Conditioning Style on Substance: Plans for Narrative Observation

Diptanil Chaudhuri[1] Rhema Ike[2] Hazhar Rahmani[3] Dylan A. Shell[1] Aaron T. Becker[2] Jason M. O'Kane[3]

***Abstract*— We consider a robot tasked with observing its environment and later selectively summarizing what it saw as a vivid, structured narrative. The robot interacts with an uncertain environment, modelled as a stochastic process, and must decide what events to pay attention to (substance), and how to best make its recording (style) for later compilation of its summary. If carrying a video camera, for example, it must decide where to be, what to aim the camera at, and which stylistic selections, like the focus and level of zoom, are most suitable. This paper examines planning algorithms that help the robot predict events that (1) will likely occur; (2) would be useful in telling a tale; and (3) may be hewed to cohere stylistically. The third factor, a time-extended requirement, is entirely neglected in earlier, simpler work. With formulations based on underlying Markov Decision Processes, we compare two algorithms: a monolithic planner that jointly plans over events and style pairs and a decoupled approach that prescribes style conditioned on events. The decoupled approach is seen to be effective and much faster to compute, suggesting that computational expediency justifies the separation of substance from style. Finally, we also report on our hardware implementation.**

I. Introduction

Becky and Alice, being excited to participate in the Boston marathon, hire an autonomous robot to produce a custom video of their race. Afterward, the robot will assemble a video clip from events it recorded, in order of occurrence, to tell the tale of their day. The video might show them neck-and-neck, with one crossing the finish line just moments before the other; or perhaps they were widely separated and, while Alice was sprinting past Boston College, Becky was crossing the Johnny Kelley statue. Beyond the mechanics of autonomously navigating, tracking, and shooting video, the robot needs to be strategic about what it tries to capture. Various events will be occurring simultaneously at distinct places and it is possible that events might fit multiple narrative arcs. Clearly the form of the final story depends on how the day actually unfolded, with both predictable structure (start–middle–finish) but also unexpected, serendipitous detail.

Deciding which events to capture was the subject of our earlier, initial foray into this problem [1]. But merely concatenating a sequence of clips, one for each event, gives a rather poor result. Indeed, videography is a sophisticated craft involving wide set of cinematic choices that include framing and positioning, camera focus and depth of field, filters and motion. These choices are complex and constrained (it may only be physically feasible to place cameras in some few positions); the choices have semantic consequences and depend on the subject and scene (e.g., reinforcing the action, or portraying a contrast for rhetorical effect). Finally, the choices are made within the context of a broader flow, as transitions, cuts, and sequencing will alter the overall result. Throughout, we refer to all these aspects under the single umbrella term 'style'. Distinct from style, the substantive elements of narrative are formed by sequences of events.

Events are assumed to be generated by a stateful stochastic process, causally unaffected by the robot's recording activities. Events are treated as atomic items that the robot captures. In the model we propose, as a minimal idealized conception of style, the robot is expected to make some choices about *how* to attempt to capture an event. We encode constraints and suitability by limiting the choices available for certain events. The notions of context and flow will be formalized, drawing inspiration from natural language processing, via a structure we dub a style-gram.

But is style actually different from substance? The contribution of this paper is an examination of this question from an algorithmic standpoint. We define a generalized version of the problem of planning to capture events to fit a narrative structure, allowing occurrences of events to be stochastic given a current state, and introducing our formulation of the style-gram. We describe the necessary extensions of our prior algorithm [1] to apply it to the problem of jointly planning event–style pairs. Further, we introduce a new decoupled method that first makes choices for events, then solves for style. The second step grants extra flexibility by settling on the style choice later, potentially using freshly revealed information. Underlying the solution is a conversion of the problem from a Markov chain evolving in time to another evolving on successful captures.

We have conducted an empirical comparison of the two approaches in simulations of our motivating marathon scenario, examining the impact of style. The decoupled solution is, generally, seen to be a favorable choice: gaining substantial efficiency with attractive performance. Finally, we describe our hardware implementation of a video recording robot which records video using our planning approach.

II. Related Work

Our results build upon prior work [2] which used simple predictions of future events to coordinate robots' efforts in capturing important events, as determined by a specification which used a weighting scheme. More broadly, the problems of selecting effective viewpoints [3]–[5], and of active perception generally [6]–[8], have long histories. The objective underlying those lines of work is informativeness and usually in scenes which are understood to be static.

[1]Dept. of Computer Science and Engineering, Texas A&M University
[2]Dept. of Electrical and Computer Engineering, University of Houston
[3]Dept. of Computer Science and Engineering, University of South Carolina

This material is based upon work supported by the National Science Foundation under Grants IIS-1849303, IIS-1849249, and IIS-1849291.

When the captured observations will be post-processed to provide a summarization for human eyes, the closest work is that of [9], [10], who attack the *vacation snapshot problem* via an online algorithm to collect extremum samples.

Compared with this existing body of work, in our problem the precise temporal ordering in which observations are made matters more. Importance of sequence and temporal flow is central to the computational literature on stories and narrative structure [11]–[13]; that work inspires our notion of an automaton to specify stories. Also notable is the work of Yu and LaValle [14], which focuses on (passively) processing sequences of observations, determining if they fit a story.

Much synergistic work focuses on the taxonomy of camera shots achievable by drones and how to plan collision-free sequences of these shots [15], [16]. We focus on the orthogonal question of collecting footage to tell a desired story. More generally, when the question is not on which observations to make, but rather how to process what is already available, the area of video summarization has several methods [17]–[23]. Other research seeks to generate commentary [24], [25] or to produce narrative text *de novo* [26]–[34].

III. Problem Definition

Compared to most robotic planning problems, uncommonly many structures need to be defined in preparation for the formal problem statement.

