# Inferring Hidden Statuses and Actions in Video by Causal Reasoning

Amy Fire and Song-Chun Zhu
University of California, Los Angeles
amy.fire@ucla.edu, sczhu@stat.ucla.edu

## Abstract

*In the physical world, cause and effect are inseparable: ambient conditions trigger humans to perform actions, thereby driving status changes of objects. In video, these actions and statuses may be hidden due to ambiguity, occlusion, or because they are otherwise unobservable, but humans nevertheless perceive them. In this paper, we extend the Causal And-Or Graph (C-AOG) to a sequential model representing actions and their effects on objects over time, and we build a probability model for it. For inference, we apply a Viterbi algorithm, grounded on probabilistic detections from video, to fill in hidden and misdetected actions and statuses. We analyze our method on a new video dataset that showcases causes and effects. Our results demonstrate the effectiveness of reasoning with causality over time.*

## 1. Introduction

Humans, motivated by triggering conditions [6], perform actions to cause changes in fluents (specifically the time-varying properties of objects and humans [15]). In this paper, we apply short-term causal knowledge consistently over the course of a video in order to jointly infer actions and fluents from video, even when they are unobservable. This improves detection and moves toward higher-level cognition, answering the questions of "why" and "how".

To study the causal relationships between actions and fluents, we introduce a new causality video dataset in Section 3, some examples of which are shown in Figure 1. In this new dataset, 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 defined by their causal effects: a "blowing" action is not detectable, but can be reasoned from the expanding

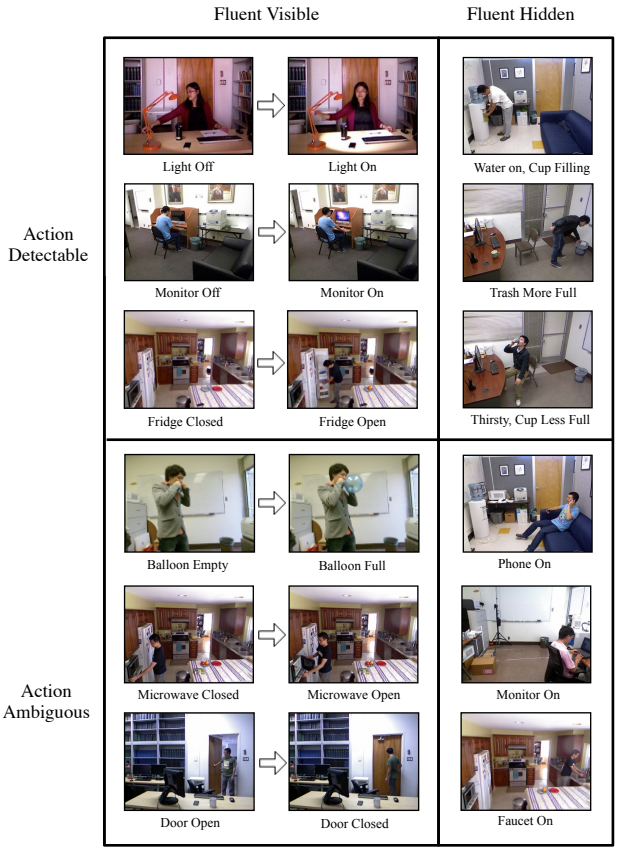


Figure 1. Fluents are time-varying properties of objects and may be visible or hidden (*e.g.*, invisible or viewpoint occluded); they change as a result of causing actions. Some actions may be easily detectable, while others are ambiguous (*e.g.*, motions too small/occluded, or confused for other actions). Under the context of causal relationships between actions and fluents, detections improve.

