

ACADEMIA ROMÂNĂ

Școala de Studii Avansate a Academiei Române

Institutul de Matematică „Simion Stoilow”

Rezumatul Tezei de Doctorat

Înțelegerea Procesului de Înțelegere Vizuală: De la Semnale Cerebrale la Reprezentări Multimodale Explicabile

CONDUCĂTOR ȘTIINȚIFIC:
Prof. Dr. Marius Leordeanu

DOCTORAND:
Ing. Nicolae Cudlenco

București, 2026

Cuprins

Rezumat **3**

1 Introducere **4**

2 Cercetări Preliminare: EEG și Înțelegerea Vizuală **7**
2.1 Setul de date 7
2.2 Metodă 8
2.3 Experimente și Rezultate 8
2.4 Discuții și Implicații 9

3 Graful Evenimentelor în Spațiu și Timp **10**
3.1 Modelul GEST 10
3.2 Viziunea: GEST ca Hub Multimodal 10
3.3 Text spre GEST 11
3.4 Validare Inițială 12
3.4.1 Potrivire de Grafuri pentru Similaritatea Textelor 12
3.4.2 Evaluare Umană a Videoclipurilor Generate 13
3.5 Discuții 13

4 GEST-Engine: De la Grafuri de Evenimente la Video Sintetic **14**
4.1 Privire de Ansamblu și Evoluție 14
4.2 Editorul de Lume 15
4.3 Privire de Ansamblu a Arhitecturii 17
4.4 Colectare Multi-Modală a Artefactelor 18
4.5 Generare Procedurală de GEST-uri 19
4.6 Pipeline-ul de Generare a Textului 19
4.7 Orchestrarea Producției 19

5 Generare Agentică Text-GEST **20**
5.1 Introducere 20
5.2 Abordarea LLM cu Etape 20
5.3 Sistemul Agentic cu Apel de Funcții 21

5.4 Evaluare 22
5.5 Sistemul Agentic ca Realizare Completă a GEST 23

6 Evaluare Experimentală **24**
6.1 Setul de date GTASA 24
6.2 Evaluare Umană a Generării Video 25
6.3 Captioning Video 26
6.4 Sondare Spațio-Temporală 26
6.5 Studiu de caz: Punctele Slabe ale VLM-urilor 27
6.6 Discuții 28

7 Concluzii și Direcții Viitoare **29**
7.1 Sinteza Contribuțiilor 29
7.2 Direcții Viitoare 30
7.3 Considerații Finale 31

Bibliografie **32**

Rezumat

În această teză investigăm înțelegerea vizuală din două perspective complementare — modul în care creierul uman procesează informația vizuală și modul în care reprezentările structurate pot reduce diferența dintre vederea computațională și limbajul natural — culminând cu un sistem care generează videoclipuri narative controlabile cu mai mulți actori, însoțite de adnotări spațio-temporale exacte.

Un studiu empiric al semnalelor cerebrale înregistrate în timpul recunoașterii vizuale arată că electrozii frontali, asociați cu procesarea cognitivă de nivel superior, contribuie mai mult la acuratețea clasificării decât electrozii occipitali plasați deasupra cortexului vizual primar și că features EEG sunt complementare features vizuale extrase prin rețele neuronale profunde (recunoașterea crește de la 91% la peste 97%). Pornind de la această observație, propunem Graful Evenimentelor în Spațiu și Timp (GEST), o reprezentare formală care descompune scenele vizuale în evenimente conectate prin relații temporale, spațiale și semantice, validată atât ca măsură de similaritate semantică, cât și ca specificație pentru generarea de video.

Am creat GEST-Engine, o platformă care execută GEST-uri în mod determinist într-un motor de joc 3D, producând videoclipuri cu mai mulți actori și adnotări dense la nivel de cadru — RGB, segmentare, grafuri de relații spațiale calculate per cadru, alinieri temporale eveniment-cadru și descrieri în limbaj natural — fără cost suplimentar de adnotare. Pentru a îmbina coerența narativă cu validitatea simulatorului, propunem un sistem agentic de conversie text-GEST în care un LLM planifică povestea în limbaj natural, iar un backend programatic impune constrângerile prin apeluri de funcții validate, obținând o rată de succes end-to-end de 80% față de 0% pentru o variantă bazată exclusiv pe LLM organizat în etape.

Folosind motorul, am produs corpusul GTASA: 938 de videoclipuri cu mai mulți actori, 892 de milioane de perechi de relații spațiale distribuite pe 1,45 milioane de cadre. Un studiu cu evaluatori umani arată o validitate fizică de 69% și o aliniere semantică de 4,09/5 față de 18%/2,50 (VEO 3.1) și 13%/1,75 (WAN 2.2). Am demonstrat că antrenarea unor VLM-uri pe date GEST este mai benefică , decât atunci când folosim exclusiv VEO sau date reale. Datele noastre a făcut posibilă investigarea reprezentării spațiale interne a modelelor și am descoperit că aceasta este mai bogată la auto-encoders decât la VLM.

Cuvinte-cheie – *înțelegere vizuală, grafuri de evenimente, generare de video*

Capitolul 1

Introducere

Această teză investighează înțelegerea vizuală din două perspective complementare — modul în care creierul uman procesează informația vizuală și modul în care reprezentările structurate pot apropia vederea computațională de limbajul natural.

