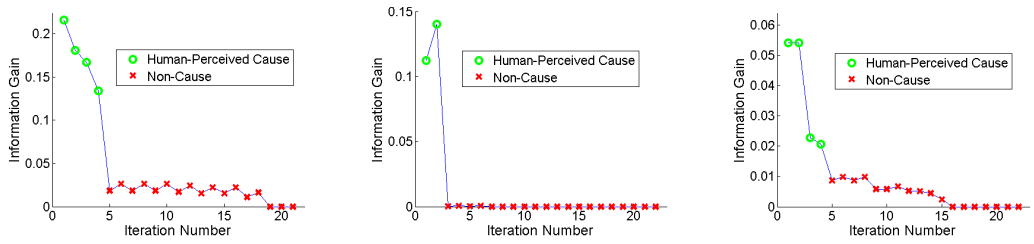
Table 4.4 shows information gains during the pursuit process for the door fluent. In the first 4 iterations, all four correct causal relations are selected. Once

the relation has been fit, the model does not gain information for that relation.

Table 4.4: Information gains for the top 13 causal relations involving the door fluent (columns) over 13 iterations (rows). The highest information gain in each iteration is shown bolded. True causes are shown with a gray background.

	$C{\to}O$ A_3	$O{\to}C$ A_4	$O{\to}C$ A_2	$C{\to}O$ A_1	$O{\to}C$ A_6	$C{\to}O$ A_6	$O{\to}C$ A_7	$C{\to}O$ A_7	$O{\to}C$ A_8	$C{\to}O$ A_8	$O{\to}C$ A_{10}	$C{\to}O$ A_{10}	$O{\to}C$ A_5
$k=1$	**0.2161**	0.1812	0.1668	0.1344	0.0185	0.0185	0.0185	0.0185	0.0185	0.0185	0.0170	0.0170	0.0155
$k=2$	0.0000	**0.1812**	0.1668	0.1344	0.0185	0.0185	0.0185	0.0185	0.0185	0.0185	0.0170	0.0170	0.0155
$k=3$	0.0000	0.0000	**0.1668**	0.1344	0.0185	0.0185	0.0185	0.0185	0.0185	0.0185	0.0170	0.0170	0.0155
$k=4$	0.0000	0.0000	0.0000	**0.1344**	0.0185	0.0185	0.0185	0.0185	0.0185	0.0185	0.0170	0.0170	0.0155
$k=5$	0.0000	0.0000	0.0000	0.0000	**0.0185**	**0.0185**	**0.0185**	**0.0185**	0.0185	0.0185	0.0170	0.0170	0.0155
$k=6$	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0264**	0.0185	0.0185	0.0185	0.0185	0.0170	0.0170	0.0155
$k=7$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0185**	**0.0185**	0.0185	0.0185	0.0170	0.0170	0.0155
$k=8$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0264**	0.0185	0.0185	0.0170	0.0170	0.0155
$k=9$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0185**	**0.0185**	0.0170	0.0170	0.0155
$k=10$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0264**	0.0170	0.0170	0.0155
$k=11$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0170**	**0.0170**	0.0155
$k=12$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0244**	0.0155
$k=13$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	**0.0155**

Figure 4.7 shows plots of information gains for causal relations in the order pursued, separated by fluent. Causes are added to the model before non-causes. Clear cutoffs of information gains for the door and light fluents separate causes from non-causes.



(a) Door (b) Light (c) Monitor

Figure 4.7: Office data information gains for causal relations in the order pursued, separated by fluent. Green circles label causes.

The correct cutoff is less clear for the computer monitor, in part due to only acquiring partial causal information. The monitor's display status has preconditions of power and computer status, which were not detectable.

4.5.2.1 Comparisons: Hellinger's χ^2 and TE

As learning causal structure is new to vision research, there are no benchmarks for comparison. Instead, we compare our learning technique to ranks of causal effects and to measurements of independence.

Potential causes can be ranked based on their causal effect. One such measure is the treatment effect, TE, of treatment A over $\neg A$, introduced in Section 2.1:

$$TE = E(\Delta F|do(A)) - E(\Delta F|do(\neg A)). \tag{4.40}$$

The larger $|TE|$ is, the stronger the causal effect.

As a further comparison tool, one standard measurement of independence is the χ^2 statistic. Due to low expected cell frequencies, the standard χ^2 measure is insufficient. Instead, we compare our results to Hellinger's χ^2, a more robust measure.

On this experiment, our results are validated with similarly ranked values of TE and χ^2.

4.5.2.2 Noisy Data

Randomly changing different percentages of action detections leads to the curves shown in Figure 4.8. For a 0.05 information gain threshold, the correct causal relations are picked up when 10% of actions and fluent instances are misdetected, but the causal relations are completely missed for 20% misdetections. As more noise enters the system, the information gained by considering causal relations decreases. While learning works amid noisy scenes (many actions happening simultaneously), clean detections are important.

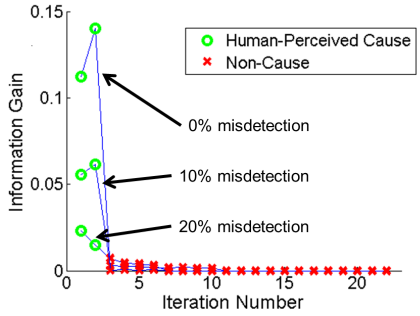


Figure 4.8: Noisy data information gains in the order pursued, separated by fluent.

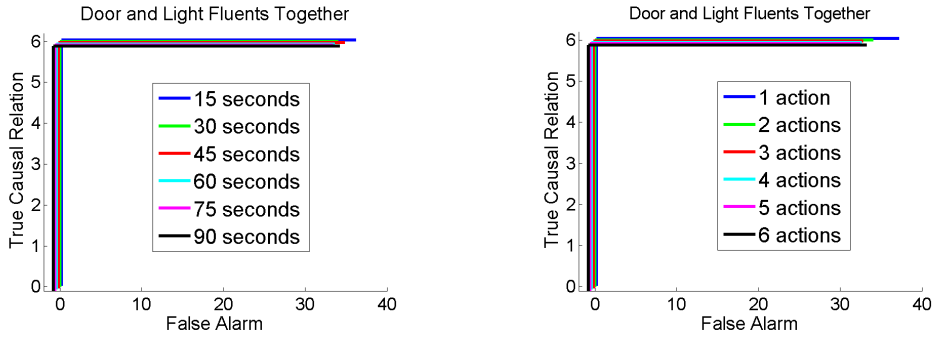
4.5.2.3 Varying Video Clip Length: The Effect of $d()$ and $\delta_{\max}$

This experiment explores the choice of $d()$ and $\delta_{\max}$ as described in Section 4.2.2, with simultaneous pursuit of door, light, and monitor causal relations. We measure $d()$ in three ways: as absolute time with a given number of frames, as an action count preceding the fluent, and as a combination of the two.

Example construction: limiting the number of actions or absolute time. Focusing on the clear causal relations of the door and the light, Figure 4.9 shows their causal relations are 100% detectable when constructing examples using a fixed number of seconds, shown in (a), or a fixed number of actions, shown in (b).

Because the monitor's power both confounds the causal effects of the keyboard and mouse, and is a cause itself, detecting all causal relations for the monitor is difficult, as shown in Figure 4.10. The learning process sees some examples where these actions lead to the fluent change and some where they do not, but there are no cues to differentiate between those cases. Lower TE and χ^2 reflect the confusion in detecting the causal status of the power button.

Jointly limiting the number of actions and absolute time. A video may show periods of clutter with many actions happening at once, whereas other times show no actions at all. We take $d()$ as a minimum (or maximum) over



(a) Seconds

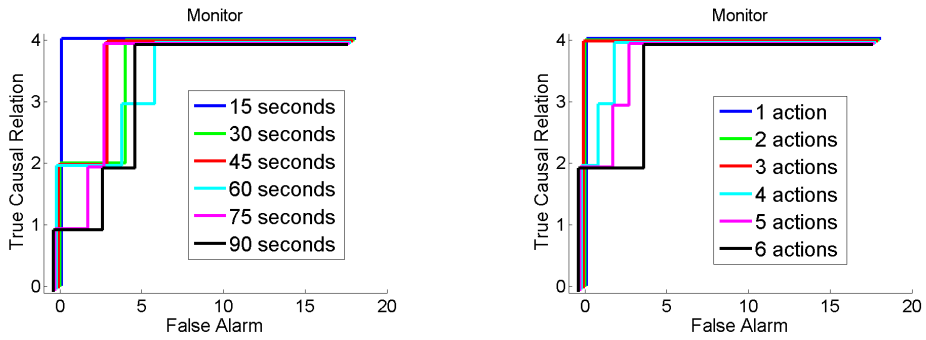
(b) Actions

Figure 4.9: ROC curves for joint matching pursuit of the door and light, restricting example size to a fixed number of (a) seconds and (b) actions. Six total ground-truth causal relations.

two methods for measuring time (counting seconds and counting actions). This ensures an example has a short duration if nothing is happening while simultaneously limiting the number of actions considered. Figure 4.11 highlights that the minimum outperforms the maximum.