A. The World: Event Model

In our scenario, the elements to capture are atomic events, whose occurrence is assumed to be structured as such:

Definition 1 (Event Model (EM)): An event model is a 5-tuple $\mathcal{M} = (W, \mathbb{E}, \tau_e, w_0, g)$, in which (1) W is a state space; (2) $\mathbb{E}$ is the set of all possible events; (3) $\tau_e : W \times W \to [0,1]$ is the transition probability function, such that $\forall w \in W$, $\sum_{w' \in W} \tau_e(w, w') = 1$; (4) $w_0 \in W$ is the initial state; (5) $g : W \times \mathbb{E} \to [0,1]$ is an occurrence labeling function, such that for any state $w \in W$ and an event $e \in \mathbb{E}$, $g(w, e)$ is the probability that e occurs, or 'goes on', in the state w. We assume that $\forall e \in \mathbb{E}$, $g(w_0, e) = 0$.

Starting with $w(0) = w_0$, an EM transitions from state to state in accordance with the probabilities $\tau_e(w(t), w(t+1))$, as time t progresses, in the conventional way for Markov chains. As $\mathcal{M}$ enters state $w(t)$, each event $e \in \mathbb{E}$ either happens or does not, determined independently for each with probability $g(w(t), e)$.

B. Narratives: Story Automaton

As the world evolves and events happen, the robot will attempt to capture events and produce a story portraying what occurred. The story is a subsequence of captured events selected to match a specification of suitable stories. These are given in the form of an automaton [35]:

Definition 2 (Story Automaton): A story automaton is a deterministic finite automaton $\mathbb{A} = (S_A, \mathbb{E}, \tau_A, a_0, F_A)$ with: (1) S_A a nonempty finite set of states; (2) $\mathbb{E}$, its alphabet, a set of all possible events; (3) $\tau_A : S_A \times \mathbb{E} \to S_A$ its transition function; (4) $a_0 \in S_A$, the initial state; and (5) $F_A \subseteq S_A$, the set of final (accepting) states.

Let $\mathcal{L}(\mathbb{A})$ denote the set of event sequences accepted by $\mathbb{A}$, i.e., those sequences reaching some element of F_A, starting at a_0, and after tracing transitions via τ_A. Also, by $\mathcal{L}_{\text{pre}}(\mathbb{A})$ denote those sequences in $\mathcal{L}(\mathbb{A})$ containing no proper prefixes that are themselves in $\mathcal{L}(\mathbb{A})$.

C. Style: Constraints and Sequential Structure

Each time the robot attempts to capture an event, it must choose values for the various parameters that influence the videographic details of the recording. Let Γ be the set of all possible parameter values, or *styles*, available to the robot. Physical hardware and positioning constraints will typically mean that not all events can be captured in all possible styles. Let the *style catalogue*, $\kappa : \mathbb{E} \to 2^{\Gamma}$, map each event to the set of styles that may be used in capturing it. The style catalogue, despite being simple, impels the robot to plan ahead: in seeking to optimize style sequence properties, the style catalogue —as a constraint— binds the robot's choices about styles to decisions it makes about events as well.

Next, we introduce a measure for the efficacy of a sequence of style choices. Watching film with high production value, the choices made for consecutive shots possesses a temporal structure having a flow, which helps establish or reinforce a cinematic style. Taking inspiration from bi-grams, tri-grams, and N-grams as statistical models in natural language processing [36], we formalize this as follows:

Definition 3 (Style-gram): For $k \in \mathbb{Z}_+$, a k-order style-gram is a triple $\mathbb{S} = (\Gamma, \sigma, \omega)$, such that: (1) Γ is the set of all available styles; (2) $\sigma \in \Gamma$ is a special 'empty' symbol; (3) $\omega : \Gamma^{k-1} \times \Gamma \to [0,1]$ is the *efficacy*, so, for each $s_1 s_2 \dots s_{k-1} \in \Gamma^{k-1}$ and $s' \in \Gamma$, it holds that $\sum_{s' \in \Gamma} \omega(s_1 s_2 \dots s_{k-1}, s') = 1$.

The intuition is that, for some cinematic style, a style-gram essentially encodes the probability of a shot with style $s \in \Gamma$ conditioned on the $k-1$ immediately preceding style choices. One imagines that the oeuvres of Alfred Hitchcock, or Quentin Tarantino or Spike Jonze, might be summarized by style-grams that are quite different. Of course, much detail and many various artistic factors are discarded by such a model. But, much as in natural language processing, the approach offers a valuable approximation for various more complex aspects and affords practical advantages: (1) the model can be induced from representative corpora; (2) via k, the range of temporal correlation is parametrizable; (3) it enables direct incorporation into planning considerations.

To combine the style-gram with other elements of our formulation, where Markov assumptions mean that states are sufficient statistics, we flatten the style-gram into a graph:

Definition 4 (Style Graph): For a k-order style-gram $\mathbb{S} = (\Gamma, \sigma, \omega)$, its associated *style graph* is the weighted directed graph $\mathcal{M}_S = (S_S, C, \omega_S)$, where

- $S_S \subseteq \Gamma^{k-1}$ is the set of states, with $\sigma^{k-1} = \underbrace{\sigma\sigma\dots\sigma}_{k-1}$ being the initial state;
- $C \subseteq S_S \times S_S$ is the set of edges such that: $\forall s_k \in \Gamma$, $(s_1 s_2 \dots s_{k-2} s_{k-1}\ ,\ s_2 s_3 \dots s_{k-1} s_k) \in C$;
- $\omega_S : C \to [0,1]$ is the edge weight constructed from ω, $\omega_S(s_1 \dots s_{k-1}\ ,\ s_2 \dots s_k) = \omega(s_1 \dots s_{k-1}, s_k)$.