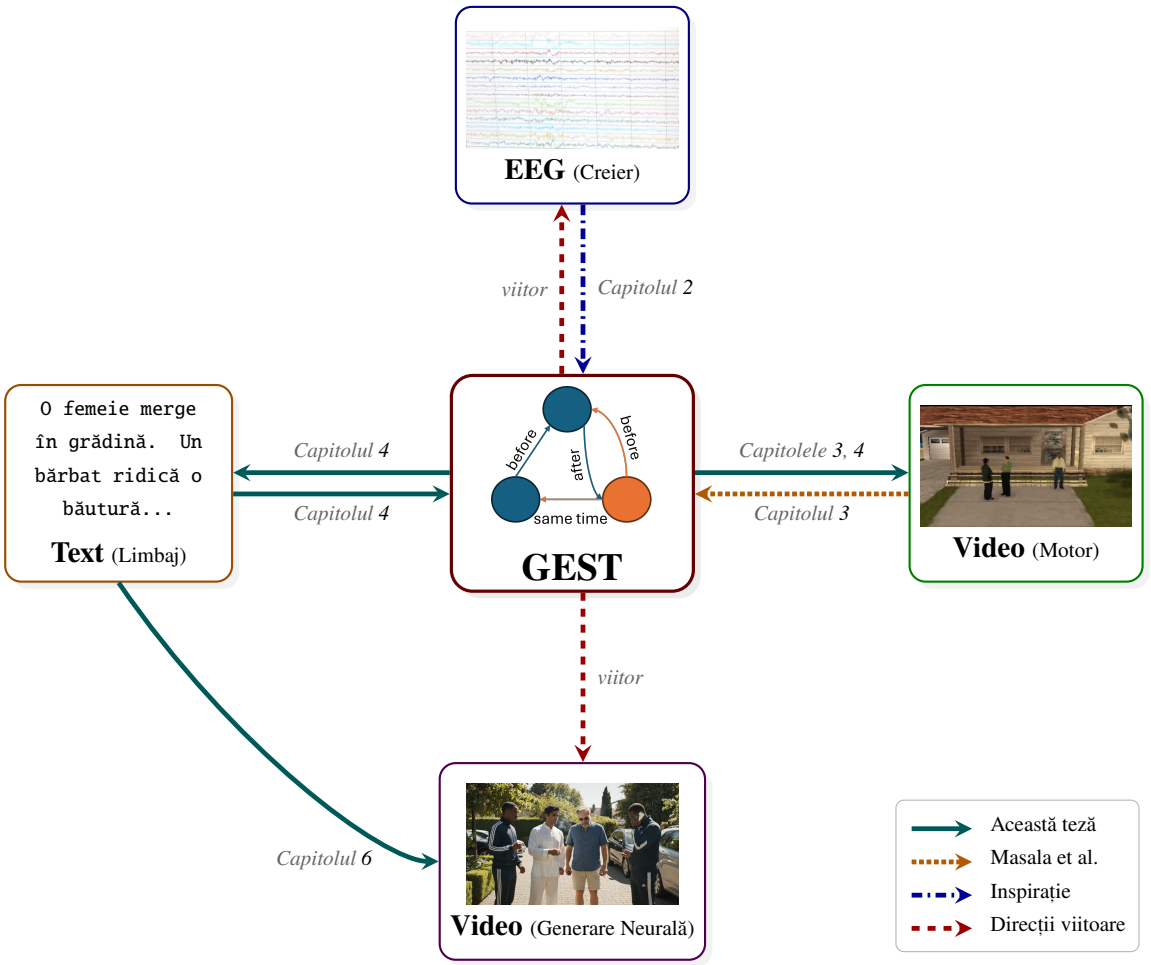


Figura 1.1: **Privire de ansamblu asupra tezei.** GEST este reprezentarea centrală care conectează EEG, limbaj și video. Săgeți continue: contribuțiile acestei teze. Săgeată punct-linie: inspirația din studiul EEG (Capitolul 2). Săgeată punctată: lucrări complementare ale lui Masala et al. [22]. Săgeată întreruptă: direcții viitoare de cercetare.

Argumentul central este că înțelegerea vizuală beneficiază de reprezentări explicite și structurate care fac relațiile dintre entități, evenimente și diferite modalități de reprezentare transparente și calculabile. În loc să ne bazăm exclusiv pe tipare statistice implicite învățate din date brute, propunem ancorarea conținutului vizual într-o structură formală bazată pe grafuri, în care evenimentele sunt noduri, iar relațiile temporale, spațiale și semantice sunt muchii. O astfel de reprezentare permite o generare de video mai fidelă, o antrenare mai eficientă a modelelor și o înțelegere mai profundă a informației pe care sistemele actuale o învață despre lumea vizuală.

Teza este organizată în jurul a cinci întrebări de cercetare (ÎC):

ÎC1: Ce pot dezvălui semnalele cerebrale despre natura înțelegerii vizuale?

ÎC2: Poate o reprezentare explicită bazată pe grafuri să reducă decalajul dintre modalitățile vizuală și lingvistică?

ÎC3: Poate un sistem bazat pe un model al lumii explicit să producă videoclipuri cu mai mulți actori, fidele semantic și cu adnotări spațio-temporale exacte?

ÎC4: Pot agenții bazați pe LLM-uri să genereze grafuri GEST coerente narativ și valide pentru simulator?

ÎC5: Pot datele sintetice structurate să îmbunătățească modelele de înțelegere video și să dezvăluie ce învață acestea despre spațiu și timp?

Tabelul 1.1 pune în corespondență fiecare contribuție cu publicația, întrebarea de cercetare și capitolul aferent. Le rezumăm mai jos.

Tabela 1.1: Sinteza contribuțiilor, a publicațiilor asociate, a întrebărilor de cercetare și a capitolelor.

Contribuție	**Publicație**	**ÎC**	**Capitol**
Semnalele creierului și înțelegerea vizuală	Cudlenco et al., Neurocomputing 2020 [10]	ÎC1	Cap. 2
Reprezentarea GEST	Masala, Cudlenco et al., ICCV 2023 [21]	ÎC2	Cap. 3
Motorul GEST	Cudlenco et al., publicată la ICLR 2026 Workshop on World Models [7]; trimisă spre publicare la ECCV 2026 (arXiv:2604.10385) [9]	ÎC3	Cap. 4
Generarea agentică text-spre-GEST	Cudlenco et al., trimisă spre publicare la CVPR Workshop 2026 (arXiv:2604.10383) [8]	ÎC4	Cap. 5
Validarea experimentală	Cudlenco et al., trimisă spre publicare la ECCV 2026 (arXiv:2604.10385) [9]	ÎC5	Cap. 6

1. **Semnalele creierului și înțelegerea vizuală** [10] (Capitolul 2): un studiu EEG care arată că electrozii frontali asociați procesării cognitive de nivel superior contribuie mai mult la clasificarea vizuală decât electrozii occipitali plasați deasupra cortexului vizual primar și că features EEG sunt complementare features vizuale extrase prin rețele neuronale profunde (91% → 97%).

2. **Reprezentarea GEST** [21] (Capitolul 3): Graful Evenimentelor în Spațiu și Timp, validat ca măsură de similaritate semantică (depășind BLEU, METEOR, ROUGE, SPICE) și ca specificație pentru generarea de video (preferință umană de 87% în raport cu metodele neurale de referință).

3. **Motorul GEST** [7, 9] (Capitolul 4): un sistem complet pentru generarea de videoclipuri cu mai mulți actori, însoțite de adnotări dense la nivel de cadru — RGB, segmentare la nivel de instanță, grafuri de relații spațiale per cadru, alinieri temporale eveniment-cadru și descrieri în limbaj natural — fără cost suplimentar de adnotare. Sistemul cuprinde aproximativ 100.000 de linii de cod în Lua, C++, Python și JavaScript.

4. **Generarea agentică text-spre-GEST** [8] (Capitolul 5): un sistem bazat pe LLM care obține o rată de succes de 80% față de 0% al unei variante bazată exclusiv pe LLM prompting, organizat în etape. Poveștile agentice câștigă 79% dintre comparațiile bazate pe text și 74% dintre cele bazate pe video în raport cu metodele procedurale de referință.