The longer the time span used to build an example of the desired fluent, the more confusing actions enter as potentially relevant. Keeping the number of considered actions small makes the examples cleaner, decreasing noise obscuring the causal links. Optimally, the timespan used will be short, but special attention is required when considering events subject to a time delay.

A minor point on detection error: the detected cause may not be considered complete before the detection of the effect is begun. One way around this problem is to compare the start time of the cause against the end time of the effect, but this could have stronger implications on the temporal lag considered. As we showed, large lag obfuscates the causal relationships with so few examples.



(a) Seconds (b) Actions

Figure 4.10: ROC curves for joint matching pursuit of the monitor, restricting example size to a fixed number of (a) seconds and (b) actions. Four total ground-truth causal relations.

4.5.2.4 Number of Examples Needed to Perceive Causal Structure

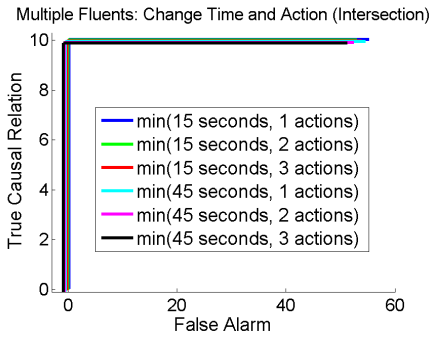
Reducing the number of examples used to learn causal relations has a detrimental effect on detection. Taking N random samples from the 97 examples, Figure 4.12(a) shows that as the number of examples used in training decreases, the ability to detect causal relations for the door fluent also decreases.

Figure 4.12(b) emphasizes the importance of quality, not quantity, in examples. While causes were recoverable in (a) with 5 examples, causes will never be recovered under the sample of 30 examples in (b). To identify a cause, there must be positive and negative examples.

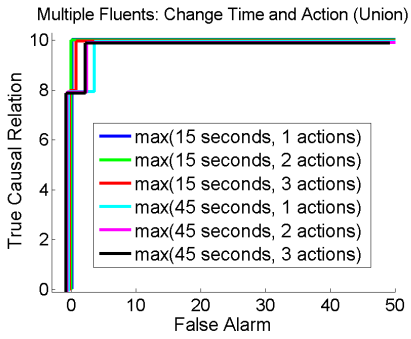
4.5.3 Locked Door Data: Hierarchical Action Selection

Where compound actions are required for the effect (e.g., in the doorway scene, unlocking with a key or entering a code, followed by pushing/pulling the door), the causing actions may come from any level of the action hierarchy. Figure 3.2 showed the learned Causal And-Or Graph for the doorway scenes.

Our method maintains dependencies for actions that occur together; actions

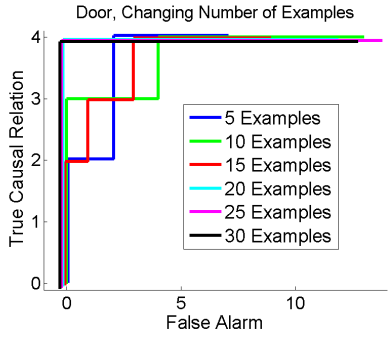


(a) Minimum

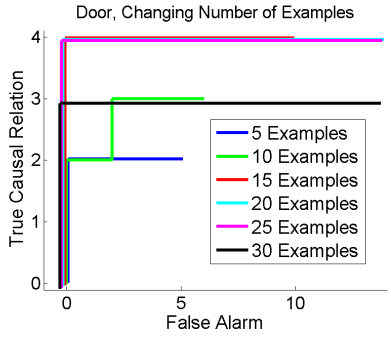


(b) Maximum

Figure 4.11: ROC curves for joint matching pursuit of door, light, and monitor, restricting example size to (a) the minimum between a fixed number of seconds and action and (b) the maximum. Ten total causal relations.

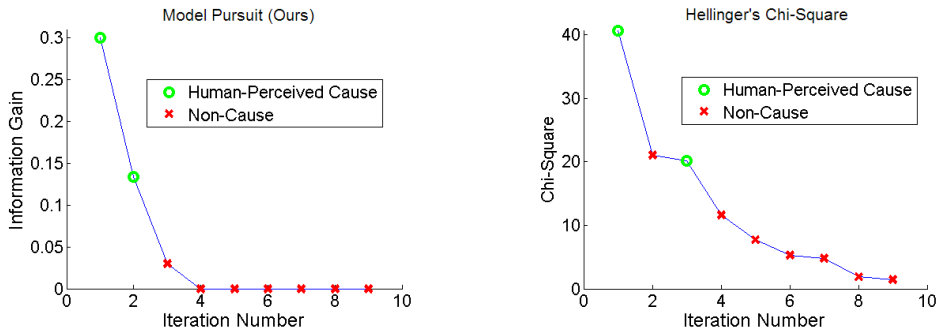


(a) "Good" samples



(b) "Bad" samples

Figure 4.12: ROC curve using N randomly selected examples to determine causal relations for the door fluent.



(a) Our Method

(b) Hellinger's χ^2

Figure 4.13: Hierarchical example: The locked door, pursuit order of causes.

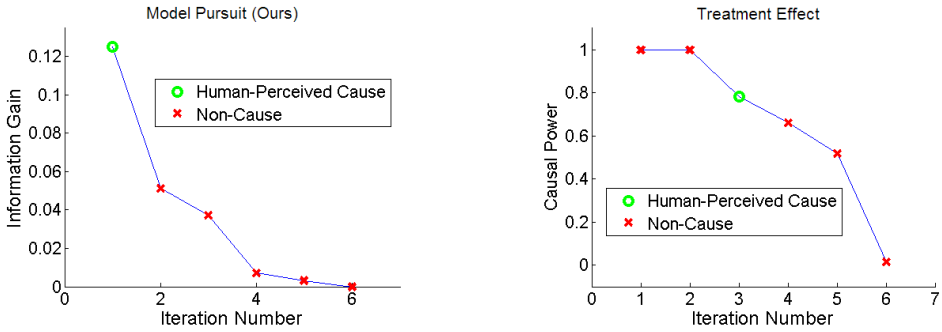
related to each other are suppressed once the cause is selected. Figure 4.13 shows our method outperforms Hellinger's χ^2. Unable to suppress the dependence between hierarchically-related actions once a parent (or child) action is selected, Hellinger's χ^2 identifies a non-cause before a real cause.

4.5.4 Elevator Data: Delayed Effects

This experiment uses detections from video of an elevator waiting area. For an elevator, the only detectable causing action to open the door is pushing the button that calls the elevator.

In this example, our method outperforms the treatment effect, TE, (Eq. 4.40) as shown in Figure 4.14.

In this scenario, for all 4 times that someone walked away, the elevator doors opened (because they had first pushed the call button). As a measure, the treatment effect favors relationships when an action coincidentally occurs with a fluent change 100% of the time—regardless of how infrequently the relationship is observed. Of the 19 total instances of opening doors, only 16 occur with the pushing button action under clip construction, so the true cause is not perfectly detected and cannot compete with the 100% coincidence as compared with TE. Our method, however, incorporates the frequency with which the relationship



(a) Our Method (b) TE

Figure 4.14: Confounded example: the elevator, pursuit order of causes.

occurs by examining the full contingency table.

4.5.5 Reasoning in Surprising Circumstances

Answers for "why" queries are obtained using MAP estimation. Observing a person pushing on the door while another agent walks by and yet another picks an object up, the learned probability model returns the correct reason for why the door is open.

If the door opens spontaneously (i.e., in a manner not seen by the system during learning), the probability model on the Causal And-Or Graph resolves the discrepancy by juggling the prior against which detection is more likely to be incorrect: the fluent change or the lack of action.

During the learning process for the monitor, however, the system saw several unexplained examples (i.e., when the computer put the monitor to sleep after sufficient time). In this case, the system learned to explain the status through the unexplained change, awarding 12% maximum posterior probability to the spontaneous change when no action was detected for turning the screen off.

4.6 Summary and Discussion

In this chapter, we have provided a learning framework for the perceptual causal structure between actions and fluents in video. Causal relations were incrementally determined using an information projection principle, and we provided analytic formulas for selecting and adding the best causal relation to the current probability model.

The information projection framework allows perceptual causal knowledge to be learned alongside actions and objects under other information projection frameworks, where information gains can be compared, and, for example, an important causal link could be added to a model before a less significant object or action. The learned Causal And-Or Graph aligns with forms used in vision for detecting actions, objects, and fluents, and flattens a causal network into choices. The Or-nodes in the Causal And-Or Graph place a prior on causality, to deal with the ambiguities of detections in vision.