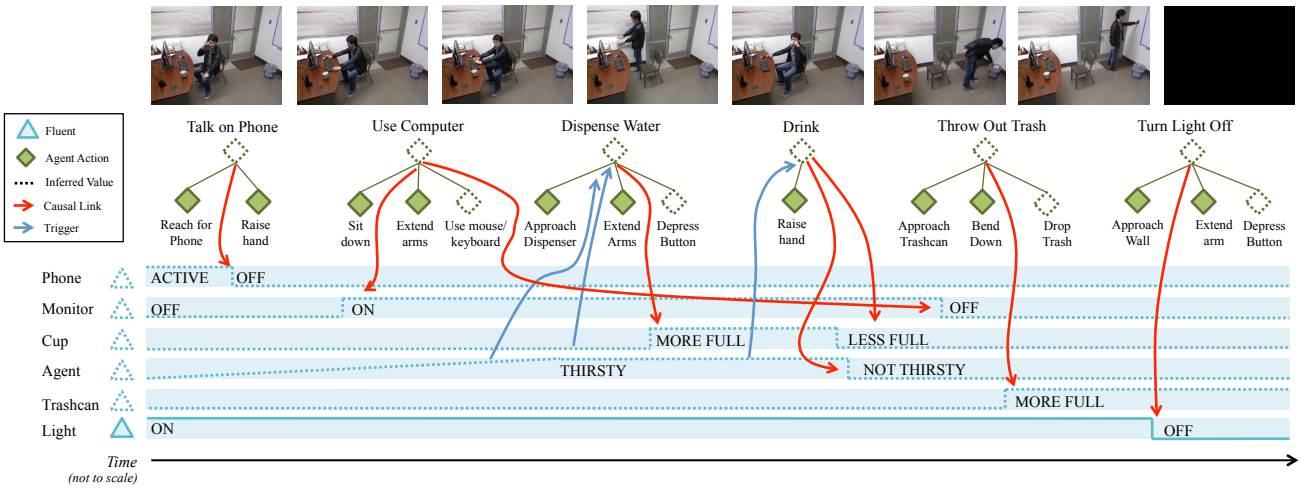


Figure 2. Example causal inference. Over time, observed actions are used to infer values of hidden fluents, and values of observed fluents are similarly used to infer hidden actions.

balloon.

Connecting triggering conditions to actions to effects, Figure 2 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. 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 clip, we might infer that the monitor is inactive.

### 1.1. Related work

Inferring causal relationships from video combines current computer vision detection algorithms with artificial intelligence and human thought.

**Computer vision.** Vision researchers have made great strides by studying context. Recognition rates improve for small objects when taken in the context of human actions and scenes [11] or for pedestrians when taken in the context of the scene [22]. The context of causality has been used in the spatial domain to aid segmentation [26].

Using causality, event recognition papers unidirectionally infer actions [2, 12], but they do not jointly infer causes and effects, nor do they propagate results over time. Measures of causality have been used to learn patterns of low-level actions in repeated events [19], and some early vision works used Newtonian mechanics to distinguish actions [14].

Further, action datasets (*e.g.*, Olympic Sports Dataset [17] and UCF-101 [25]) largely ignore cause and effect relationships, focusing instead on human motion (*e.g.*, HMDB51 [13]), complex activities (*e.g.*, basketball dataset[4]), or human interactions (*e.g.*, UT-Interaction Dataset [21]).

**Artificial intelligence.** AI researchers use first-order logic to reason with causality [15], but this precludes the probabilistic solutions important in computer vision for maintaining ambiguity. Placing probability atop first-order logic, Markov logic networks [20] have been applied to actions [28], but their network structure is pre-defined (not reconfigurable) and inference is slow.

While Bayesian networks are commonly used to represent causality [18], reconfigurations within a grammar model represent a greater breadth of possibilities than a single instance of a Bayesian network with pre-defined structure [10], making it more suitable for vision applications. The And-Or Graph graphically embodies grammar models and has been used for objects, scenes, and actions [30]. Even though HMMs and DBNs also perform event recognition [1, 3], grammar models are reconfigurable and accommodate high-level structure, both of which are needed for reasoning over time-varying detections of actions and fluents.