5. **Validarea experimentală** [9] (Capitolul 6): corpusul GTASA (938 de videoclipuri, 892 de milioane de relații spațiale, 1,45 milioane de cadre) și trei direcții de evaluare — evaluare umană (69% validitate față de 18%/13% pentru VEO/WAN), pretraining pentru captioning video (datele GEST depășesc VEO și antrenarea exclusiv pe date reale) și sondare spațială a video-encoders (auto-supervizate > encoders vizuale ale modelelor VLM).

Capitolul 2

Cercetări Preliminare: EEG și Înțelegerea Vizuală

Dovezile din neuroștiințele cognitive arată că înțelegerea vizuală nu este limitată la cortexul vizual primar, ci implică regiuni cognitive de nivel superior distribuite în întregul creier [16, 24]. Înainte de a urmări această direcție pe plan computațional — care conduce la Graful Evenimentelor în Spațiu și Timp (GEST) din Capitolul 3 — am încercat să explorăm mai întâi această întrebare empiric, folosind cea mai directă metodă disponibilă: înregistrarea electroencefalogramelor (EEG) ale unor persoane care privesc imagini.

2.1 Setul de date

Am colectat un set de date EEG nou, conceput special pentru acest studiu. Setul cuprinde înregistrări de la șase voluntari (patru bărbați, două femei), toți absolvenți de studii universitare cu vârste între 25 și 35 de ani. Fiecare subiect a participat la două sesiuni de înregistrare: una pentru datele de antrenare și una pentru datele de testare, separate de o pauză de câteva ore în care subiecții au scos casca și și-au desfășurat activitățile obișnuite. Această separare este caracteristica definitorie a protocolului nostru experimental — a fost concepută pentru a asigura că niciun clasificator antrenat pe aceste date nu poate obține o acuratețe ridicată bazându-se pe tipare de zgomot specifice sesiunii, autocorelații temporale sau starea cognitivă de moment a subiectului.

Setul de stimuli conține șase categorii vizuale, alese în mod deliberat pentru a include atât obiecte, cât și scene naturale: flori, avioane, mașini, parc, malul mării și oraș. Fiecare clasă conține 60 de imagini, împărțite în 40 pentru antrenare și 20 pentru testare, însumând un total de 360 de imagini. Imaginile au fost prezentate secvențial, câte una la fiecare 7 secunde. În sesiunea de antrenare, subiecții au fost instruiți să se concentreze asupra fiecărei imagini și să se gândească activ la concept, în loc să privească pasiv stimulul.

[0] Acest capitol se bazează pe: **Cudlenco**, Popescu, Leordeanu — Neurocomputing 2020 [10].

Tabela 2.1: Perechi de electrozi clasificate după acuratețea de clasificare. Tabel din [10].

Loc	Canale	Acc (%)	Zone cerebrale
1	FC5/FC6	66,21	Lob frontal-central
2	F7/F8	64,14	Lob frontal
3	AF3/AF4	63,02	Lob prefrontal
4	P7/P8	62,94	Lob parietal
5	T7/T8	61,69	Lob temporal
6	F3/F4	59,66	Lob frontal
7	O1/O2	59,25	Occipital — Cortex vizual

2.2 Metodă

Semnalele EEG brute au fost transformate în features în domeniul frecvenței prin filtrare cu wavelet-uri Gabor pe nouă scări, aplicată pentru toate cele 14 canale. Am evaluat trei clasificatori: Regresie Ridge, o rețea neuronală convoluțională (CNN) și LSTM-uri pe mai multe niveluri. Pentru experimentul de fuziune, features EEG (extrase dintr-un LSTM cu două straturi) au fost concatenate cu features vizuale produse de o rețea Inception-v3 antrenată pe imaginile-stimul și transmise printr-un clasificator comun.

2.3 Experimente și Rezultate

Studiul importanței electrozilor. O evaluare sistematică a fiecărei perechi de electrozi (Tabelul 2.1) a arătat că perechea cu cea mai mare acuratețe individuală de clasificare a fost FC5/FC6, plasată deasupra lobului frontal-central — o regiune asociată cu procesarea cognitivă de nivel superior, nu cu percepția vizuală (Figura 2.1). Primele trei perechi (FC5/FC6, F7/F8, AF3/AF4) sunt toate frontale sau prefrontale. Perechea occipitală O1/O2, plasată direct deasupra cortexului vizual primar, s-a clasat ultima. Folosind doar primele trei perechi frontale s-a obținut o acuratețe de 69,39% față de 66,76% atunci când au fost folosite toate canalele. Acest tipar a fost constant la toți cei șase subiecți, întărind concluzia că efectul nu este un artefact al variației individuale, ci reflectă o proprietate reală a modului în care creierul procesează categoriile vizuale.

Compararea modelelor. Regresia Ridge cu filtrare Gabor (29,2%) a depășit CNN-ul (26,5%) și LSTM-ul (25,67%) pe sarcina multiclasă, în concordanță cu regimul cu date limitate, în care modelele mai simple generalizează mai stabil. Toate modelele au depășit baseline-ul aleator de 16,67%.

Fuziunea EEG–vizuală. Combinarea features EEG cu features vizuale profunde a îmbunătățit clasificarea de la 91% la peste 97% în medie pe subiecți. Toți subiecții au înregistrat o îmbunătățire (între 4 și 6 puncte procentuale), confirmând că semnalele cerebrale poartă informație complementară, necaptată de clasificatorii bazați doar pe imagini.