General causal networks were too vague for our purposes. Cognitive science informed what variables to consider as causes and effects, how to partition a long video into "examples", and when to causally relate actions and fluents.

Our results match human perceptions of the causal connections between actions and fluent changes, showing that the Causal And-Or Graph is learnable from co-occurrence and the other heuristics (short temporal lag and agent actions cause fluent changes). Our method produced a better causal structure than TE and Hellinger's χ^2-statistic. It has the precision to select the correct action from a hierarchy, where a parent action may explain a fluent change better than any of its children actions separately or vice versa.

Causal knowledge is a required part to being able to fully explain the content of image and video data from an agentive point-of-view. Even with the threat of missing confounders, learning perceptual causality as given by the heuristics

brings vision research closer to higher-level reasoning.

CHAPTER 5

Inferring on the Causal And-Or Graph

In Equation 3.3 in Chapter 3, we introduced the probability for a parse graph at t from the Causal And-Or Graph:

$$P(pg_t|V[t-\delta,t]) \propto P(pg_t;\Theta) \prod_{l\in L(pg_t)} P(l|pg_t) \tag{5.1}$$

where $L(pg)$ is the set of included terminal leaf nodes, including both actions and fluents. In this chapter, we develop this grounded probability model, extend the Causal And-Or Graph to a model over time, and provide a Viterbi algorithm for inference.

5.1 Inference of a Single Parse Graph: The Energies

The prior model for causality, $P(pg;\Theta)$, indicates the level of prior belief for what the current fluent value is and the reason why the fluent took that value. We calculate $P(pg;\Theta)$ with the energy, $\mathcal{E}(pg)$, where $P(pg) \propto \exp(-\mathcal{E}(pg))$. The final score, $\mathcal{E}(pg)$, is recursively propagated to the top-level nodes in the Causal And-Or Graph by the following rules:

Or-nodes. The energy of an Or-node, O, is

$$\mathcal{E}(O) = \max_{v\in ch(O)} (\mathcal{E}(v) + \langle \lambda_v, \Theta_v \rangle) \tag{5.2}$$

where $ch(O)$ represents the children. λ_v indicates how likely each child is of causing the parent, and Θ_v indicates which child is selected. $\langle \lambda_v, \Theta_v \rangle$ returns the prior probability of selecting that particular child. λ_v is learned by maximum likelihood

estimation, giving the proportion of training examples that included child Θ_v. The learned λ_v favors the status quo, returning that the fluent maintained status a priori.

And-nodes. The energy of an And-node, A, with children $ch(A)$, ensures probabilities from all children are passed up to the top node, and is given by

$$\mathcal{E}(A) = \sum_{v \in ch(A)} \mathcal{E}(v|A). \tag{5.3}$$

Temporal relations. Top-level actions are detected as triads of sub-actions, with each allowing a variable number of pose detections. Relations preserve the temporal order of sub-actions. For relation R across nodes $\tilde{v} = v_{i_1}, \ldots, v_{i_k}$,

$$\mathcal{E}(R) = \psi_{\tilde{v}}(\tilde{v}), \tag{5.4}$$

and is described further in Section 5.3.1.

Leaf nodes. Terminal leaf nodes anchor the Causal And-Or Graph to features extracted from video. Using machine learning approaches, action and fluent detection algorithms independently provide $P(l|pg)$. The fluent energies, $\mathcal{E}(l_F|F)$, and the action energies, $\mathcal{E}(l_A|A)$, are calculated from the detected features, trained separately with machine learning approaches as described in Section 5.3.1.

Intuitively, $\mathcal{E}(A)$ and $\mathcal{E}(O)$ recursively score a complete parse graph. Decomposing the recursion,

$$\mathcal{E}(pg_t|V[t-\delta,t]) = \sum_{l_F \in L_F(pg)} \mathcal{E}(l_F|F) + \sum_{l_A \in L_A(pg)} \mathcal{E}(l_A|A) + \sum_{\tilde{v} \in R} \psi_{\tilde{v}}(\tilde{v}) + \sum_{v \in O(pg)} \langle \lambda_v, \Theta_v \rangle, \tag{5.5}$$

where $L_F(pg)$, $L_A(pg)$, $R(pg)$, and $O(pg)$ are the sets of included fluent leaves, action leaves, relations and Or-nodes, respectively.

Detections of actions and fluents are jointly considered for pg where temporal spacing between the two is within a pre-learned latent time, δ, learned by optimizing the hit rate as latency increases. Latent time between flipping a switch

and the light turning on is kept near instantaneous, whereas latent time between pushing an elevator call button and the elevator's arrival affords more leniency.

5.2 Reasoning over Time

Over time, a fluent takes a sequence of values, $F_1, \ldots, F_n$, and a series of actions $A_1, \ldots, A_k$ are performed. The Causal And-Or Graph models causal relationships as the fluent value transitions from F_{t-1} to F_t. In this section, we bind the Causal And-Or Graphs sequentially to model a sequence of parse graphs, $\mathbf{PG} = (pg_1, \ldots, pg_n)$, explaining a longer video. Greedily connecting the pg yields two concerns: (1) Subsequent parse graphs must be consistent, and (2) The process is non-Markovian.

5.2.1 Consistency of Transitions between Parse Graphs

Subsequent pg_{t-1} and pg_t from $\mathbf{PG}$ both contain the fluent value at $t-1$. Combining the local parse graphs pg_t and pg_{t-1} shown in Figure 5.1 requires pg' to maintain consistency—the final value of the former must match the incoming value of the latter. For example, multiple detections of flipping a light switch cannot all cause the light to turn on unless the light is turned off between them. The following state transition probability enforces consistency between subsequent parse graphs:

$$P(pg_t|pg_{t-1}) = \begin{cases} 0, \text{ if } pg_{t-1}, pg_t \text{ inconsistent} \\ 1, \text{ otherwise.} \end{cases} \tag{5.6}$$

5.2.2 Non-Markovian Duration

Fluents such as the computer monitor are non-Markovian: rather than following an exponential fall-off, the screensaver activates after a set amount of time

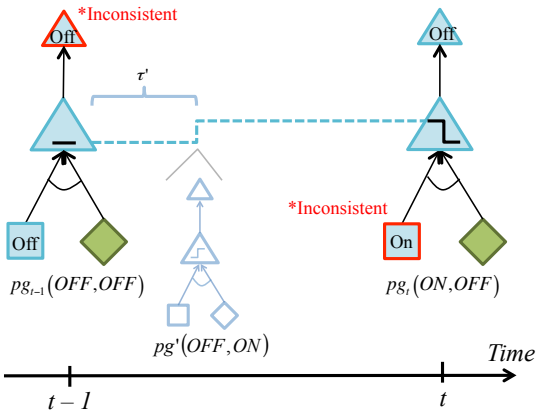


Figure 5.1: Inconsistent state transition. Combining local parse graphs pg_t and pg_{t-1} shown here requires insertion of pg' to maintain consistency—the final value of the former must match the incoming value of the latter.

(usually 5 minute increments), following a predictable distribution such as shown in Figure 5.2. Further, while a Markov process can insert the hidden trigger "thirst" between two subsequent observations of "drink", it has difficulty consistently matching human estimates as to where the insertion should go.

Both problems can be resolved by modeling the duration for which a given fluent maintains a particular value [Murphy, 2002]. We assume subsequent durations are independent, given the fluent value, or $P(\tau|F)$. We have had success approximating $P(\tau|F)$ with step functions, discretizing the probability model. Where commonsense knowledge is available, the models for $P(\tau|F)$ are directly coded (e.g., screensaver). When evidence is available, they are learned from observation by maximum likelihood estimation.