**Cognitive science.** Causal connections are so strong in humans that they can even override spatial perceptions [24]. Studies in developmental psychology show humans innately form causal models through correlation [9], restricted by heuristics such as only considering causes that are human actions [5, 23]. Learning this *perceptual causality* was introduced to vision in a simple experimental setting [8] with a grammar model, the Causal And-Or Graph (C-AOG) [7].

### 1.2. Contributions

In this paper, we develop a probability model for the C-AOG [7] that integrates with real detections. We extend the C-AOG to a sequential model, allowing long-term infer-

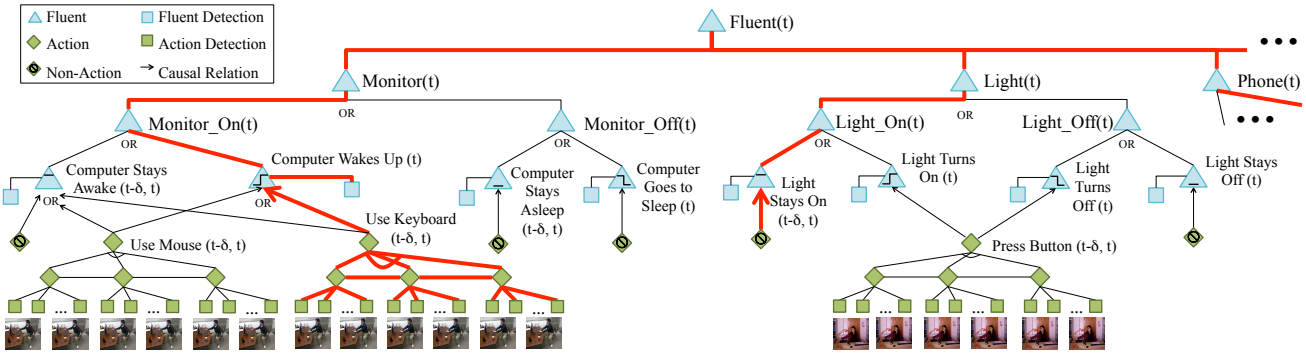


Figure 3. A C-AOG for an office at time $t$. Fluent values are consequences of their children. Arcs connect children of And-nodes. A single selection at the Or-nodes (red, bold lines here) provides a parse graph, explaining the current instance of time. Terminal leaf nodes ground the C-AOG on video, linking input from detected features. Step functions indicate types of fluent changes: step up for turning "on", step down for "off".

ence 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. For the evaluation of causal relationships, we introduce a dataset that combines both actions and fluents (Section 3).

## 2. Inferring perceptual causality

The Causal And-Or Graph (C-AOG) adds a causal layer to 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 [7, 8]. In this section, we formalize the sequential C-AOG by grounding a probability model for it on computed features, by extending the C-AOG over time, and by providing a Viterbi algorithm for inference.

### 2.1. Perceptual causality: the C-AOG

Given a short video sequence, $V[t-\delta, t]$, the C-AOG represents causal explanations for fluents at time $t$ where causing actions occur within the $\delta$ 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).

In the C-AOG, Or-nodes represent alternate means of causation (*e.g.*, a monitor can be woken by someone using a mouse *or* a keyboard). And-nodes group sub-actions and conditions (*e.g.*, the sub-actions used to detect "use keyboard"). Terminal leaf nodes represent low-level features for detecting actions and fluent changes in video. Horizontal links connect nodes with temporal relationships (*e.g.*, a person nears the computer before using it). Arrows point from causes to effects.

A parse graph ($pg$) from the C-AOG is formed by making a selection at each Or-node (*e.g.*, the thicker, red lines in Figure 3) and captures the causal reason that the fluent changed value at time $t$. The best parse graph at $t$ is given by selecting the best children per

$$P(pg_t|V[t-\delta,t]) \propto P(pg_t;\Theta) \prod_{l\in L(pg_t)} P(l|pg_t) \quad (1)$$

where $L(pg)$ is the set of terminal leaf nodes included in $pg$. As explained in the next section, this posterior is a product of the prior defined over the C-AOG (with parameter vector $\Theta$) and the likelihood of all leaf nodes for fluent and action detectors.