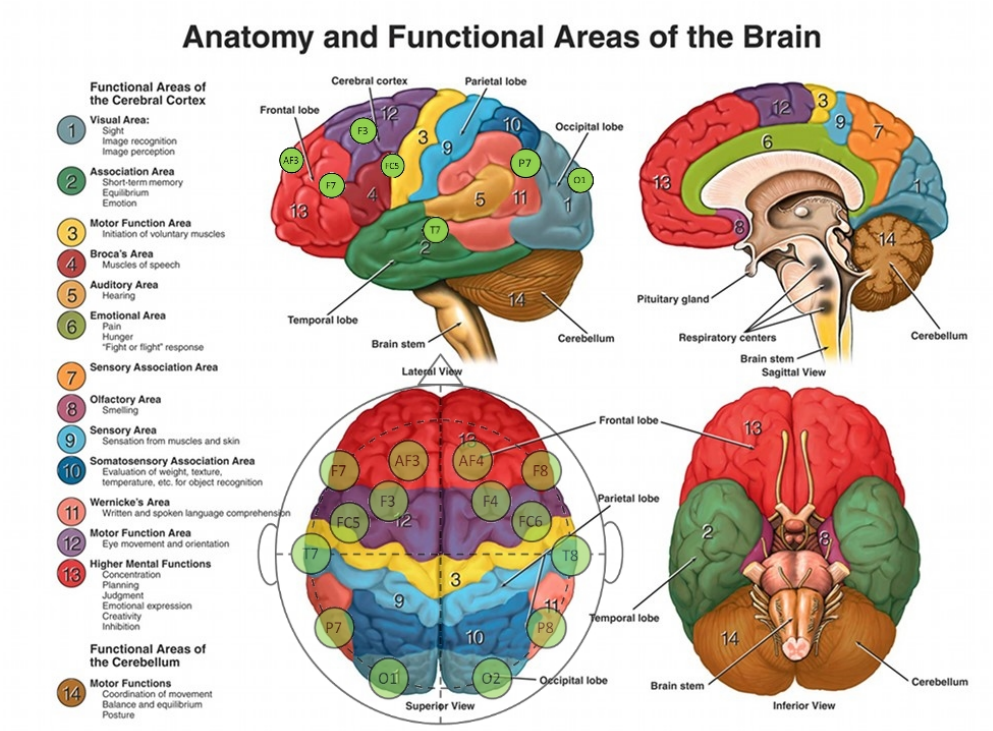


Figura 2.1: **Localizarea electrozilor în raport cu zonele funcționale ale creierului.** Cei mai informativi electrozi pentru clasificarea vizuală (FC5/FC6, F7/F8, AF3/AF4) se află deasupra zonelor frontale, nu deasupra cortexului vizual occipital. Figură din [10].

2.4 Discuții și Implicații

Aceste rezultate au o relevanță directă pentru tema centrală a tezei. Dacă înțelegerea vizuală în creier implică rețele corticale distribuite — inclusiv zone responsabile de limbaj, planificare și raționament abstract — atunci abordările computaționale ale înțelegerii vizuale ar trebui să încerce să capteze acest nivel de procesare structurată, conceptuală. O reprezentare care operează doar la nivelul pixelilor și al features statistice învățate, oricât de puternică ar fi, ar pierde tipul de informație relațională și semantică pe care creierul îl aduce, evident, chiar și într-o simplă sarcină de clasificare a imaginilor. Această observație ne-a motivat să dezvoltăm Graful Evenimentelor în Spațiu și Timp (GEST), o reprezentare explicită bazată pe grafuri care codifică evenimentele, actorii și relațiile lor temporale și spațiale la nivel conceptual. GEST este subiectul capitolului următor.

Capitolul 3

Graful Evenimentelor în Spațiu și Timp

3.1 Modelul GEST

Un GEST este un graf orientat $G = (\mathcal{V}, \mathcal{E})$ în care nodurile reprezintă evenimente, iar muchiile codifică relațiile dintre acestea. Fiecare nod-eveniment E_i specifică o acțiune, entitățile implicate, o locație, un interval de timp și proprietăți suplimentare. O caracteristică distinctivă este faptul că obiectele fizice sunt și ele reprezentate ca evenimente (de exemplu, „mașina lui Ion există"), unificând astfel acțiunile și entitățile sub aceeași mașinărie reprezentațională. Noduri speciale `Exists` declară actorii și obiectele, stabilind inventarul de entități.

Muchiile leagă două evenimente și pot codifica relații temporale (before, after, same_time, concurrent), spațiale (near, behind, left, right), cauzale (because, causes) sau semantice (same, co-reference). Relațiile temporale se inspiră din algebra intervalelor a lui Allen [1].

Poate cea mai puternică proprietate a GEST este capacitatea de a opera la mai multe niveluri de abstractizare prin două operații: *colaps graf-nod* (orice subgraf poate fi comprimat într-un singur eveniment de nivel superior) și *expansiune nod-graf* (orice eveniment poate fi extins într-un subgraf detaliat). Această proprietate ierarhică este esențială pentru a face legătura dintre vederea computațională și limbajul natural, unde descrierile aceleiași scene variază în granularitate. Figura 3.1 ilustrează un GEST pentru o narațiune cu mai multe evenimente.

3.2 Viziunea: GEST ca Hub Multimodal

GEST a fost gândit ca un hub care conectează mai multe modalități (Figura 3.2) prin patru direcții funcționale: (A) text spre GEST — construirea unui graf din limbaj natural; (B) GEST spre text — generarea de limbaj pornind de la un graf; (C) GEST spre video — generarea unui videoclip dintr-o specificație; și (D) video spre GEST — extragerea unui graf dintr-un videoclip. Oricare două direcții pot fi înlănțuite pentru conversia între modalități, graful

[0]Acest capitol se bazează pe: Masala, **Cudlenco**, Rebedea, Leordeanu — ICCV 2023 [21].

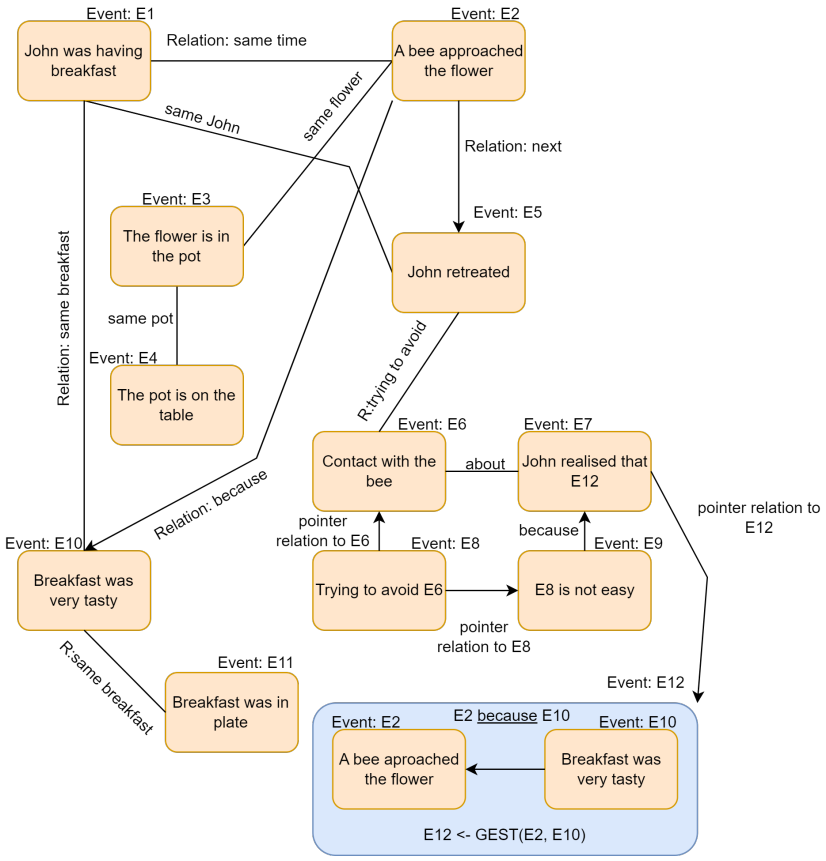


Figura 3.1: **Exemplu de GEST.** O narațiune descompusă în douăsprezece noduri-eveniment, conectate prin muchii temporale, cauzale, spațiale și de coreferință. Jos: colapsul graf-nod agregă evenimentele într-un eveniment de nivel superior. Figură din [21].

servind ca etapă intermediară explicită și inspectabilă, ce poate fi examinată și corectată independent. Un singur GEST poate produce mai multe videoclipuri distincte (aceeași narațiune în medii diferite) și mai multe texte (aceleași evenimente la niveluri diferite de detaliu), iar mai multe texte sau videoclipuri pot fi puse în corespondență cu același GEST dacă împărtășesc aceeași structură de evenimente.