5.2.3 Inference of the Sequential Parse Graphs

To accommodate the non-Markovian duration terms while enforcing consistency, we use a hidden semi-Markov model, or variable-duration Markov model, [Murphy, 2002]. The graphical model shown in Figure 5.3 captures our assumed dependen-

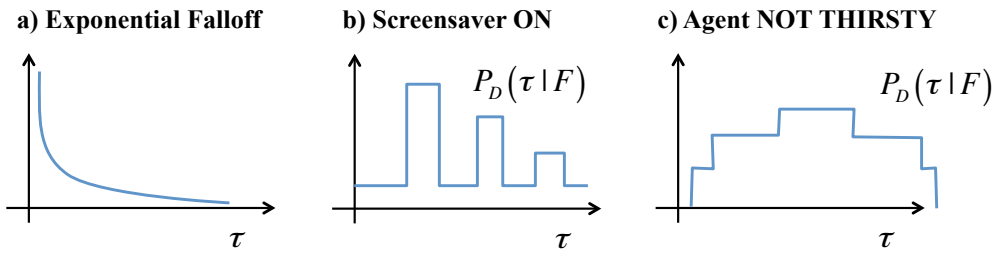


Figure 5.2: Fluent durations.

cies. In this model, PG_t from the Causal And-Or Graph is repeated for a duration of τ_t. L_t represents the sequence of observed fluents and actions under PG_t. The following conditional probability distributions govern the state transitions as well as handle a counter for the duration:

$$P(PG_t = pg | PG_{t-1} = pg', \tau_{t-1} = d) = \begin{cases} \delta(pg, pg'), \text{ if } d > 0 \\ \qquad \text{(remain in same state)} \\ P(pg|pg'), \text{ if } d = 0 \\ \qquad \text{(transition per Eq. 5.6).} \end{cases} \tag{5.7}$$

$$P(\tau_t = d' | PG_t = pg) = \begin{cases} \delta(d', d-1), \text{ if } d > 0 \\ \qquad \text{(decrement)} \\ P(\tau|F), \text{ if } d = 0 \\ \qquad \text{(per Sec. 5.2.2).} \end{cases} \tag{5.8}$$

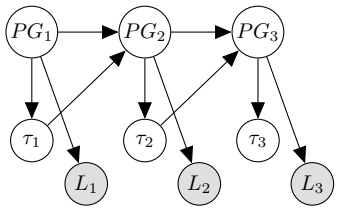


Figure 5.3: Hidden semi-Markov model

d and d' count down the duration, and δ is the Dirac delta function. The

optimal sequence explaining the video is given by

$$\mathbf{PG}^*, \tilde{\tau}^* = \underset{\mathbf{PG},\tilde{\tau}}{\operatorname{argmax}} P(\mathbf{PG}, \tilde{\tau}|V), \tag{5.9}$$

where $\tilde{\tau} = (\tau_1, \ldots, \tau_n)$ represents the durations corresponding to elements of $\mathbf{PG}$. To calculate $\mathbf{PG}^*$ and $\tilde{\tau}^*$, we run a Viterbi algorithm. For the hidden semi-Markov model, the Viterbi equations are

$$\begin{aligned} V_t(pg, \tau) &\triangleq \max_{pg',\tau'} P\left(PG_t = pg, \tau_t = \tau, PG_{t-1} = pg', \tau_{t-1} = \tau', L_{1:t} = l_{1:t}\right) && (5.10) \\ &= P(l_{t-\tau+1:t}|pg) \max_{pg',\tau'} P(pg, |pg')P(\tau|F)V_{t-\tau}(pg', \tau'). && (5.11) \end{aligned}$$

where $l_{1:t}$ is the subsequence emitted from 1 to t, consisting of action and fluent detections. By defining $V_t(pg) \triangleq \max_\tau V_t(pg, \tau)$, we can separate the maximization over τ from the state space:

$$V_t(pg) = \max_\tau \left[P\left(l_{t-\tau+1:t}|pg\right) P\left(\tau|F\right) \max_{pg'} P\left(pg|pg'\right) V_{t-\tau}(pg') \right] \tag{5.12}$$

Derivations are provided in Section 5.5. By precomputing $P\left(l_{t-\tau+1:t}|pg\right)$ (see action detection in Section 5.3.1), the complexity is $O(T \cdot |PG|^2 \cdot |\tau|)$ where $|\tau|$ is the maximum number of discrete durations considered. While this model can be approximated by an HMM with the addition of more nodes, complexity would increase.

To reduce complexity, we index t over detected change points (time points with either a fluent change or action detection). In order to accommodate this simplification, we assume at most one missed fluent change occurred between them. This is sufficient because our considered fluents are binary: in particular, we consider it possible that a light gets turned off between two detections of turning on, but we ignore the chance that there would be multiple missed detections of on/off. If pg_{t-1} and pg_t are inconsistent, we try to optimally insert a new change point, $t' \in (t-1, t)$ as shown in Figure 5.1, interpreting the inconsistency as missed information. $P(\tau|F)$ informs where to insert this change.

In general, all instances between these change points are best explained by the non-action causal parse graph: the fluent maintains status because no change-inducing action occurred. By jointly optimizing the parse graphs over time, we avoid early decisions, allowing new information to revise previous conflicts.

5.3 Experiment 1: Causal Grammar vs. Detections

To evaluate reasoning values of hidden fluents and actions, we use video captured with a Kinect in multiple scenes. The 4D-Kinect data includes RGB images with depth information and extracted human skeletons. Table 5.1 lists the 13 fluents included in the data and summarizes the number scenes, clips, and frames of each. Some examples were shown in Figure 1.2. The average clip length is approximately 300 frames. The data separates out a small labeled training set, providing between 3 and 10 instances of each fluent change (average of 13 frames per example), action (average of 98 frames per example), and causal relationship. Fluents with a small number of clips are case studies, and not included in summary results.

5.3.1 Baseline: Bottom-Up Fluent and Action Detection

We use machine learning algorithms for the bottom-up detection of fluent changes and actions.

Fluents: To calculate $\mathcal{E}(l_F|F)$, we use a 3-level spatial pyramid to compute features with 1, 4, and 16 blocks as shown in Figure 5.4. People detected by the Kinect are removed. The feature vector contains the mean, maximum, minimum, and variance of intensity and depth changes between subsequent frames at each level, using 6 window sizes from 5 to 30 frames. The GentleBoost algorithm [Friedman et al., 2000] is trained on 3 to 7 examples of each fluent change. The detectors for the light select one weak classifier: the mean of intensity change at the highest level. Other fluent changes need more than 20 weak classifiers.

Table 5.1: Dataset Included Action/Fluent Relationships

Object	Fluent	Causing Actions	nScenes	nClips	nFrames
door	open/closed	open door, close door	4	50	10611
light	on/off	turn light on/off	4	34	16631
screen	on/off	use computer	4	179	56632
phone	active/off	use phone	5	68	30847
cup	more/less/ same	fill cup, drink	3	48	16564
thirst	thirsty/not	drink	3	48	16564
waterstream	on/off	fill cup	3	40	14061
trash	more/less/ same	throw trash out	4	11	2586
microwave	open/closed, running/not	open door, close door turn on	1	3	4245
balloon	full/empty	blow up balloon	1	3	664
fridge	open/closed	open door, close door	1	2	2751
blackboard	written on/ clear	write on board, erase	1	2	5205
faucet	on/off	turn faucet on/ off	1	2	3013

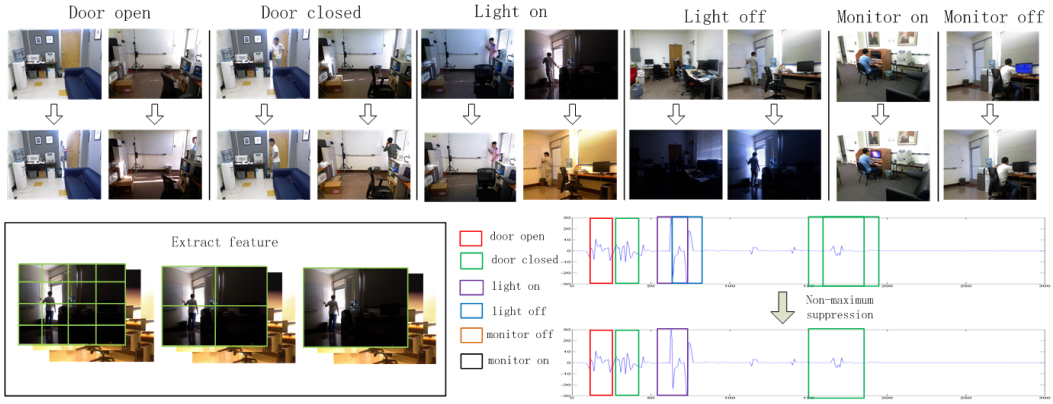


Figure 5.4: Fluent detection. Fluents are extracted with spatial pyramids and non-maximum suppression.

Actions: To compute $\mathcal{E}(l_A|A)$, we calculate pose features from the relative locations of each joint of the human skeleton as detected by the Kinect, shown in Figure 5.5. To calculate $\mathcal{E}(R)$, we bind the nodes in the relation by modeling $\psi(\tilde{v}) = P(v_n|v_{n-1}, d_{n-1})$ (where d_{n-1} is the duration the pose has been classified as v_{n-1}) with logistic regression over n, similar to [Wei et al., 2013]; model parameters were trained with a multi-class SVM [Chang and Lin, 2011]. Dynamic programming beam search [Tillmann and Ney, 2003] runs over the video, retaining only the top k performing action parse graphs. It is important to keep k high as beam search runs the risk of omitting the true action detection; we used $k = 1000000$. These values are propagated up the graph, providing a per-frame probability of each action category, over which we slide windows of 50, 100, and 150 frames to recognize complete top-level actions at different scales. These top-level action detections provide the "detection" baseline for actions and are used to precompute $P\left(l_{t-\tau+1:t}|pg\right)$.