D. Connecting the pieces: the robot and its capture choices

Up until time t, the robot keeps the sequence of events that it has recorded as ξ_t, the sequence of styles that the robot used to record the event sequence as ζ_t, along with the (unique) story automaton state a_t reached by tracing those events on $\mathbb{A}$. Just before $t+1$ commences, the robot predicts a single event $e_{t+1} \in \mathbb{E}$ that it will attempt to capture. Additionally, the robot picks an appropriate style $s \in \kappa(e_{t+1})$ for its capture. If the prediction was correct and the event indeed happens, the capture is successfully made and in the associated style. If the prediction was incorrect (that is, if the event does not occur), then nothing is captured. The sequence of captured events ξ_t, the sequence of styles ζ_t, and the story automata state a_t, are updated, all as follows:

$$\xi_{t+1} = \begin{cases} \xi_t e_{t+1} & \text{if } e_{t+1} \text{ was captured in state } w_{t+1} \\ \xi_t & \text{otherwise;} \end{cases}$$

$$\zeta_{t+1} = \begin{cases} \zeta_t s & \text{if event } e_{t+1} \text{ was captured with style } s \\ \zeta_t & \text{otherwise;} \end{cases}$$

$$a_{t+1} = \begin{cases} \tau_A(a_t, e_{t+1}) & \text{if } e_{t+1} \text{ was captured in } w_{t+1} \\ a_t & \text{otherwise.} \end{cases} \quad (1)$$

Initially, $\xi_0 = \epsilon$, the empty sequence, and $\zeta_0 = \sigma^{k-1}$. When $a_t \in F_A$, the robot terminates its execution.

Note that the robot traces *all* events captured, stopping when that sequence is in $\mathcal{L}(\mathbb{A})$; One might ask about selection of sub-sequences of events. This seeming shortcoming in (1), in fact is not one. As examined in our earlier work [1], edits to the captured sequence such as selecting a subsequence (and other sorts of edits as well) can be encoded by mutating the story automaton, producing a new one. Hence, we will assume without losing generality, that whatever post-production steps are permissible, have already been expressed in $\mathbb{A}$.

E. Capture Criterion and Optimization Problem

For style graph $\mathcal{M}_S$, we define *sequence efficacy* via function $v_S : \Gamma^{\mathbb{Z}_+} \to \mathbb{R}$, so that for $\zeta = s_1 s_2 \dots s_{|\zeta|}$,

$$\bar{v}_S(\zeta) = \prod_{i=k}^{|\zeta|} \omega_S(s_{i-k+1} \dots s_{i-1}, s_{i-k+2} \dots s_i).$$

Further, we define *accepted sequence efficacy* for a pair of sequences of events ξ and styles ζ, and automata $\mathbb{A}$ as

$$v_S(\xi, \zeta) = \begin{cases} \bar{v}_S(\zeta) & \text{if } |\zeta| = |\xi| \text{ and } \xi \in \mathcal{L}_{\text{pre}}(\mathbb{A}), \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Now, we use (2) as an optimization criterion.

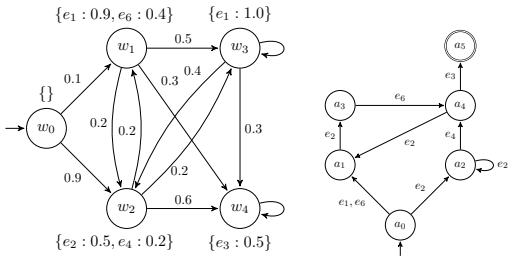

Fig. 1: Example: (a) Event Model, and (b) Story Automata.

Optimization Problem: Styled Video Capture (SVC)

Given: Event model $\mathcal{M} = (W, \mathbb{E}, \tau_e, w_0, g)$ over event set $\mathbb{E}$, a DFA $\mathbb{A} = (S_A, \mathbb{E}, \tau_A, a_0, F_A)$, a set of available styles Γ, a style-graph $\mathcal{M}_S = (S_S, C, \omega_S)$, and a style catalogue $\kappa : \mathbb{E} \to 2^\Gamma$, and a finite horizon $N \in \mathbb{Z}_+$.

Output: Some prescription for events and styles to capture so the expected accepted sequence efficacy $\mathbf{E}_N[v_S(\xi, \zeta)]$ is maximal, expectation being taken over sequences ξ and ζ arising from at most N opportunities to capture an event.

One can anticipate that, unless you have an exceptionally lucky robot, generally $|\xi|$ will be less than N.

For SVC to be formally defined, detail of what is meant by 'prescription' must be determined. Different choices lead to different solutions, in an interesting way.

IV. Optimization: Joint Solution

A quite natural choice, perhaps one already anticipated by the reader, is for the robot's predictions to be governed by a *capture policy*, $\pi_c : W \times S_A \times \Gamma^{k-1} \to \Delta(\mathbb{E} \times \Gamma)$, that uses the state of the world (event model), the story captured so far (story automaton state), and history of recent styles (encapsulated as a state of the style graph) to select a pair comprising an event and a style. ($\Delta(X)$ denotes the set of probability distributions over X.) A conceptually straightforward solution approach is to search over a joint action space, with choices comprising such pairs. This yields a Markov Decision process [37]:

Definition 5 (Monolithic MDP): For an SVC problem, construct MDP $\mathbb{M}_{\text{mon}} = (X_E, x_0, A_E, P_E, R_E)$, where