### 2.2. Inference of a single parse graph: the energies

$P(pg;\Theta)$ defines a prior on causality, indicating a level of prior belief for the current fluent value and why the fluent took that value. We calculate $P(pg;\Theta)$ with the energy $\mathcal{E}(pg)$, where $P(pg) \propto \exp(-\mathcal{E}(pg))$. $\mathcal{E}(pg)$ is recursively propagated to the top-level nodes in the C-AOG by the following rules:

**Or-nodes**. The energy of an Or-node, $O$, is $\mathcal{E}(O) = \max_{v\in ch(O)} (\mathcal{E}(v) + \langle\Theta_v, \lambda_v\rangle)$ where $ch(O)$ represents the children of Or-node $O$. $\Theta_v$ indicates how likely each child is of causing the parent, and $\lambda_v$ indicates which child is selected. $\langle\Theta_v, \lambda_v\rangle$ returns the prior probability of selecting that particular child. $\Theta_v$ can be learned by MLE, giving the proportion of training examples that included child $\lambda_v$. The learned $\Theta_v$ favors the status quo, *i.e.*, that the fluent maintained status a priori.

**And-nodes**. The energy of an And-node, $A$, with children $ch(A)$ passes probabilities from all children up to the top node, and is given by $\mathcal{E}(A) = \sum_{v\in ch(A)} \mathcal{E}(v|A)$.

**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 $\tilde{v} = v_{i_1}, \ldots, v_{i_k}$, $\mathcal{E}(R) = \psi_{\tilde{v}}(\tilde{v})$, and is described further in Section 3.5.

**Leaf nodes**. Terminal leaf nodes anchor the C-AOG to features extracted from video. 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 3.5. Treated independently, $\mathcal{E}(l_A|A)$ and $\mathcal{E}(l_F|F)$ sum to provide $\mathcal{E}(l|pg)$.

$\mathcal{E}(A)$ and $\mathcal{E}(O)$ recursively compute energies for all included nodes. Decomposing the recursion,

$$\begin{aligned}\mathcal{E}(pg_t|V[t-\delta,t]) = &\sum_{l_F\in L_F(pg)} \mathcal{E}(l_F|F) \\ &+ \sum_{l_A\in L_A(pg)} \mathcal{E}(l_A|A) + \sum_{\tilde{v}\in R} \psi_{\tilde{v}}(\tilde{v}) \\ &+ \sum_{v\in O(pg)} \langle \Theta_v, \lambda_v \rangle ,\end{aligned} \tag{2}$$

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 latent time, $\delta$, which can be pre-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.

## 2.3. 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 C-AOG models causal relationships as the fluent value transitions from $F_{t-1}$ to $F_t$. In this section, we bind the C-AOGs 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.

### 2.3.1 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 parse graphs $pg_t$ and $pg_{t-1}$ shown in Figure 4 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:

$$P(pg_t|pg_{t-1}) = \begin{cases} 0, \text{ if } pg_{t-1}, pg_t \text{ inconsistent} \\ 1, \text{otherwise.} \end{cases} \tag{3}$$

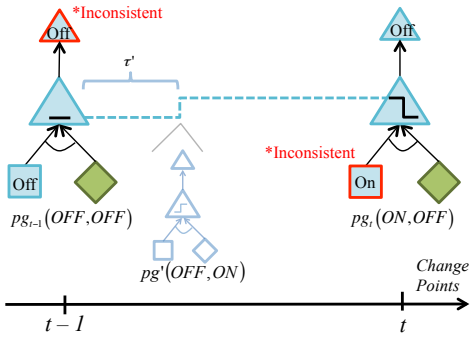


Figure 4. Inconsistent state transition.

### 2.3.2 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 (usually 5 minute increments), following a predictable distribution such as shown in Figure 5. 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.