Non-maximum surround suppression provides fluent and action detections for the "detection" baseline. The action and fluent detections exhibit missed and incorrect detections typical in vision.

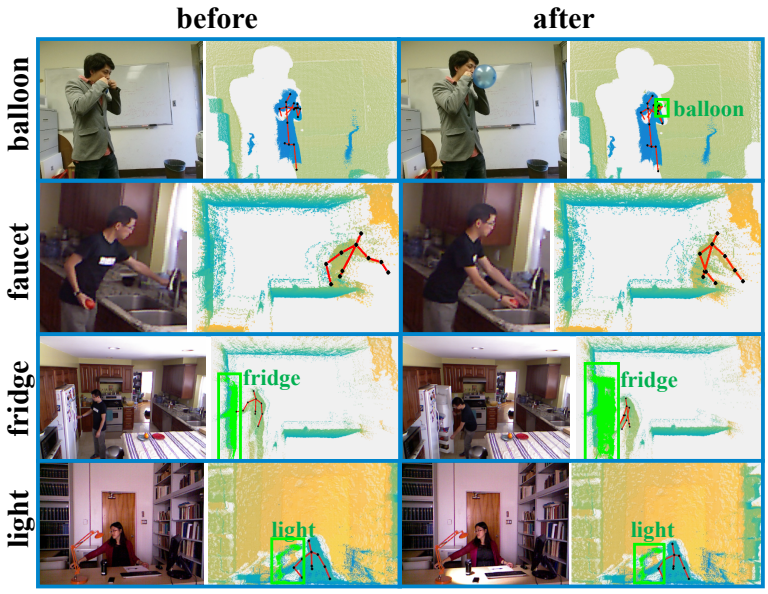


Figure 5.5: Human poses and depth images (before and after a fluent change) for actions as captured by the Kinect, together with sample frames.

5.3.2 Baseline: Random Noise

"Noise" answers all queries as equally likely, and provides a comparison lower bound.

5.3.3 Human Annotation

To evaluate results, we collected multiple human annotations by showing video clips showcasing actions, fluent changes, and non-actions. Participants provided an estimation on a scale of 0 to 100 for actions and fluent changes in each clip (e.g., Did the human dispense water to the cup? Is the cup more full, less full, or the same as in the previous clip? Is the human thirsty?). Between 1 and 7 clips were shown sequentially to create larger video sequences that included up to 4 objects. Participants were encouraged to revise their answers when new information warranted.

5.3.4 Protocol for Experiment Evaluation

Because we expect reasoning to occur across the clips, we compare the computer to the nearest human response, that is the human whose response for the video sequence is closest to the computer's as measured by the Manhattan distance. Hits are calculated when they *exactly* match the nearest human response for a single query. Ground truth *positives* are registered when the nearest human awarded more than 50% to a single answer.

5.3.5 Results

Raw fluent and action detections in Figure 5.6 show that causal relationships improve detections and clarify understanding. The action detectors (second and third plots) use pose to detect open and close actions, without distinguishing objects. Causal grammar combines these action detections with those of the microwave fluent (first plot) and shows that only some should be labeled "opening (or closing) the microwave".

Figure 5.7 shows results from detectors and causal grammar for the light and screen fluents. The fluent detectors erroneously detect multiple light and monitor changes as the light turns on (once) and the camera adjusts; causal reasoning mostly corrects these.

When asked for the monitor's status, humans produced the probabilities shown in the heat maps at the bottom of Figure 5.7. The computer screen is not visible, and humans (generally and specifically) exhibited large variability in examining hidden values. While they all agreed that the actor was using the computer, they did not have a consensus as to whether the screen was on or off or transitioning between the two.

Table 5.2 shows performance on individual actions and fluents. In all categories (as well as overall—causal grammar: average precision is $AP = 0.63$, and

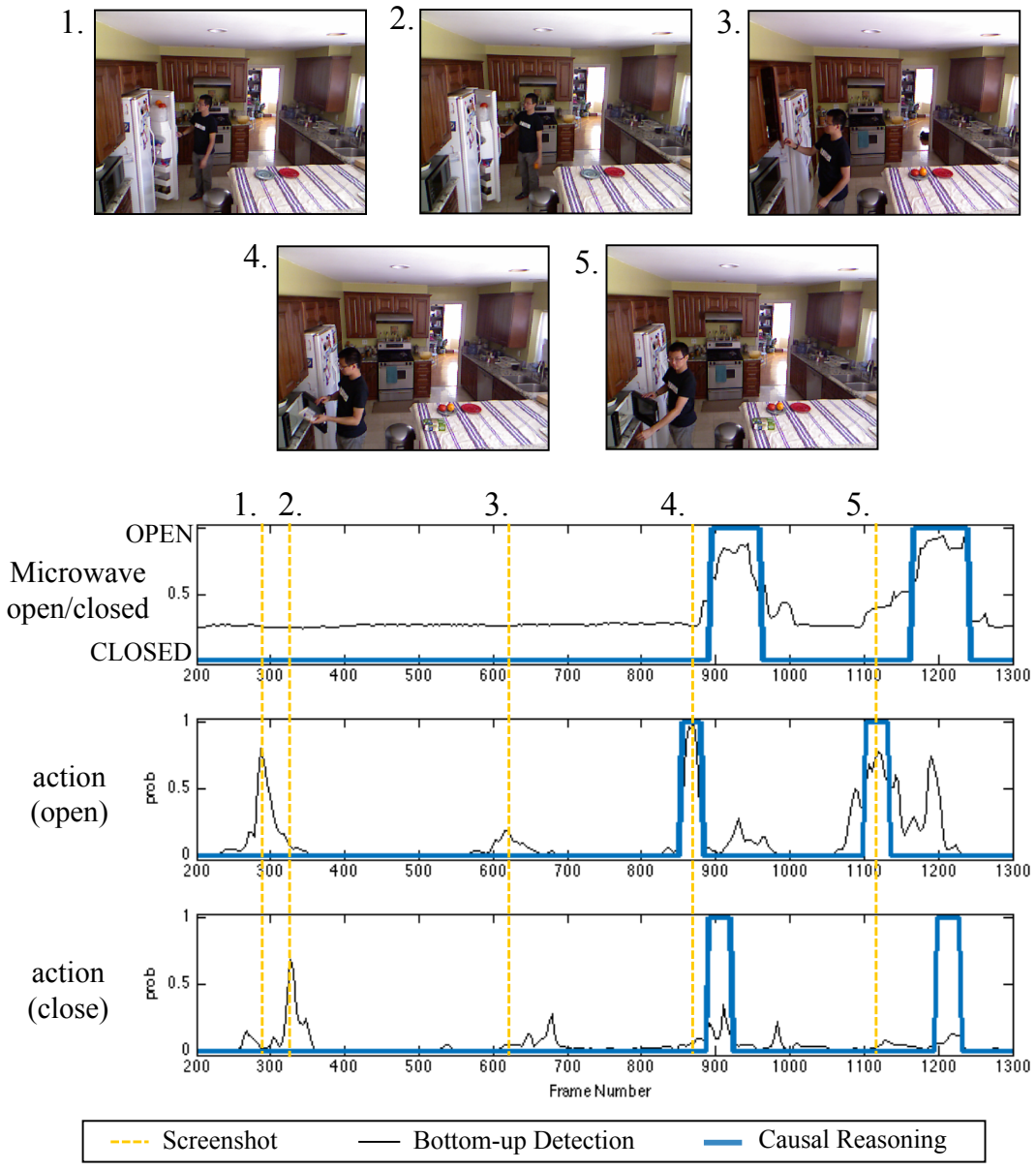


Figure 5.6: Microwave results from fluent detectors (top) and action detectors (middle and bottom), superimposed with causal reasoning results. Step functions mark fluent changes–up for turning on, down for turning off.