- $X_E \subseteq W \times S_A \times S_S$ is the finite state space;
- $x_0 = (w_0, a_0, \sigma^{k-1})$ is the initial state;
- $A_E = \mathbb{E} \times \Gamma$ is the action space comprising pairs of events and sanctioned styles: $(e, s) \in A_E$ iff $s \in \kappa(e)$;
- $P_E : X_E \times A_E \times X_E \to [0, 1]$ gives transition probabilities for $(w_i, a_i, \zeta_i), (w_j, a_j, \zeta_j) \in X_E$, $(e, s_k) \in A_E$, with $\zeta_i = (s_1 s_2 \dots s_{k-1})$, $\zeta_j = (s_2 \dots s_{k-1} s_k)$,

$$P_E((w_i, a_i, \zeta_i), (e, s_k), (w_j, a_j, \zeta_j)) = \begin{cases} \tau_e(w_i, w_j) \cdot g(w_j, e) & \text{if } a_j = \tau_A(a_i, e) \text{ and } g(w_j, e) > 0, \\ \tau_e(w_i, w_j) \cdot (1 - g(w_j, e)) & \text{if } a_j = a_i, \\ 0 & \text{otherwise.} \end{cases}$$

- $R_E : X_E \times A_E \times X_E \to \mathbb{R}$ is a reward function such that for $(w_i, a_i, \zeta_i), (w_j, a_j, \zeta_j) \in X_E$, $(e, s_k) \in A_E$, with $\zeta_i = (s_1 s_2 \dots s_{k-1})$, $\zeta_j = (s_2 \dots s_{k-1} s_k)$, we have

$$R_E((w_i, a_i, \zeta_i), (e, s_k), (w_j, a_j, \zeta_j)) = \log \omega_S(\zeta_i, s_k),$$

and all other inputs $R_E(\cdot, \cdot, \cdot)$ takes some constant value r_- with $r_- < \min_{\zeta, s} \log \omega_S(\zeta, s)$.

An optimal policy $\pi^* : X_E \to \Delta(A)$ for this MDP provides a capture policy. Such a policy can be obtained using standard finite-horizon solution techniques; a deterministic policy may be sought, but it is not strictly required.

V. Optimization: Decoupled Solution

Though, in the problem, the choice of event and style are coupled via κ, one might posit that the separation of substance from style is typically quite clean and often useful. Thus, we might approach the problem by decomposing a capture policy into two functions, one to decide events (π_e), another for styles (π_s). Earlier work [1] can compute an event policy of the form $\pi_e : W \times S_A \to \Delta(\mathbb{E})$ efficiently. Then, given those event choices, one might ask the restricted question of how to pick a suitable style. This decomposition, however, also spurs thoughts about new opportunities.

Capturing an event often entails a large-scale activity, needing time to execute—e.g., moving into position to film Becky passing a statue. For this reason the model has the robot predict what will occur, rather than merely discovering what is occurring and then quickly trying to capture it. In contrast, style choices are smaller and more local, so requiring predictions seems less necessary for style choices.

Suppose that after time t, the robot predicts event e_{t+1} might occur, but delays committing to a style for it. As the robot begins executing actions to capture e_{t+1}, the world evolves to w_{t+1}. Suppose that during the course of this execution, the robot learns w_{t+1}. So long as $g(w_{t+1}, e_{t+1}) > 0$, e_{t+1} can still occur and knowing w_{t+1} helps inform the choice of style. For instance, the guess that e_{t+1} will happen may turn out to be true, though perhaps the robot expected the event to happen in w_{t+1}, but the world actually evolved to w'_{t+1} instead. When the events that may occur subsequent to w'_{t+1} differ from those of w_{t+1}, a markedly different choice of style may be warranted. To make such late-breaking style selections, we compute a π_s that uses the state of the world, current progress in the story, and the last $k-1$ styles to make a conditional style selection. This conditional style selection is a function from $W \times \mathbb{E} \to \Gamma$, where the first input would now be w_{t+1}, i.e., the newly realized world state. In other words, we solve a planning problem for $\pi_s(w_t, a_t, (s_{t-k+1} \dots s_t))$ whose output itself can be seen as a sort of local policy.

A. Joint Evolution of the World and Story

A source of complexity in thinking about the robot's interaction with its world is that it only progresses toward a story when it guesses events correctly. With a π_e in hand, one can ignore the detail of the robot making guesses, and model instead the (stochastic) progression of successful event captures. We do this via the *Time/Event Graph (TEG)* which, in essence, abstracts away the time-based evolution, for a progression based on story automaton transitions.

Definition 6 (Time/Event Graph): For event model $\mathcal{M} = (W, \mathbb{E}, \tau_e, w_0, g)$, automaton $\mathbb{A} = (S_A, \mathbb{E}, \tau_A, a_0, F_A)$, and given event policy $\pi_e : W \times S_A \to \Delta(\mathbb{E})$, construct $\mathcal{G}_{\pi_e} = (V, v_0, \tau_S, \tau_U, \omega_G)$, where

- $V \subseteq W \times S_A$ are the vertices of the graph;
- $v_0 = (w_0, a_0)$ is the starting vertex;
- $\tau_S \subseteq V \times \mathbb{E} \times V$, are the successful transitions, such that $((w_i, a_i), e, (w_j, a_j)) \in \tau_S$ iff $\tau_e(w_i, w_j) > 0$, $e \sim \pi_e(w_i, a_i)$, $g(w_j, e) > 0$, and $\tau_A(a_i, e) = a_j$;
- $\tau_U \subseteq V \times \mathbb{E} \times V$, are the unsuccessful transitions, such that $((w_i, a_i), e, (w_j, a_j)) \in \tau_U$ iff $\tau_e(w_i, w_j) > 0$, $e \sim \pi_e(w_i, a_i)$, and $\tau_A(a_i, e) \neq a_j$;
- $\omega_G : \tau_S \cup \tau_U \to [0, 1]$ is the edge weight function, such that for $((w_i, a_i), e, (w_j, a_j)) \in \tau_S \cup \tau_U$,

$$\omega_G((w_i, a_i), e, (w_j, a_j)) = \begin{cases} \pi_e(w_i, a_i)(e) \cdot \tau_e(w_i, w_j) \cdot g(w_j, e) & \text{if } ((w_i, a_i), e, (w_j, a_j)) \in \tau_S, \\ \pi_e(w_i, a_i)(e) \cdot \tau_e(w_i, w_j) \cdot (1 - g(w_j, e)) & \text{if } ((w_i, a_i), e, (w_j, a_j)) \in \tau_U \text{ and } g(w_j, e) > 0, \\ \pi_e(w_i, a_i)(e) \cdot \tau_e(w_i, w_j) & \text{if } ((w_i, a_i), e, (w_j, a_j)) \in \tau_U \text{ and } g(w_j, e) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Given a state $(w_j, a_j) \in V$, we call the state *successful* if there is at least one state $(w_i, a_i) \in V$, and $e \in \mathbb{E}$, such that $((w_i, a_i), e, (w_j, a_j)) \in \tau_S$.