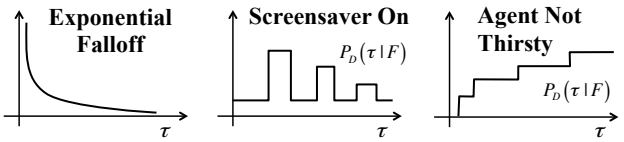


Figure 5. Fluent durations.

Both problems can be resolved by modeling the duration for which a given fluent maintains a particular value, $P(\tau|F)$. We assume subsequent durations are independent, given the fluent value. $P(\tau|F)$ can be approximated with step functions, discretizing the probability model. The models for $P(\tau|F)$ can be directly coded (*e.g.*, screensaver) where commonsense knowledge is available, and learned by MLE otherwise.

### 2.3.3 Hidden semi-Markov model for inference of the sequential parse graphs

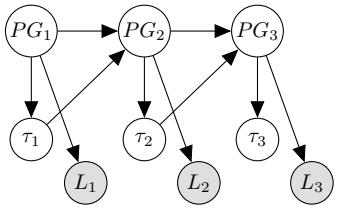


Figure 6. Hidden semi-Markov model.

A hidden semi-Markov model [16] can accommodate the non-Markovian duration terms while enforcing consistency.

The graphical model shown in Figure 6 captures our assumed dependencies. In this model, $PG_t$ from the C-AOG is repeated for a duration of $\tau_t$. $L_t$ 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:

$$P(PG_t = pg | PG_{t-1} = pg', \tau_{t-1} = d) = \begin{cases} 1(pg, pg'), \text{ if } d > 0 \text{ (remain in same state)} \\ P(pg|pg'), \text{ if } d = 0 \text{ (transition per Eq. 3).} \end{cases} \quad (4)$$

$$P(\tau_t = d' | PG_t = pg) = \begin{cases} 1(d', d-1), \text{ if } d > 0 \text{ (decrement)} \\ P(\tau|F), \text{ if } d = 0 \text{ (per Sec. 2.3.2).} \end{cases} \quad (5)$$

$d$ and $d'$ count down the duration, and 1 is the Dirac delta function. The optimal sequence explaining the video is given by

$$\mathbf{PG}^*, \tilde{\tau}^* = \underset{\mathbf{PG}, \tilde{\tau}}{\operatorname{argmax}} P(\mathbf{PG}, \tilde{\tau}|V), \quad (6)$$

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 with equations

$$V_t(pg, \tau) \quad (7)$$

$$\triangleq max_{pg', \tau'} \left( P \left( \begin{array}{c} PG_t = pg, \tau_t = \tau, \\ PG_{t-1} = pg', \tau_{t-1} = \tau', \\ L_{1:t} = l_{1:t} \end{array} \right) \right) \quad (8)$$

$$= P(l_{t-\tau+1:t}|pg) \max_{pg', \tau'} P(pg|pg') P(\tau|F) V_{t-\tau}(pg', \tau'). \quad (9)$$

By defining $V_t(pg) \triangleq \max_\tau V_t(pg, \tau)$, we can separate the maximization over $\tau$ from the state space:

$$V_t(pg) = \max_\tau \left[ \begin{array}{c} P\left(l_{t-\tau+1:t}|pg\right) P\left(\tau|F\right) \\ \max_{pg'} P\left(pg|pg'\right) V_{t-\tau}(pg') \end{array} \right]. \quad (10)$$

Derivations are provided in the supplemental materials. By precomputing $P\left(l_{t-\tau+1:t}|pg\right)$ (see action detection in Sec. 3.5), the complexity is $O(T \cdot |PG|^2 \cdot |\tau|)$ where $|\tau|$ is the maximum duration considered. This model can be approximated by an HMM with the addition of more nodes, increasing complexity.

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. 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 4, 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.