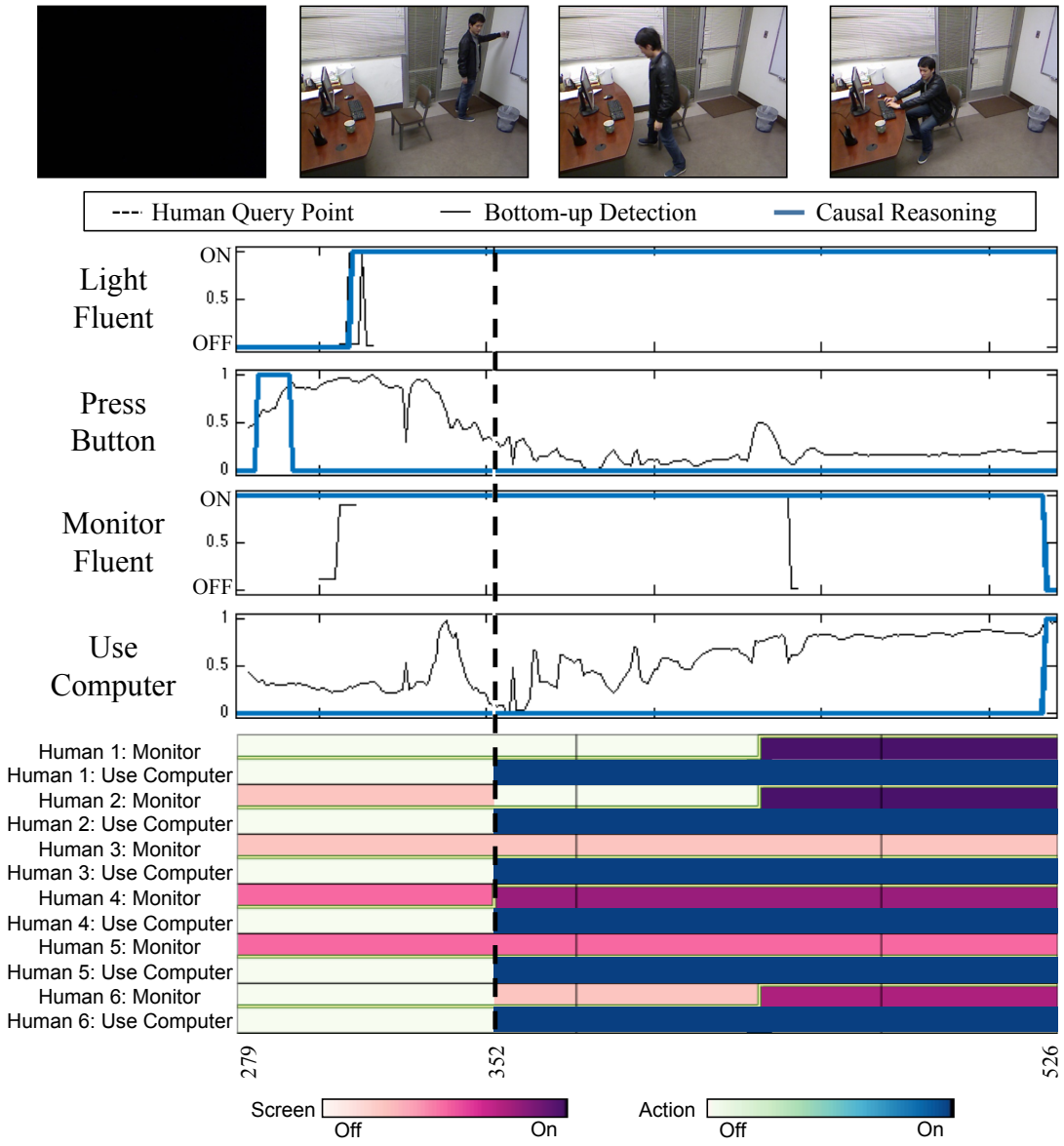


Figure 5.7: Results for screen and light example from fluent and action detectors, superimposed with causal reasoning results. Step functions mark fluent changes–up for turning on, down for turning off. Human answers to the queries of hidden variables (shown at the bottom) sometimes varied greatly. The dashed line separates the two query points for humans.

average recall is $AR = 0.69$; detections: $AP = 0.29$, $AR = 0.31$), using the sequential Causal And-Or Graph to jointly infer actions and fluents outperforms the independent fluent and action detections.

Our method achieves higher hit rates on fluents than on actions. While each action was detectable on at least some level, only door, light, and screen fluents were detectable (undetectable categories shown with italics). On detectable fluent examples, action and fluent detections compete to provide higher overall performance. Decisions for undetectable fluents are made through action detections, the prior causal understanding from the Causal And-Or Graph, and consistency over time.

Table 5.2: Hit rates for actions and fluents. Cup action is a combination of thirst and waterstream. Italics mark the undetectable fluents.

	Action			Fluent		
	Noise	Detection	Causal	Noise	Detection	Causal
trash	0.10	0.62	**0.87**	*0.00*	*0.00*	***0.77***
door	0.00	0.45	**0.58**	0.00	0.42	**0.53**
cup	N/A	N/A	N/A	*0.00*	*0.00*	***0.62***
light	0.00	0.57	**0.80**	0.00	0.43	**0.61**
screen	0.12	0.61	**0.67**	0.25	0.17	**0.74**
thirst	0.03	0.41	**0.76**	*0.08*	*0.11*	***0.57***
phone	0.00	0.33	**0.40**	*0.00*	*0.00*	***0.19***
waterstream	0.00	0.38	**0.88**	*0.00*	*0.00*	***0.81***
Average	0.04	0.48	**0.71**	0.04	0.14	**0.61**

Low detection rates in Table 5.2 indicate how challenging the dataset is. Further, categories where "noise" had a non-zero hit rate (e.g., trash) indicate that noise matched at least one human perfectly—humans had difficulty with detec-

tion for some clips. This further underscores the need for multiple annotations and how there is no so-called perfect ground truth. Evidence of using multiple annotations is evident for the thirst fluent: "detection" answered fluent queries identically to noise, but the average hit rate for "detection" is slightly higher because the action detections allowed it to be compared to a different human than "noise".

5.4 Experiment 2: Variability of Humans

To evaluate the variability of human answers, we use approximately 20 minutes of video data that was captured using a Kinect in two scenes: a hallway and an office. Table 5.3 contains a summary of the fluents contained in the video, as well as the values each fluent can take. These fluents are ambiguous in the video (e.g., light status (ambient light may be from a window or a light) or water stream (resolution is not high enough to see it) in Figure 5.8).

Table 5.3: List of fluents considered.

Computer: asleep/awake
Monitor Display: on/off
Monitor Power: on/off
Cup: more/less/same
Water Stream: on/off
Light: on/off
Phone: active/standby
Trash Can: more/less/same
Agent : thirsty/not
Agent: has trash/not

5.4.1 Human Annotation

Through a website, participants ($N = 15$) were shown the test video which paused at preset frames, e.g., those shown in Figure 5.8, and asked whether or not a fluent changed, similar to Experiment 1. At each key frame, the participant was asked to split 100 points across all possible values of each fluent, indicating his subjective probabilities of the fluent values. Each participant was allowed to revise previous judgments with information derived from subsequent frames.

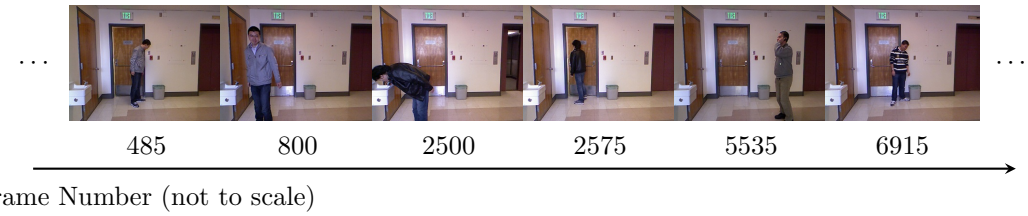


Figure 5.8: Sample of human judgment key frames.

5.4.2 Baseline Estimate (Random Noise).

For a baseline estimate, the hidden fluents were assigned uniformly, without using any detection or causal information (e.g., 50% that the light is on and 50% for off).

5.4.3 Computer Estimate (The Causal And-Or Graph).

Actions were manually segmented for a pre-specified grammar, and then poses captured by the Kinect camera were clustered. Temporal parsing transformed the clustered poses into hierarchically-labeled instances from the Temporal And-Or Graph [Pei et al., 2011]. The maximum probability action detections were used as input to the Causal And-Or Graph.

Fluent changes were detected from the video with the GentleBoost algorithm on features extracted as shown in Figure 5.4. Non-maximum suppression provided

the final detections of fluent changes.

The computer estimate is given by processing these action and fluent detections under the Causal And-Or Graph.

5.4.4 Results and Discussion

In the hallway, multiple changes in the light fluent were detected, yet no causing action was detected, presenting a common situation in vision—detections are usually imperfect. The Causal And-Or Graph corrects these errors by balancing detections with the consistency of causal explanations. Figure 5.9 shows equivalence classes of causal grammar results, sorted in order of probability.

The Causal And-Or Graph result was consistent with human judgments. Humans selected a single value for the light fluent for the duration of the video, but some selected on while others chose off. This reinforces the need to have a probabilistic model capable of maintaining multiple interpretations; the Causal And-Or Graph result included both solutions.