3.3 Text spre GEST

Au fost dezvoltate trei abordări pentru conversia textului din limbaj natural în GEST: (1) parsare bazată pe reguli, cu corecție umană, folosind parsare de dependențe și etichetare a rolurilor semantice pentru a extrage evenimente și relații; (2) fine-tuning pe GPT-3 cu exemple de GEST adnotate manual, producând GEST-uri direct din text prin few-shot learning; și (3) generare zero-shot cu ChatGPT pe baza unor prompturi structurate. Toate trei produc GEST-uri la nivel semantic, dar niciuna nu garantează că graful rezultat este executabil de către GEST-Engine. Acoperirea acestei diferențe a necesitat cuplarea modelelor lingvistice cu impunerea procedurală a constrângerilor, descrisă în Capitolul 5.

GEST spre Text. Am dezvoltat un pipeline de generare de text în două etape pentru

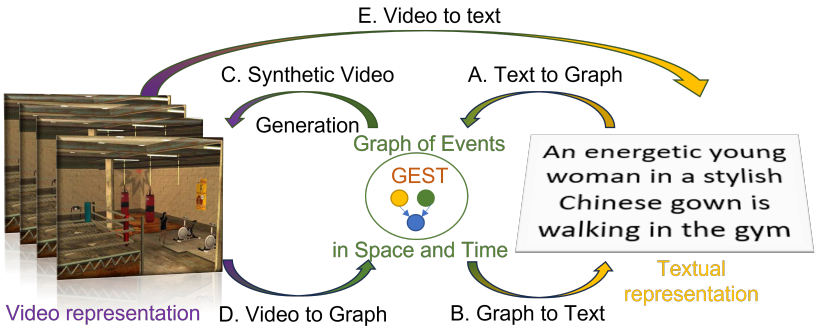


Figura 3.2: **Privire de ansamblu funcțională a platformei GEST.** GEST joacă rolul de componentă centrală, permițând conversia bidirecțională între reprezentările vizuale și cele textuale. Figură din [21].

GEST-Engine (Capitolul 4), adaptând abordarea bazată pe proto-limbaj a lui Masala et al. [22] — care generează algoritmic descrieri textuale structurate pornind de la GEST-uri, sortând și grupând evenimentele în spațiu și timp — pentru a include camerele în care se desfășoară acțiunile și genul actorilor și cuplându-l cu o etapă de rafinare prin GPT-4o, care produce descrieri naturale și fluente pornind de la rezultatele sistemului GEST din Capitolul 4.

Video spre GEST. Complementar direcției noastre de generare, Masala et al. [22] au dezvoltat o abordare pentru extragerea de GEST-uri din videoclipuri reale, combinând segmentare semantică, detecție de acțiuni și urmărire a obiectelor. Această direcție închide bucla: videoclipurile pot fi reconvertite în reprezentări de evenimente structurate, permițând compararea video-video la nivel semantic.

GEST spre Video. Am dezvoltat un motor de generare de video care primește un GEST ca intrare și produce un videoclip cu mai mulți actori într-un mediu de simulare 3D. Sistemul complet este descris în Capitolul 4.

3.4 Validare Inițială

3.4.1 Potrivire de Grafuri pentru Similaritatea Textelor

Am evaluat dacă în spațiul GEST compararea textelor surprinde similaritatea semantică mai eficient decât metricile standard. Ambele texte au fost convertite în GEST-uri și comparate folosind Spectral Matching (SM) [20] și Neural Graph Matching (NGM) [30], cu matrici de afinitate bazate pe GloVe.

Tabelul 3.1 arată că GEST-SM obține cea mai mare corelație (61,70%) și cel mai bun scor Fisher (1,20), în timp ce GEST-NGM obține cea mai bună acuratețe (86,31%). Combinarea BLEURT cu GEST-SM aduce cele mai bune rezultate la toate metricile (Tabelul 3.2), confirmând că GEST surprinde o dimensiune a similarității semantice complementară metricilor

Tabela 3.1: GEST față de metricile de similaritate a textelor pe Videos-to-Paragraphs. Tabel din [21].

Metodă	**Corr(%)**	**Acc(%)**	**F**	**AUC(%)**
BLEU@4	24,45	75,52	0,28	52,65
METEOR	58,48	84,23	1,12	73,90
ROUGE	51,11	83,40	0,72	68,92
SPICE	59,42	84,65	1,04	74,43
BERTScore	57,39	85,89	1,07	**77,93**
GEST-SM	**61,70**	84,65	**1,20**	75,47
GEST-NGM	60,93	**86,31**	0,98	76,75

Tabela 3.2: BLEURT combinat cu alte metrici. GEST-SM aduce cel mai mare câștig complementar. Tabel din [21].

Metodă	**Corr(%)**	**Acc(%)**	**F**	**AUC(%)**
BLEURT	70,93	90,04	2,03	88,02
+GEST-SM	**72,89**	**90,87**	**2,21**	**89,80**
+GEST-NGM	71,91	90,46	2,05	88,58
+SPICE	71,94	88,80	2,09	87,71
+METEOR	71,20	89,63	2,07	87,62

neurale de text.

3.4.2 Evaluare Umană a Videoclipurilor Generate

Am comparat videoclipurile generate prin GEST cu cele produse de CogVideo [15] și Text2VideoZero [19]. Șase evaluatori independenți au notat 111 perechi video-text. În 87% dintre cazuri, evaluatorii au ales videoclipul generat prin GEST drept cea mai bună potrivire semantică (scor mediu 7,93/10 față de 5,55 pentru Text2VideoZero și 4,03 pentru CogVideo). O evaluare automată cu VALOR [6] a confirmat aceeași ierarhie: videoclipurile noastre au obținut cel mai bun scor la șase din șapte metrici standard de captioning.

3.5 Discuții

Aceste rezultate confirmă GEST ca o platformă de reprezentare viabilă pentru a conecta vederea computațională și limbajul natural la nivelul conținutului semantic. Reprezentarea s-a dovedit suficient de expresivă pentru a surprinde narațiuni complexe cu mai mulți actori, cu ordonare temporală, context spațial și relații cauzale, și suficient de structurată pentru a permite atât compararea bazată pe grafuri, cât și generarea deterministă de video.