## 3. The causality video dataset and experiments

### 3.1. The causality video dataset

This paper introduces a new video dataset (examples shown throughout) to evaluate reasoning amid hidden fluents and actions. The 4D-Kinect data from multiple scenes includes RGB images with depth information and extracted human skeletons. Table 1 lists the 13 objects and the corresponding fluents included in the dataset and summarizes the number scenes, clips, and frames of each. The average clip length is approximately 300 frames. Fluents changes last an average of 13 frames, and actions take an average of 98 frames to complete. A small training set provides between 3 and 10 instances of each fluent change, action, and causal relationship. Fluents with a small number of clips are case studies, and not included in summary results. The dataset is available at `http://vcla.stat.ucla.edu/Projects/CausalReasoning/`.

Unlike the activity recognition datasets mentioned in Section 1.1, this causality dataset showcases cause and effect relationships between actions and object responses/fluent changes. This dataset includes long-term scenes that require reasoning over time.

Placed among human-centric causal contexts, the included fluents reflect a cross-section of those that are detectable (*e.g.*, the light is on or off), confusable (*e.g.*, the refrigerator door fluent is confused with the office door fluent), and inferable (*e.g.*, that the waterstream is on is inferred when the filling cup action is detected).

This dataset specifies the particular values fluents can take, discretizing the continuum in intuitive ways. For example, it is nearly impossible to infer beyond a cup having more/less/same, just as it is hard to quantify the amount of fill of a balloon (empty/not).

The dataset includes ambiguity in actions. Some viewpoints occlude actions, providing ambiguity where the action would otherwise be detectable (*e.g.*, a person positioned in front of a computer, occluding the action of using the computer). Some actions are confused for others (*e.g.*, taking a drink and making a phone call have similar poses). Other actions are hard to detect (*e.g.*, a person presses a small button to start the microwave).

Table 1. Dataset Included Action/Fluent Relationships

| Object | Fluent Values | 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 |

### 3.2. Ground truth: Human annotation

To evaluate results, we collected multiple human annotations by showing video clips with 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.

The annotators were both computer vision students and lay-people. There were 21 total annotators. Each video had between 5 and 7 independent annotators.

The responses by annotators varied, producing a higher number of distinct annotations for ambiguous (occluded) scenes than for "detectable" ones. Figure 9 shows an example of this. When asked for the monitor's status, humans produced the probabilities shown in the heat maps at the bottom of Figure 9. 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 in this case was using the computer, they lacked a consensus as to whether the screen was on or off or transitioning between the two. Each distinct response provides a different interpretation for the events in the scene (such as the two ways to interpret the Necker cube).

We base ground truth on human annotations in order to preserve these multiple interpretations. Requiring a computer to land on a single (seemingly arbitrary according to what is visible in the scene) physical interpretation fails to reflect the nature of the problem. We accept each human answer as a possible ground truth (*i.e.*, a valid interpretation of the scene), preserving all annotations.

### 3.3. Protocol for experiment evaluation

We compare each computer (noise, detection, C-AOG, sequential C-AOG) to its own 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. It is important to compare a computer to a *single* human for an entire video because we expect reasoning to occur across the clips.

Hits are calculated when a computer response *exactly* matches 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, where 50% indicates a preference for the choice. This threshold was used to determine whether a miss was a "false positive" or a "false negative"; and whether a hit was a "true positive" or a "true negative".

### 3.4. Baseline: Noise

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

### 3.5. Baseline: 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. 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 is trained on 3 to 7 examples of each fluent change.