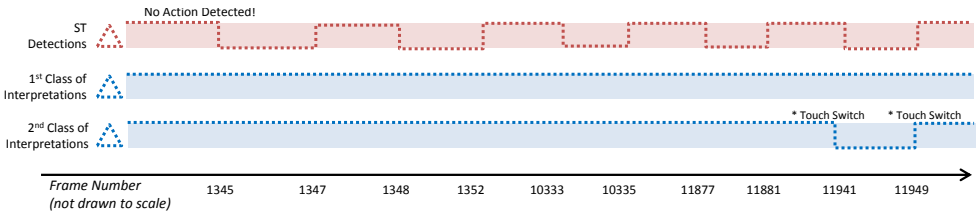


Figure 5.9: Correcting spatio-temporal detections. Given light fluent detections that move between on and off without a causing action, the Causal And-Or Graph prefers this to be explained by incorrect detections of the light fluent. The second most probable class of explanations is that two of the changes had causing actions that were missed by the detection.

MDS plots with human, computer, and baseline estimates are shown in Figure 5.10. Even though the set of possible fluent values was provided to partic-

ipants (significantly narrowing their available judgments), the MDS plots show wide variation in human responses. This is due to many factors. First, some participants initialized fluent values differently (e.g., light on versus off in Figure 5.8), resulting in a large total variation distance. Also, some participants were more cautious than others, recording judgments close to 50/50 where others took an all-or-nothing approach to assigning judgments.

In the hallway dataset, both fluent and action detections contribute to the causal inference of hidden fluents. Causal grammar performs similarly to human performance as shown in Figure 5.10(a), outperforming the noise baseline by far.

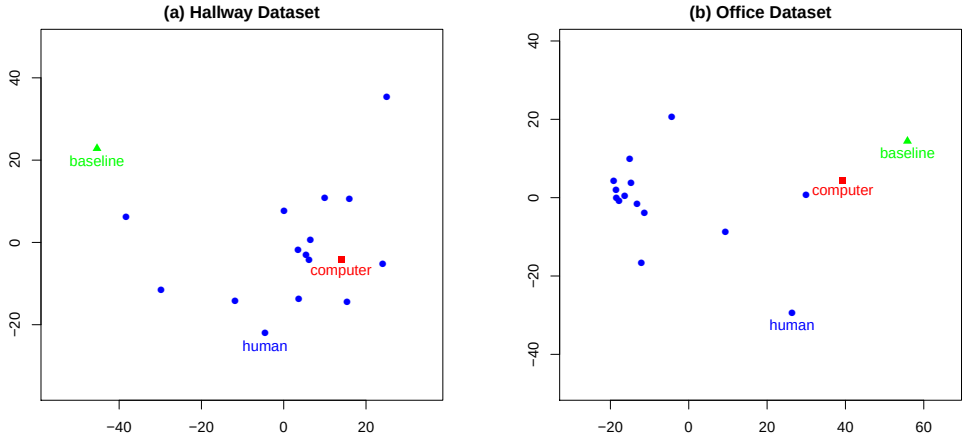


Figure 5.10: MDS plots of fluent value estimates. Blue circles: human estimates. Red squares: estimates using the Causal And-Or Graph. Green triangles: baseline estimates. See Further Discussion for notes on the human variability.

As evidenced by the Causal And-Or Graph's weak performance, the office dataset was particularly challenging. Action detections were poor and no fluent detections were available to identify conflicts, leaving the system heavily dependent on those incorrect action detections. Despite this disadvantage, the Causal And-Or Graph still provided enough reasoning capability to outperform the baseline. This underscores the importance of good vision-detection systems.

5.5 Deriving the Viterbi Algorithm

In this section, we develop the Viterbi algorithm for the hidden semi-Markov model, following the notation and development used in [Murphy, 2002]. Let $V_t(pg, \tau)$ be the maximum likelihood that partial state sequence ends at t in state pg of duration τ. We introduce C_t to indicate that τ_t is complete and, hence, the state is now allowed to change to PG_{t+1} and select a new duration τ_{t+1} as shown in Figure 5.11.

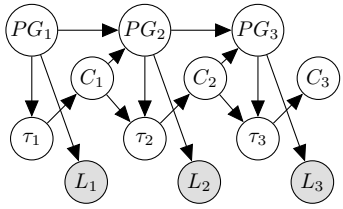


Figure 5.11: Hidden semi-Markov model with completion nodes

Under Figure 5.11, the hidden semi-Markov model is governed by the following 4 conditional probability distributions:

1. Transition states:

$$P(PG_t = pg | PG_{t-1} = pg', C_{t-1} = c) = \begin{cases} \delta(pg, pg'), \text{ if } c = 0 \\ \qquad \text{(remain in same state)} \\ P(pg|pg'), \text{ if } c = 1 \\ \qquad \text{(transition)} \end{cases} \tag{5.13}$$

2. Reset the duration counter:

$$P(\tau_t = d' | PG_t = pg, C_{t-1} = 1) = P(\tau = d' | F) \tag{5.14}$$

3. Continue counting down:

$$P(\tau_t = d' | \tau_{t-1} = d, PG_t = pg, C_{t-1} = 0) = \begin{cases} \delta(d', d-1), \text{ if } d > 0 \\ \text{undefined, if } d = 0 \end{cases} \tag{5.15}$$

4. Set to complete when counter is at 0:

$$P(C_t = 1 | \tau_t = d) = \delta(d, 0) \tag{5.16}$$

Using C_t, we define:

$$\begin{aligned} V_t(pg, \tau) \quad &\triangleq \quad \max_{pg', \tau'} P(PG_t = pg, C_t = 1, \tau_t = \tau, & (5.17) \\ & \qquad PG_{t-1} = pg', \tau_{t-1} = \tau', C_{t-1} = 1, l_{1:t}). & (5.18) \end{aligned}$$

Under assumed dependencies,

$$\begin{aligned} V_t(pg, \tau) \quad &= \quad P(l_{t-\tau+1:t} | PG_t = pg) & (5.19) \\ & \qquad \max_{pg', \tau'} [P(PG_t = pg, \tau_t = \tau | PG_{t-1} = pg') \\ & \qquad\qquad P(PG_{t-1} = pg', \tau_{t-1} = \tau', C_{t-1} = 1, l_{1:t-\tau})] \\ &= \quad P(l_{t-\tau+1:t} | PG_t = pg) & (5.20) \\ & \qquad \max_{pg', \tau'} P(PG_t = pg, \tau_t = \tau | PG_{t-1} = pg') V_{t-\tau}(pg', \tau') \\ &= \quad P(l_{t-\tau+1:t} | pg) \max_{pg', \tau'} P(pg, \tau | pg') V_{t-\tau}(pg', \tau') & (5.21) \end{aligned}$$

Since we assume the conditional independence,

$$P(pg, \tau | pg') = P(pg | pg') P(\tau | pg, pg') = P(pg | pg') P(\tau | pg) = P(pg | pg') P(\tau | F), \tag{5.22}$$

$V_t(pg, \tau)$ becomes

$$\begin{aligned} V_t(pg, \tau) \quad &= \quad P(l_{t-\tau+1:t} | pg) P(\tau | F) \max_{pg', \tau'} P(pg | pg') V_{t-\tau}(pg', \tau') & (5.23) \\ &= \quad P(l_{t-\tau+1:t} | pg) P(\tau | F) \max_{pg'} \left[P(pg | pg') \max_{\tau'} V_{t-\tau}(pg', \tau') \right]. & (5.24) \end{aligned}$$

To separate the duration from the state space, define:

$$V_t(pg) \triangleq \max_{\tau} V_t(pg, \tau). \tag{5.25}$$

Therefore,

$$V_t(pg) = \max_{\tau} \left[P(l_{t-\tau+1:t}|pg) P(\tau|F) \max_{pg'} \left[P(pg|pg') V_{t-\tau}(pg') \right] \right], \tag{5.26}$$

matching Equation 5.12.

5.6 Discussion and summary

In this chapter, we introduced a probability model for the sequential Causal And-Or Graph, enabling joint inference of the values of hidden fluents and actions over time from video. This generative model connects cognition to vision over time with higher-level reasoning.

Analogous to how humans infer actions and fluents given limited visual cues, joint inference with our Viterbi algorithm revised conclusions from early information, improved existing detections, and filled in those that were hidden or missed. While joint inference is not a cure-all for low detection rates, it is useful for mediating differences. Inference of hidden fluents (both as triggers and as effects) provides deeper cognition that can be used to understand, predict, and replicate human actions.

Action ambiguities make detection challenging. While we trained actions with 4D Kinect data for generalizability, actions were still limited to the ways our system saw them. How people turn a light on might not look the same from one room or context to the next and yet the relation to the fluent is the same: when the light turns on, we match the words "turn the light on" to the observed action. Our method suggests a meaningful way to classify actions: by their causal effects.

CHAPTER 6

Discussion and Future Work

In this dissertation, I have presented methods for learning and inferring perceptual causality from video. As a starting point for vision research, we assigned a perceptual causal link between actions and fluents when co-occurrence warranted, subject to "commonsense" heuristics. This perceptual causal knowledge acquired enhanced the computer's understanding of video, giving an explanation of why fluents changed (because an agent's action changed them) and why actions were most likely performed (to change fluents under the goal-driven view of the human mind).