Figs. 1 and 2 provide an example construction. Examining Fig. 2, one sees that it is structured: each block corresponds to a state in the story automaton (compare to Fig. 1(b)). The unsuccessful transitions revisit states within the same block, while successful transitions shift from one block to another.

To find the probability of successfully capturing an event, we compute the probability of all possible sequences of unsuccessful transitions which then lead to a successful transition and capture. Using this approach, we reduce the TEG to a new graph containing only successful transitions, along with the probability of successfully capturing events. The graph thus created encodes the joint progress of both the event model and the story automata together, so we call it the *World Story Joint Progress Graph (JPG)*.

Definition 7 (World Story Joint Progress Graph): For TEG $\mathcal{G}_{\pi_e} = (V, v_0, \tau_S, \tau_U, \omega_G)$, we construct the JPG $\mathcal{M}_{J,\pi_e} = (S_J, v_0, \mathbb{E}, \tau_J, \omega_J)$, where

- $S_J \subseteq V$, are the vertices, with v_0 the initial vertex;
- $\mathbb{E}$ is the set of all possible events;

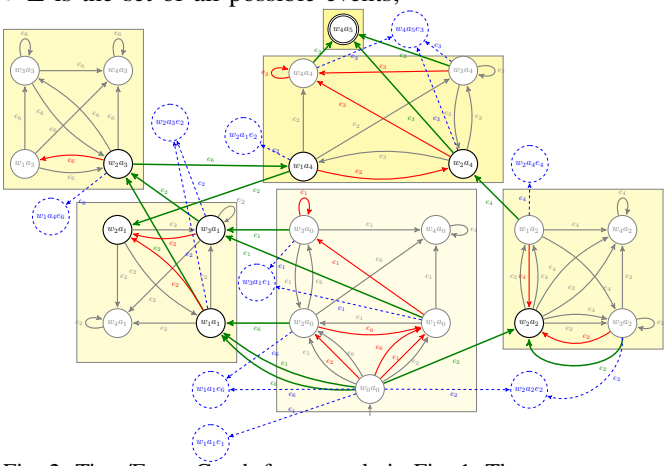

Fig. 2: Time/Event Graph for example in Fig. 1. The green arrows denote the successful transitions. Grey arrows denote unsuccessful transitions with $g(w, e) = 0$, while red arrows denote the unsuccessful transitions with $g(w, e) > 0$. Edge weights have been omitted to improve clarity. (The dashed blue elements that have been superimposed are not part of the TEG, but are an augmentation used to compute ω_J values for the JPG.)

- $\tau_J \subseteq S_J \times \mathbb{E} \times S_J$ are the successful transitions such that $((w_i, a_i), e, (w_j, a_j)) \in \tau_J$ iff $g(w_j, e) > 0$ and $\tau_A(a_i, e) = a_j$;
- $\omega_J : \tau_J \to [0, 1]$ is the probability of successful capture, such that for each $((w_i, a_i), e, (w_j, \tau_A(a_i, e))) \in \tau_J$, $\omega_J((w_i, a_i), e, (w_j, \tau_A(a_i, e)))$ is the probability that when the robot is at (w_i, a_i) the next event it successfully captures is e and the world arrives in state w_j.

The probability of capture, ω_J, is calculated via another Markov chain that is constructed by augmenting the TEG with V', a set of additional absorbing states, shown in dashed blue in Fig. 2. An absorbing state is added for each successful state, along with its incoming event. For example, in the figure, for the successful state (w_1, a_1) we add two absorbing states (w_1, a_1, e_1) and (w_1, a_1, e_6) to V'. The edge weight function $\omega_{G'} : V \times (V \cup V') \to [0, 1]$ is defined as:

– For a successful transition $((w, a), e, (w', a')) \in \tau_S$, $\omega_{G'}((w, a), (w', a', e)) = \omega_G((w, a), e, (w', a'))$.
– For unsuccessful transitions $((w, a), e, (w', a')) \in \tau_U$, $\omega_{G'}((w, a), (w', a')) = \sum_{e \in \mathbb{E}} \omega_G((w, a), e, (w', a'))$.

Markov chain $\mathcal{M}_{G'} = (V \cup V', \omega_{G'})$, with states $V \cup V'$ and transitional probabilities $\omega_{G'}$, enable calculation of absorbing probabilities (see [38, Appendix A]). These values define ω_J.