*Actions:* To compute $\mathcal{E}(l_A|A)$, we calculate pose features from the relative locations of each joint of the hu-

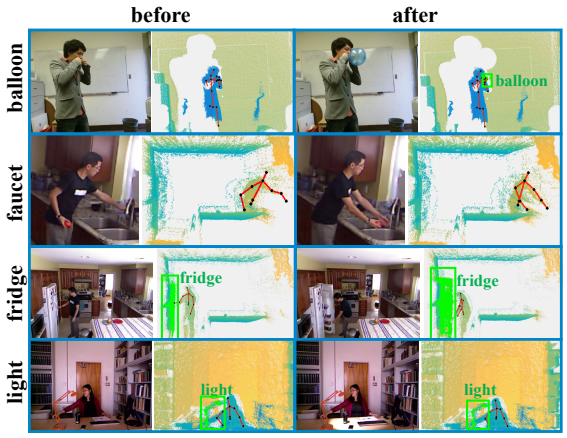


Figure 7. Human poses and depth images (before and after a fluent change) for actions as captured by the Kinect, together with sample frames.

man skeleton as detected by the Kinect, shown in Figure 7. To calculate $\mathcal{E}(R)$, we bind the nodes 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 [29]; model parameters were trained with a multi-class SVM. Dynamic programming beam search [27] 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 $k = 1,000,000$. 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 $P\left(l_{t-\tau+1:t}|pg\right)$, assigned according to the highest-scoring parse graph after the beam search.

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.

### 3.6. Results

Bottom-up fluent and action detections in Figure 8 are improved (and clarified) by applying the sequential C-AOG developed in this paper. The action detectors (second and third plots) use pose to detect open/close actions, without distinguishing objects. Using the sequential C-AOG to combine these action detections with those of the microwave fluent (first plot) shows only some should be labeled "opening/closing the microwave".

Figure 9 shows results from detectors and the sequential C-AOG for light and screen fluents. The fluent detectors erroneously detect multiple light and monitor changes as the light turns on (once) and the camera adjusts; the sequential

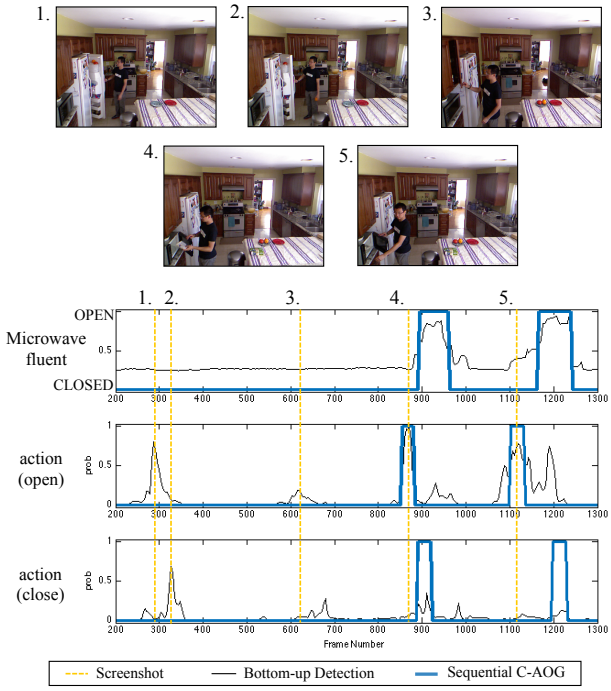


Figure 8. Microwave. Results from fluent and action detectors, superimposed with causal reasoning results. Step functions mark fluent changes–up for turning on, down for turning off.