Capitolul 4

GEST-Engine: De la Grafuri de Evenimente la Video Sintetic

Ideea folosirii unui graf al evenimentelor în spațiu și timp ca specificație pentru generarea de video a pornit de la observația că, dacă am putea controla programatic fiecare entitate dintr-o lume 3D — creând actori, plasând obiecte, declanșând animații și înregistrând rezultatele — am putea produce date de antrenare la o scară și cu o densitate de adnotări pe care niciun efort manual de etichetare nu le-ar putea egala. Structura de graf descrisă în Capitolul 3 a oferit platforma formală; rămânea de construit un sistem care să primească un astfel de graf ca intrare și să îl materializeze sub forma unui videoclip.

4.1 Privire de Ansamblu și Evoluție

GEST-Engine funcționează pe Grand Theft Auto: San Andreas (2004) prin intermediul Multi Theft Auto (MTA), o platformă open-source pentru modding multiplayer care expune motorul jocului printr-o interfață de scripting Lua. Am ales această platformă pentru caracterul ei open-source (GPLv3), pentru cerințele hardware reduse (care permit execuția paralelă pe mai multe mașini virtuale rulând pe hardware comun) și pentru bogăția resurselor preexistente: 733 de modele de obiecte împărțite în 32 de categorii, peste 2.500 de animații în 101 categorii, 312 skinuri de actori și peste 70 de medii distincte.

GEST-Engine reprezintă un efort de inginerie consistent. Sistemul cuprinde aproximativ 38.000 de linii de Lua pentru nucleul motorului, 8.500 de linii de C++ pentru modulul nativ de captură de imagine și pentru pathfinding, 53.000 de linii de Python pentru generatorul procedural și sistemul agentic text-GEST (Capitolul 5) și 600 de linii de JavaScript pentru vizualizatorul interactiv de GEST — în total, aproximativ 100.000 de linii de cod pe toate componentele. Întregul cod a fost scris de un singur cercetător pe parcursul programului doctoral. Decizia de a construi peste un joc existent, în loc să creăm o lume virtuală de la

[0] Acest capitol se bazează pe: **Cudlenco**, Masala, Leordeanu — publicată la ICLR 2026 Workshop on World Models [7]; și **Cudlenco**, Masala, Leordeanu — trimisă spre publicare la ECCV 2026 (arXiv:2604.10385) [9].



Figura 4.1: **Editorul de Lume în uz.** Un episod de tip living (stânga) și unul de tip sală (dreapta), cu POI-uri (cilindri), vârfuri de regiune, poziții de cameră și liste de acțiuni. POI-urile verzi au acțiuni definite; cele roșii nu au.

zero, s-a dovedit esențială pentru a face proiectul fezabil: folosind resursele, randarea, fizica și sistemele de animație deja prezente în GTA San Andreas, am putut concentra întregul efort de inginerie pe straturile de orchestrare a grafului, coordonare temporală și colectare a adnotărilor. Pentru comparație, VirtualHome [25] — sistemul cel mai apropiat ca scop — a fost dezvoltat de o echipă de șapte cercetători, cu nouă contributori la codul public, pe o perioadă comparabilă (2018–2024, conform istoricului de commit-uri publice), și totuși rămâne limitat la programe secvențiale cu un singur actor în medii interioare închise, fără orchestrare temporală sau adnotări spațiale dense. Codul sursă este public[1,2].

4.2 Editorul de Lume

O limitare a GTA San Andreas prin MTA este că, deși avem acces programatic la obiectele, animațiile și personajele jocului, nu avem acces la geometria mediului — nu există un API care să interogheze planul camerelor, pozițiile mobilierului sau traseele pe care se poate merge prin clădire. Pentru a depăși această limitare, am dezvoltat un Editor de Lume: un instrument integrat în joc care ne permite să parcurgem orice mediu și să definim interactiv toate informațiile spațiale și semantice necesare simulării.

Editorul de Lume definește următoarele elemente, serializate în fișiere JSON încărcate la rulare: *Regiuni* (limite poligonale corespunzătoare unor zone spațiale precum dormitoare, bucătării, holuri), *Puncte de Interes (POI-uri)* (locații specifice în interiorul regiunilor unde se pot executa acțiuni, fiecare cu lanțuri de acțiuni și tipuri de obiecte definite), *Templates și Supertemplates* (configurații reutilizabile de obiecte la POI-uri, supertemplates oferind selecție aleatoare între variante de templates pentru diversitate vizuală), *Camere* (poziționate în interiorul regiunilor pentru cinematografie automată) și *Grafuri de pathfinding* (rețele de navigare care permit actorilor să meargă între regiuni și POI-uri evitând obstacolele).

[1] `https://github.com/ncudlenco/mta-sim`
[2] `https://github.com/ncudlenco/multiagent_story_system`