B. Planning

Solving SVC with event and style decoupling, needs to produce a π_s. To do this we construct a Markov decision process, called the *Style Planning Automaton (SPA)*:

Definition 8 (Style Planning Automaton): Given JPG $\mathcal{M}_{J,\pi_e} = (S_J, v_0, \mathbb{E}, \tau_J, \omega_J)$, events $\mathbb{E}$, available styles Γ, and style graph $\mathcal{M}_S = (S_S, C, \omega_S)$, construct $\mathbb{M}_S = \mathcal{M}_J \times \mathcal{M}_S$, as a tuple $\mathbb{M}_S = (S_{J\times S}, j_0, A, P, R)$,

- $S_{J\times S} \subseteq S_J \times S_S$ is a finite set of states, encapsulating a state in the joint graph, and styles of recent captures;
- An initial state $j_0 = (w_0, a_0, \sigma^{k-1})$;
- The set of actions A contains elements, each indicating the next style to be chosen. Specifically each $a \in A$ is a function prescribing which style to employ, given the attempted capture of some particular event and the current world state is observed; So a is a function that maps $w_k \in W$ and $e_\ell \in \mathbb{E}$ to a style permissible for e_ℓ:
$$W \times \mathbb{E} \ni (w_k, e_\ell) \xmapsto{a} s \in \kappa(e_\ell).$$
- $P : S_{J\times S} \times \mathbb{E} \times S_{J\times S} \to \mathbb{R}$ is the transition probability function such that for $(w_i, a_i, \zeta_i), (w_j, a_j, \zeta_j) \in S_{J\times S}$ and e, we have, when $((w_i, a_i), e, (w_j, a_j)) \in \tau_J$, $P((w_i, a_i, \zeta_i), (e, s_k), (w_j, a_j, \zeta_j)) = \omega_J((w_i, a_i), e, (w_j, a_j))$ and, for all other inputs, $P(\cdot, \cdot, \cdot)$ is 0.
- $R : S_{J\times S} \times \Gamma \times S_{J\times S} \to \mathbb{R}$ is the reward function such that for $(w_i, a_i, \zeta_i), (w_j, a_j, \zeta_j) \in S_{J\times S}$, and $s_k \in \Gamma$, where $\zeta_i = (s_1 s_2 \dots s_{k-1})$, and $\zeta_j = (s_2 \dots s_{k-1} s_k)$, $R((w_i, a_i, \zeta_i), s_k, (w_j, a_j, \zeta_j)) = \log \omega_S(\zeta_i, s_k)$, and for other inputs takes some constant value r_- with $r_- < \min_{\zeta, s} \log \omega_S(\zeta, s)$.

For this MDP with horizon N, a policy can be obtained using standard finite-horizon solution techniques. An optimal deterministic policy $\pi_s^* : S_{J\times S} \to A$ serves as a style policy.

Is the loss of precision incurred by planning for events (in the absence of style considerations), and then prescribing styles afterward, offset by the extra flexibility obtained by delaying the style selection? And, given that the output of the decoupled problem is more complex than the joint one, is the cost to compute favourable compared to the monolithic solution? Next, we examine these questions empirically.

VI. Case Studies

Consider an athletic race between runners A and B. The events of interest are: e_A, A is running; e_B, B is running; e_{AB}, A is overtaking B; e_{BA}, B is overtaking A; e_{AF}, A is crossing the finish line; e_{BF}, B is crossing the finish line; e_E, the race is ended. We want to capture those events from five relative poses, namely: front, rear, left, right, and front-left side, to represent which we respectively use styles s_f, s_b, s_l, s_r, and s_{fl}. We model the contest with the event model in Fig. 3c. The style catalogue and style-gram for the model appear in Fig. 3b. Note that $k = 2$. Each row of the style-gram gives a preference over the styles reflecting choices for possible positions to which it might move. For example, based on the second row, if the robot is currently capturing an event from the front, then the right (rear) has the highest (lowest) desirability for next capturing an event.

We now consider three variations on this scenario. In the first variation, the desired story is specified by the story automaton in Fig. 3e. Accordingly, we are interested in only two stories, $e_A e_{AB} e_E$ and $e_A e_{BA} e_E$. We computed optimal policies using both the monolithic and the decoupled approaches, and we used each computed policy in 100 simulations. Using the decoupled approach, the robot was able to capture $e_A e_{BA} e_E$ with style sequence $s_f s_r s_{fl}$ and $e_A e_{AB} e_E$ with style sequence $s_b s_l s_{fl}$, each one in 55 and 45 simulations, respectively. Both these two style sequences have an efficacy value of 0.1, which is optimal. Using the monolithic approach, the robot captured the style sequence $s_f s_l s_{fl}$ for story $e_A e_{AB} e_E$ and $s_f s_r s_{fl}$ for $e_A e_{BA} e_E$, which have efficacy values 0.05 and 0.1 respectively. The average style sequence efficacy values for those 100 simulations was 0.0755. This means that the decoupled approach did better than the monolithic approach in capturing stories with better qualities in terms of style sequence efficacy. This result follows from the fact that in the decoupled approach, the robot has the freedom to choose the style after it observes that the event it predicted is occurring while in the monolithic approach, the robot does not have such a freedom and, in fact, it chooses an event and a style together.

For the second variation, the robot is tasked to capture a story specified by the story automaton in Fig. 3f. For 100 simulations using the decoupled approach: the robot captured either $e_{AB} e_{AF}$ with style sequence $s_l s_l$ or $e_{BA} e_{BF}$ with style sequence $s_r s_l$ in 43 simulations, and captured $e_{BA} e_{BF}$ with style sequence $s_r s_l$ in 57 simulations. The efficacy of these two sequences was 0.0105 and 0.0049, respectively. The monolithic approach captured $s_A s_B s_E$ with style $s_f s_r s_{fl}$, the accepted sequence efficacy of which is 0.1. In this scenario, the monolithic approach yielded a