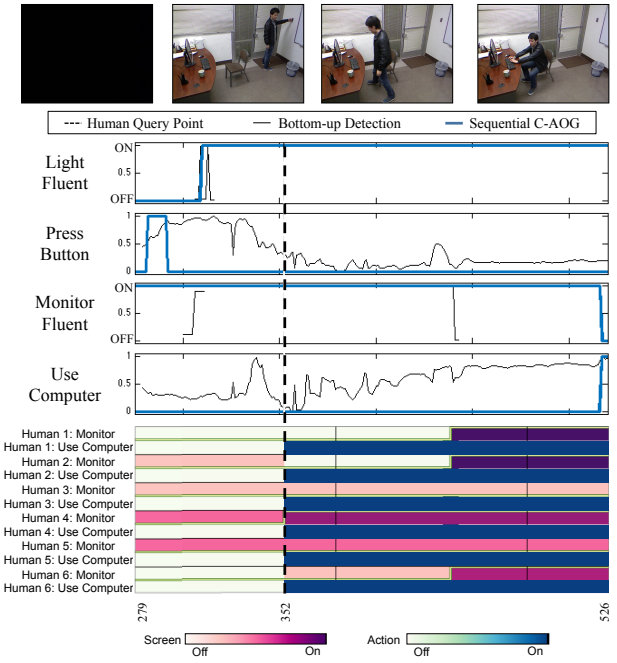


Figure 9. Screen and light fluents with human answers. The dashed line separates the two query points for humans. Human responses varied widely (*e.g.*, Human 1 was certain that the monitor changed from off to on at the second query point, human 3 thought that the screen remained off, and human 4 thought the screen was on).

Table 2. Hit rates for actions and fluents. Cup action is a combination of thirst and waterstream. Italics mark the undetectable fluents.

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

Table 3. Average PR over fluents and actions combined.

| | Precision | Recall |
|---|---|---|
| Detection | 0.29 | 0.31 |
| C-AOG | 0.55 | 0.61 |
| Seq. C-AOG | **0.63** | **0.69** |

C-AOG mostly corrects these.

Table 2 shows performance on individual actions and fluents. In all categories (as well as overall—see Table 3), using the sequential C-AOG to jointly infer actions and fluents outperforms the independent fluent and action detections. Only the door, light, and screen fluents were detectable (undetectable fluents shown with italics). On these examples, action and fluent detections integrate and compete to provide higher overall performance under the sequential C-AOG. For undetectable fluents, the sequential C-AOG combines action detections with the prior causal understanding and consistency over time.

Low detection rates in Table 2 indicate how challenging the dataset is. Nonetheless, "detection" outperforms "noise", and the sequential C-AOG outperforms both.

Table 2 also highlights that humans had difficulty annotating some clips. Categories where "noise" had a non-zero hit rate (*e.g.*, trash) indicate that noise matched at least one human perfectly, or that some humans were completely uncertain for some queries. This underscores the need for multiple annotations and how there is no so-called perfect ground truth.

Finally, Table 2 provides evidence that different annotations were used as ground truth for different computers. Since the thirst fluent is hidden, "detection" and "noise" both consider it equally likely for the agent to be thirsty, not thirsty, or transitioning between the two. However, action detections allowed "detection" to be compared to a different human than "noise".

Table 3 compares overall precision and recall for results obtained using detectors alone, the C-AOG, and the sequential C-AOG developed in this paper. The sequential C-AOG outperforms both raw detectors and the non-sequential C-AOG, highlighting the need to bind the C-AOG over time.

## 4. Discussion and summary

In this paper, we introduced a probability model for the sequential C-AOG, enabling joint inference of hidden fluents and actions 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. Inference of hidden fluents (both as triggers and as effects) provides deeper cognition that can be used to understand, predict, and replicate human actions.

This paper introduced a video dataset to study cause and effect relationships, bridging the gap left by current action datasets.

While the size of this dataset prohibited the use of deep learning detectors (*e.g.*, CNN), it nevertheless reflects what is possible for human knowledge acquisition: humans can learn causal relationships from a small number of examples. Further, "strong" detectors for fluents would have made little difference since most of the fluent misdetctions were occluded.

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, 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. Classifying actions according to their causal effects can provide a meaningful way to resolve ambiguity for judging actions.


### Acknowledgments

This work is supported by NSF IIS 1423305, ONR MURI project N00014-16-1-2007 on Commonsense Reasoning, and DARPA XAI project N66001-17-2-4029.


## References


[1] 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.

[2] 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.

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

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

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

[6] 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.

[7] 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.

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

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

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

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

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

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

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

[15] E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

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

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

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

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

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

[21] 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.

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

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

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

[25] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01, 2012.

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

[27] 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.

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

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

[30] S.-C. Zhu and D. Mumford. *A Stochastic Grammar of Images*. Now Publishers Inc., Hanover, MA, USA, 2006.