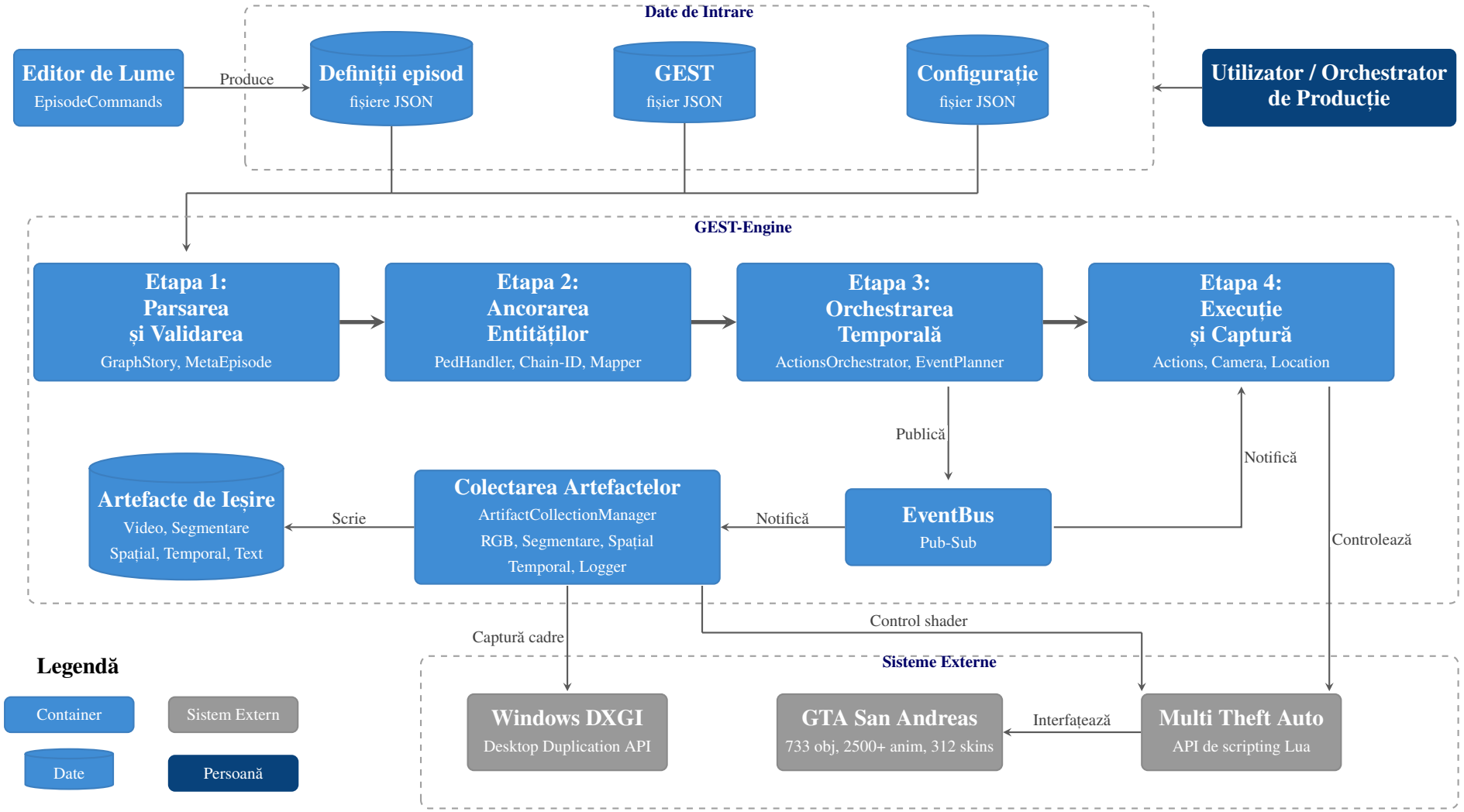


Figura 4.2: **Arhitectura sistemului GEST-Engine.** Datele de intrare alimentează pipeline-ul în patru etape: **Etapa 1** (Secțiunea 4.3) parsează și validează GEST-ul, **Etapa 2** (Secțiunea 4.3) ancorează actorii, obiectele și acțiunile în lumea 3D, **Etapa 3** (Secțiunea 4.3) orchestrează execuția cu mai mulți actori și publică evenimentele, **Etapa 4** (Secțiunea 12) execută acțiunile prin Multi Theft Auto. EventBus notifică atât Etapa 4, cât și sistemul de Colectare a Artefactelor (Secțiunea 4.4), care capturează adnotări multi-modale dense.

Algorithm 1: Calculul segmentelor temporale prin planificare ghidată de dependențe.

Intrare: Evenimente GEST $\mathcal{V}$, constrângeri temporale C
Ieșire: Segmente temporale $\{S_1, S_2, \ldots, S_k\}$ care partiționează $\mathcal{V}$

$H \leftarrow$ BuildHappensBeforeGraph(C);
$H^* \leftarrow$ FloydWarshallClosure(H);
satisfied $\leftarrow \emptyset$; $k \leftarrow 0$;
while $|satisfied| < |\mathcal{V}|$ **do**
    ready $\leftarrow \{e \in \mathcal{V} \setminus \text{satisfied} \mid \text{IsReady}(e, H^*, \text{satisfied})\}$;
    if *ready* $= \emptyset$ **then**
        eroare „Deadlock: dependență circulară detectată”;
    groups $\leftarrow$ GroupByConcurrency(ready, H^*);
    foreach *grup* $g \in$ *groups* **do**
        $k \leftarrow k + 1$; $S_k \leftarrow g$;
        satisfied $\leftarrow$ satisfied $\cup\, g$;
return $\{S_1, \ldots, S_k\}$;

4.3 Privire de Ansamblu a Arhitecturii

Motorul transformă o specificație GEST într-un videoclip adnotat în patru etape (Figura 4.2).

Etapa 1: Parsarea și Validarea Grafului. GEST-ul de intrare este parsat, cerințele sale (actori, obiecte, acțiuni, locații) sunt extrase, iar un set valid de episoade este selectat printr-un algoritm de tip set-cover cu backtracking. Pentru poveștile cu mai multe locații, un wrapper MetaEpisode agregă mai multe episoade și generează acțiunile de deplasare între ele. Un grup valid de episoade este ales aleator, asigurând astfel mai multe simulări posibile pentru același GEST.

Etapa 2: Ancorarea Entităților și a Acțiunilor. Entitățile abstracte ale grafului sunt puse în corespondență cu obiecte concrete 3D printr-un sistem de chain-ID care asigură identitatea consistentă a obiectelor de-a lungul acțiunilor și previne conflictele între actori pe resurse comune. Fiecare obiect și fiecare secvență de acțiuni cerute sunt mapate la toate lanțurile valide de POI-uri din episoadele alese, prin mapări bidirecționale care permit selecția amânată, la rulare, dintre alternativele valide.

Etapa 3: Orchestrarea Temporală. ActionsOrchestrator coordonează execuția cu mai mulți actori. Un EventPlanner construiește un graf happens-before pornind de la muchiile temporale și aplică închiderea tranzitivă Floyd-Warshall [13] pentru a obține o ordine totală, apoi împarte evenimentele în segmente temporale prin planificare ghidată de dependențe (Algoritmul 1). Evenimentele dintr-un segment se execută concurent; segmentele se execută secvențial. Astfel sunt suportate trei tipare de sincronizare: secvențial ($e_1 \rightarrow e_2$), paralel ($e_1 \leftrightarrow e_2$ via `same_time`) și întrețesut ($e_1 \parallel e_2$ via `concurrent`).

Etapa 4: Execuție și Captură. Acțiunile sunt distribuite handlerelor de actori din mediul 3D. PedHandler gestionează ciclul de viață al actorilor, 55 de clase de acțiuni implementează