The task of acquiring causal knowledge is a challenging one, relying on the accurate detection of both causes and effects. Vision systems have detection error: classifiers are not perfect; misdetection, occlusion, and bad data are common problems. Hand-labeling training video greatly improves detection, but is time-consuming.

Results presented here were limited to pre-specified action and fluent categories so that appropriate detectors could be trained. However, any choice of dictionary will exclude possible cases, despite the best intentions to include many potential confounders. Further, using the heuristics excludes possible causes (where, for example, a light might turn off because the building lost power).

The causal relationships studied here can occur in many different scenarios. A light could turn on by flipping a switch on a wall, toggling a switch on a lamp, using a remote, or many other ways, including non-manual. Actions, especially

paired with pose-based detection algorithms, have several types of ambiguities. Identifying that a light turned on, however, can be useful in matching novel actions to "turn the light on".

Any of the above ambiguities can lead to weak correlation. It is difficult to determine whether weak correlation signals the existence of a true confounding cause or is just due to noisy data. Our heuristics disallowed studying confounders, such as monitor power status, because our pre-specified action hierarchy excluded that interaction.

Even assuming our system captured true causal sufficiency, we are still unsure which causal questions are important. For example, consider observing a person using a keyboard and seeing the monitor activate. What action activated the monitor? Was it that the person was typing on the keyboard, or that the person was typing their password?

Even with its problems, perceptual causality allowed us to construct a working model of causality from video. And getting things wrong can be okay: if a human repeatedly perceives something, they still form a model based on that. The model may not be correct, but it can yield useful results. Nonetheless, there are many areas for improvements.

As a first step, the learning process was designed to be consistent with spatio- and temporal-And Or Graph models. This needs to be integrated with current models so that perceptual causal relationships can be learned alongside actions and objects.

One area for future research is online learning: adapting the model as new "surprising" information comes in, as a human would. Largely observational data (captured, for example, with a static surveillance camera) cannot guarantee both positive and negative examples of each action considered, which are needed to accurately assign causes to effects. A dynamic experimental design could help

to determine what on-camera interventions would best confirm (or refute) the model's belief about causal relationships. To solve this problem, we need to introduce measurements for variability and uncertainty to estimate how the system is learning in the absence of ground truth.

Future work also includes investigating other paradigms for learning. How can we computationally make a Bayesian prior model work?

More future work includes expanding the reasoning capacity of the Causal And-Or Graph. How can we, for example, infer that an agent is unlocking a door when this action is always completely hidden (and unnecessary if the door is unlocked)? Or, what is the intent when someone exits a building, approaches his car, and returns to the building? Did he forget his key, forget to do something inside, or was approaching the car irrelevant? Future work includes exploring other factors involved with causal relationships, such as latency between action and effect, agentive intent, and different ways to impute hidden causes. More immediately, the Causal And-Or Graph reasons positively, but a dual graph that reasons "why not" instead of "why" can also aid the reasoning process.

Bibliography

M. Al-Hames and G. Rigoll. A multi-modal mixed-state dynamic bayesian network for robust meeting event recognition from disturbed data. In *ICME*, 2005.

M. Albanese, R. Chellappa, N. Cuntoor, V. Moscato, A. Picariello, V.S. Subrahmanian, and O. Udrea. Pads: A probabilistic activity detection framework for video data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2246–2261, 2010.

J.F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579, 1994.

M. Brand. The "inverse hollywood problem": From video to scripts and storyboards via causal analysis. In *Proceedings of the National Conference on Artifial Intelligence*, pages 132–137, 1997.

M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *CVPR*, 1997.

W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. In *CVPR*, 2011.

S. Carey. *The origin of concepts*. Oxford University Press, 2009.

C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27, 2011.

G. Csibra and G. Gergely. Obsessed with goals: Functions and mechanisms of teleological interpretation of actions in humans. *Acta psychologica*, 124(1):60–78, 2007.

I. Csiszár and P. C. Shields. Information theory and statistics: A tutorial. *Communications and Information Theory*, 1(4):417–528, 2004.

S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, 1997.

A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

A. Fire and S.-C. Zhu. Learning perceptual causality from video. In *AAAI Workshop: Learning Rich Representations from Low-Level Sensors*, 2013a.

A. Fire and S.-C. Zhu. Using causal induction in humans to learn and infer causality from video. In *The Annual Meeting of the Cognitive Science Society (CogSci)*, 2013b.

A. Fire and S.-C. Zhu. Learning perceptual causality from video. *ACM Trans. Intell. Syst. Technol.*, 7(2):23:1–23:22, 2016.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

T.L. Griffiths and J.B. Tenenbaum. Structure and strength in causal induction. *Cognitive Psychology*, 51(4):334–384, 2005.

T.L. Griffiths and J.B. Tenenbaum. Two proposals for causal grammars. *Causal learning: Psychology, philosophy, and computation*, pages 323–345, 2007.

A. Gupta, A. Kembhavi, and L.S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(10):1775–1789, 2009a.

A. Gupta, P. Srinivasan, J. Shi, and L.S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009b.

A. Hakeem, Y. Sheikh, and M. Shah. Caseˆe: A hierarchical event representation for the analysis of videos. In *NCAI*, 2004.

D. Heckerman. A bayesian approach to learning causal networks. In *UAI*, 1995.

D. Hoiem, A.A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

Y.A. Ivanov and A.F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.

H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

J.L. Mackie. Causes and conditions. *American philosophical quarterly*, 2(4):245–264, 1965.

R. Mann, A. Jepson, and J.M. Siskind. The computational perception of scene dynamics. *Computer Vision and Image Understanding*, 65(2):113–128, 1997.

E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006. ISBN 0123693888.

K. Murphy. Hidden semi-markov models (hsmms). Unpublished notes, 2002.

I. Newton. *The method of fluxions and infinite series: with its application to the geometry of curve-lines.* printed by Henry Woodfall; and sold by John Nourse, 1736.

J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

J. Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984. ISBN 0-201-05594-5.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

M. Pei, Y. Jia, and S.-C. Zhu. Parsing video events with goal inference and intent prediction. In *ICCV*, 2011.

T.-H. Pham, A. Kheddar, A. Qammaz, and A. Argyros. Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *CVPR*, 2015.

K. Prabhakar, S. Oh, P. Wang, G.D. Abowd, and J.M. Rehg. Temporal causality for the analysis of visual events. In *CVPR*, 2010.

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62 (1):107–136, 2006.

D.B. Rubin. The design versus the analysis of observational studies for causal effects: parallels with the design of randomized trials. *Statistics in medicine*, 26(1):20–36, 2007.

M. S. Ryoo and J. K. Aggarwal. Ut-interaction dataset, international conference on pattern recognition (icpr) contest on semantic description of human activities (sdha). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

M.S. Ryoo and J.K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *CVPR*, 2006.

M. Saberian, Z. Cai, J. Lee, and N. Vasconcelos. Using context to improve cascaded pedestrian detection. In *International SoC Design Conference (ISOCC)*, 2014.

R. Saxe and S. Carey. The perception of causality in infancy. *Acta psychologica*, 123(1):144–165, 2006.

R. Saxe, JB Tenenbaum, and S. Carey. Secret agents inferences about hidden causes by 10-and 12-month-old infants. *Psychological Science*, 16(12):995–1001, 2005.

Anne Schlottmann and David R Shanks. Evidence for a distinction between judged and perceived causality. *The Quarterly Journal of Experimental Psychology*, 44 (2):321–342, 1992.

B. J. Scholl and K. Nakayama. Illusory causal crescents: Misperceived spatial relations due to perceived causality. *PERCEPTION-LONDON-*, 33:455–470, 2004.

Z. Si, M. Pei, Z.Y. Yao, and S.-C. Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *ICCV*, 2011.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2000.

B. Taylor, V. Karasev, and S. Soatto. Causal video object segmentation from persistence of occlusions. In *CVPR*, 2015.

C. Tillmann and H. Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics*, 29(1):97–133, 2003.

S. Tran and L. Davis. Event modeling and recognition using markov logic networks. In *ECCV*, 2008.

Z. Tu, X. Chen, A.L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2): 113–140, 2005.

P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for event and object recognition. In *ICCV*, 2013.

D. Xie, S. Todorovic, and S.-C. Zhu. Inferring "dark matter" and "dark energy" from videos. In *ICCV*, 2013.

B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

S.-C. Zhu and D. Mumford. *A Stochastic Grammar of Images*. Now Publishers Inc., Hanover, MA, USA, 2006. ISBN 1601980604, 9781601980601.

S.-C. Zhu, Y.N. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.

Y. Zhu, Y. Zhao, and S.-C. Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *CVPR*, 2015.