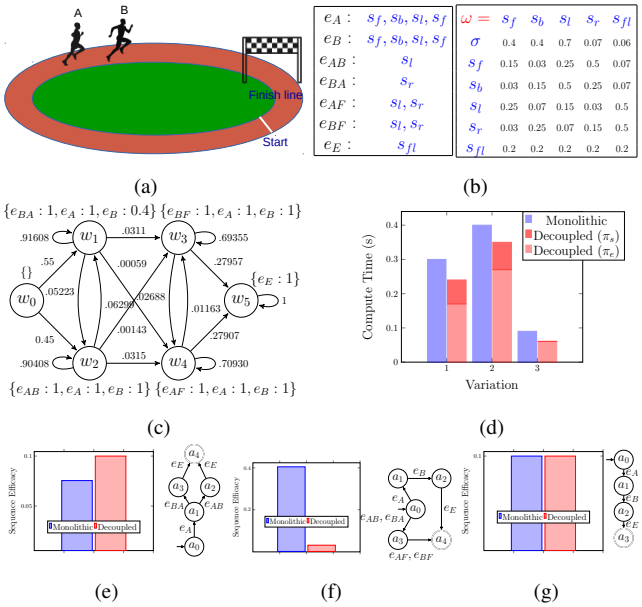

Fig. 3: (a) A race with runners A and B, (b) Style catalogue (left) and style-gram (right) for the event model. (c) An event model for the contest. (d) Computation times for both approaches. (e) Story automaton for the first variation and a plot comparing the accepted sequence efficacies of the decoupled and the monolithic approach, over 100 simulations. (f) Story automaton for the second variation and its plot, and (g) the same for the third variation.

better style sequence. This is justified by the fact that, while the monolithic approach chooses a tuple of event-style that maximizes the excepted value of accepted sequence efficacy, the decoupled approach chooses styles to capture in the next time step, but for events which were chosen (independent of style considerations) to minimize the expected number of steps. In the third variation, the robot captures the sequence in the story automaton in Fig. 3g, Both approaches yielded identical style sequences ($s_f s_r s_{fl}$), having efficacy of 0.1.

A single-threaded implementation was tested on a 2.4GHz Ubuntu 16.04 computer. Figure 3d shows the times to compute the optimal policy for the monolithic and the decoupled approaches. Note that the state space of the monolithic MDP is $W \times S_A \times S_S$. In contrast, the decoupled solution has states $W \times S_A$ for computing the event policy and $S_J \times S_S$ for the style policy. These respective state spaces mean that the decoupled version scales better than the monolithic solution. The decoupled approach was consistently faster than the monolithic approach, with the difference becoming significant when the story automaton and the event model are large, leading to a prohibitively large product automaton.

VII. Robot Implementation

The algorithm was implemented on a mobile robot, shown in Fig. 4. Each runner carries a GPS tracker (Raspberry Pi Zero, mobile hotspot and GPS module) uploads their GPS coordinates of the runner at 1 Hz to a central database. The robot videographer uploads its location at the same rate. An OpenCV human detection algorithm is used to identify runners and a pan-tilt camera mechanism keeps the runners in frame while the robot moves. A custom web application shown in Fig. 5 was designed to keep track of the world

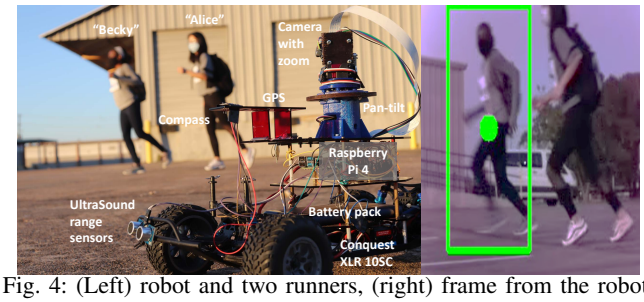

Fig. 4: (Left) robot and two runners, (right) frame from the robot.

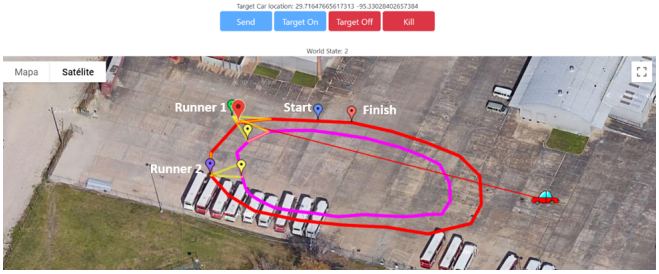

Fig. 5: Screenshot from custom web application. The red line shows the track of runners, the purple line is the robot's path (chosen to film the runners without collisions when filming side shots) and the yellow points show where the robot successfully captured side shots. The satellite image is from the Google Maps API, but during our experiment no busses were in the parking lot.

state and calculate points on the input tracks the robot can maneuver to capture shots of different styles. A computer running the director code, sends styles and events to capture to the database based on the current world state and on whether the previous event was captured or not. The robot queries the database for an event and its corresponding style to capture and it sends back a Boolean of the event being successfully captured in the desired style. The robot has a dictionary mapping styles to the track points relative to the runner. For instance, a front-view shot of runner one will direct the robot to go to the track point directly in front of the runner and a side shot will direct the car to go to the closest point to the runner on the purple track shown in Fig. 5. The robot has a fixed amount of time to get to the specified location. It sends its success status back to the database and based on that, the director algorithm sends the next style and event.

VIII. Conclusion

We formulated the problem of autonomously capturing stylistically sound and narratively coherent sequences of events in uncertain environments. Two approaches were presented and the relative merits of each demonstrated a small simulation case study and implemented on a robot. Future work might examine approaches that allow the event models, story automata, style-grams and catalogues to be induced from data. Some aspects are straightforward: probabilities may be obtained from empirical frequencies, while more structural aspects may be harder to learn. A second direction would be to examine extensions beyond regular languages. For instance, LTL would allow alternative ways to specify stories. Finally, we recently posed one multi-robot version of this problem [39], but other treatments are possible.

REFERENCES

[1] H. Rahmani, D. A. Shell, and J. M. O'Kane, "Planning to chronicle," in *Algorithmic Foundations of Robotics XIV*. Cham: Springer International Publishing, 2021, pp. 277–293.

[2] D. A. Shell, L. Huang, A. T. Becker, and J. M. O'Kane, "Planning coordinated event observation for structured narratives," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7632–7638.

[3] C. I. Connolly, "The determination of next best views," in *International Conference on Robotics and Automation (ICRA)*, vol. 2, 1985, pp. 432–435.

[4] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding Horizon "Next-Best-View" Planner for 3D Exploration," in *International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[5] N. Palomeras, N. Hurtós, E. Vidal, and M. Carreras, "Autonomous exploration of complex underwater environments using a probabilistic next-best-view planner," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1619–1625, 2019.

[6] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[7] Y. Aloimonos, *Active perception*. Lawrence Erlbaum Associates, Inc, 1993.

[8] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[9] Y. Girdhar and G. Dudek, "Efficient on-line data summarization using extremum summaries," in *Proc. IEEE International Conference on Robotics and Automation*, 2012.

[10] Y. Girdhar, P. Giguere, and G. Dudek, "Autonomous adaptive exploration using realtime online spatiotemporal topic modeling," *International Journal of Robotics Research*, vol. 33, no. 4, pp. 645–657, 2014.

[11] V. Y. Propp, *Morphology of the Folktale*. University of Texas Press, 1968, vol. 9.

[12] A. Dundes, "On computers and folk tales," *Western Folklore*, vol. 24, no. 3, pp. 185–189, 1965.

[13] P. Gervás, "Propp's Morphology of the Folk Tale as a Grammar for Generation," in *Workshop on Computational Models of Narrative*, M. A. Finlayson, B. Fisseni, B. Löwe, and J. C. Meister, Eds., vol. 32. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 106–122.

[14] J. Yu and S. M. LaValle, "Story validation and approximate path inference with a sparse network of heterogeneous sensors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4980–4985.

[15] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, "Autonomous uav cinematography: A tutorial and a formalized shot-type taxonomy," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–33, 2019.

[16] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas, "High-level multiple-uav cinematography tools for covering outdoor events," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019.

[17] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[18] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Automatic video summarization by graph modeling," in *Proc. IEEE International Conference on Computer Vision*, 2003.

[19] R. Hong, J. Tang, H.-K. Tan, C.-W. Ngo, S. Yan, and T.-S. Chua, "Beyond search: Event-driven summarization for web videos," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 7, no. 4, p. 35, 2011.

[20] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Proc. European conference on computer vision*. Springer, 2014, pp. 540–555.

[21] L. Feng, Z. Li, Z. Kuang, and W. Zhang, "Extractive video summarizer with memory augmented neural networks," in *Proc. of ACM International Conference on Multimedia*, 2018, p. 976–983.

[22] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden markov models," in *Proc. International Conference on Image Processing*, 2002.

[23] M. H. Kolekar and S. Sengupta, "Event-importance based customized and automatic cricket highlight generation," in *Multimedia and Expo, 2006 IEEE International Conference on*. IEEE, 2006, pp. 1617–1620.

[24] H. Hajishirzi, J. Hockenmaier, E. T. Mueller, and E. Amir, "Reasoning in Robocup Soccer Narratives," in *Proceedings of 27th Conference on Uncertainty in Artificial Intelligence (UAI'11)*, 2011.

[25] S. Rosenthal, S. P. Selvaraj, and M. Veloso, "Verbalization: Narration of autonomous robot experience," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16)*, 2016, pp. 862–868.

[26] M. O. Riedl and R. M. Young, "Narrative Planning: Balancing Plot and Character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217–268, 2010.

[27] N. D. Allen, J. R. Templon, P. S. McNally, L. Birnbaum, and K. J. Hammond, "Statsmonkey: A data-driven sports narrative writer," in *Computational Models of Narrative, Papers from the 2010 AAAI Fall Symposium, Arlington, Virginia, USA, November 11-13, 2010*, 2010.

[28] A. Jhala and R. M. Young, "Cinematic Visual Discourse: Representation, Generation, and Evaluation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, June 2010.

[29] S. Chen, A. M. Smith, A. Jhala, N. Wardrip-Fruin, and M. Mateas, "RoleModel: towards a formal model of dramatic roles for story generation," in *INT3'10: Proceedings of the Intelligent Narrative Technologies III Workshop*, New York, NY, USA, 2010, pp. 1–8.

[30] N. Szilas, M. Axelrad, and U. M. Richle, "Propositions for Innovative Forms of Digital Interactive Storytelling Based on Narrative Theories and Practices," *Transactions on Edutainment*, vol. VII, pp. 161–179, 2012.

[31] H. Yu and M. O. Riedl, "Personalized Interactive Narratives via Sequential Recommendation of Plot Points," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, vol. 6, no. 2, 2014.

[32] A. Amos-Binks, C. Potts, and R. M. Young, "Planning Graphs for Efficient Generation of Desirable Narrative Trajectories," Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies, 2017.

[33] J. Robertson and R. M. Young, "Narrative Mediation as Probabilistic Planning," Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies, 2017.

[34] C. Barot, M. Branon, R. E. Cardona-Rivera, M. Eger, M. Glatz, N. Green, J. Mattice, C. Potts, J. Robertson, M. Shukonobe, L. Tateosian, B. R. Thorne, and R. M. Young, "Bardic: Generating Multimedia Narrative Reports for Game Logs," Working Notes of the AIIDE Workshop on Intelligent Narrative Technologies, 2017.

[35] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed. Addison-Wesley, 2006.

[36] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Prentice Hall, 2008.

[37] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[38] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

[39] D. Chaudhuri, H. Rahmani, D. Shell, and J. M. O'Kane, "Tractable Planning for Coordinated Story Capture: Sequential Stochastic Decoupling," in *Proceedings of International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2021, (to appear).