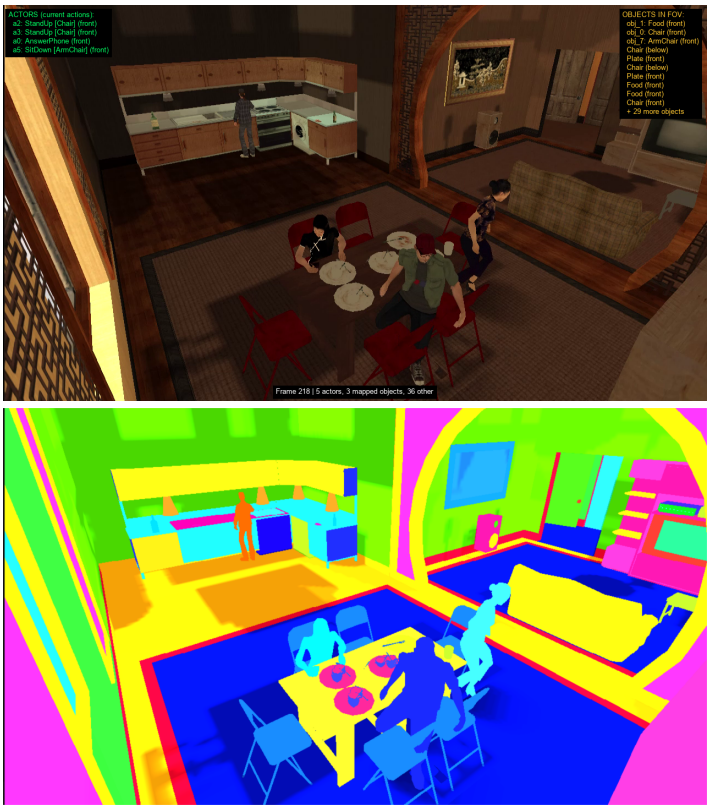


Figura 4.3: **Date multi-modale.** Sus: cadru RGB cu evenimentele active și adnotări spațiale relative la cameră. Jos: segmentare la nivel de instanță prin hashing de texturi în HLSL — fiecare textură unică este mapată la o culoare distinctă. Figură din [7].

întreaga taxonomie de acțiuni, sistemul de locație gestionează semantica spațială și selecția POI-urilor, iar sistemul de cameră oferă cinematografie automată cu urmărirea actorilor pe baza focusului. Un EventBus decuplează orchestrarea de sistemul de colectare a artefactelor printr-un mecanism de tip publisher-subscriber.

4.4 Colectare Multi-Modală a Artefactelor

Sistemul de colectare a artefactelor surprinde adnotări dense, aliniate pe cadru, în timpul simulării, fără cost suplimentar. Un ArtifactCollectionManager coordonează capturarea printr-o mașină de stări de tip freeze/collect/unfreeze, asigurând captura multi-modală consistentă pe cadru. Sistemul produce trei tipuri principale de artefacte (Figura 4.3):

Captura RGB folosește un modul nativ C++ construit peste API-ul Windows Desktop Duplication (DXGI), care preia cadrele direct din front buffer-ul GPU-ului, cu latență minimă. Cadrele sunt codificate în H.264 prin Windows Media Foundation, tot direct pe GPU.

Segmentarea la nivel de instanță este produsă de un pixel shader HLSL care atribuie fiecărei texturi unice o culoare distinctă prin hashing FNV-1a, oferind o segmentare la nivel de textură mai fină decât măștile clasice la nivel de obiect.

Grafuri de relații spațiale per cadru sunt calculate interogând poziția și rotația 3D a

fiecărei entități la fiecare cadru și calculând relațiile dintre toate perechile de entități — nu doar entitate-cameră — cu identificatorii la nivel de poveste păstrați pe parcursul simulării.

Alinierea temporală eveniment-cadru este obținută prin abonarea la semnalele `graph_event_start` și `graph_event_end`, înregistrând cadrul exact la care fiecare eveniment GEST începe și se termină.

4.5 Generare Procedurală de GEST-uri

Generatorul procedural produce GEST-uri aleatoare, dar valide, pornind de la registrul de capabilități al motorului. El funcționează prin generarea de lanțuri de acțiuni pe runde: pentru fiecare regiune din succesiunea spațială a poveștii, fiecare actor primește câte un lanț de acțiuni pe rundă, construit urmând secvențe valide de acțiuni la POI-urile disponibile. Lanțurile sunt construite în buffere temporare și sunt definitivate în GEST doar atunci când sunt complet valide, garantând executabilitatea prin construcție. Interacțiunile dintre actorii aflați în aceeași locație (strângeri de mână, conversații, schimburi de obiecte) sunt create cu constrângeri `same_time` pentru execuție sincronizată. Relațiile temporale dintre runde și tranzițiile între regiuni asigură o ordonare secvențială curată.

În afara generării de sine stătătoare, generatorul procedural servește și ca backend de gestionare a stării pentru sistemul agentic descris în Capitolul 5, expunând starea sa internă (actori, evenimente, constrângeri, urmărirea obiectelor, capacitatea POI-urilor) printr-o interfață de delegare ce permite unui agent extern bazat pe LLM să invoce aceleași operații validate.

4.6 Pipeline-ul de Generare a Textului

Motorul produce descrieri în limbaj natural ale poveștilor simulate printr-un pipeline în două etape. Mai întâi, un generator de proto-limbaj concatenează mecanic numele acțiunilor cu descrierile entităților și conectorii temporali, producând o ieșire intenționat incorectă gramatical, dar completă din punct de vedere informațional. Apoi, GPT-4o rescrie proto-limbajul într-un text natural și corect gramatical, păstrând toate acțiunile și relațiile temporale.

4.7 Orchestrarea Producției

Pipeline-ul de producție gestionează execuția în lot pe mai multe mașini virtuale paralele. Un orchestrator Python centralizat se ocupă de distribuția joburilor, monitorizarea stării, detecția și recuperarea automată în caz de eroare, validarea artefactelor și încărcarea în cloud. Corpusul GTASA, format din 938 de videoclipuri, a fost produs pe o singură stație de lucru cu 128 GB de RAM, folosind 25 de mașini virtuale paralele.

Capitolul 5

Generare Agentică Text-GEST

5.1 Introducere

Abordările text-spre-GEST din Capitolul 3 produc GEST-uri la nivel semantic, însă niciuna nu garantează că graful rezultat este executabil de către GEST-Engine. Un GEST care face referire la acțiuni, obiecte sau locații care nu există în mediul de simulare ori care încalcă constrângerile temporale și regulile lanțurilor de acțiuni va eșua la execuție. Provocarea este de a genera GEST-uri care sunt în același timp *coerente narativ* (spun o poveste cu sens) și *valide pentru simulator* (respectă constrângerile motorului). LLM-urile excelează la primul aspect, dar nu pot urmări precis starea pe parcursul celor 50–200 de evenimente ale unei povești cu mai mulți actori, ceea ce este necesar pentru cel de-al doilea.

5.2 Abordarea LLM cu Etape

Am construit mai întâi un pipeline de rafinare LLM organizat pe etape: un workflow LangGraph care înlănțuie șase etape specializate — *Concept* (GEST ierarhic abstract), *Casting* (alocarea de skinuri), *Episode Placement* (mapare scene la episoade), *Setup* (pregătire în afara cadrului), *Screenplay* (translatarea narațiunii în secvențe de acțiuni) și *Scene Detail* (GEST complet executabil) — fiecare cu validare a ieșirii structurate prin Pydantic. Pipeline-ul a produs povești coerente narativ, dar din 50 de încercări de generare, **niciuna** nu a produs o specificație executabilă. Modurile de eșec au inclus acțiuni halucinate, lanțuri de acțiuni invalide, încălcări ale ciclului de viață al obiectelor, cicluri temporale și depășirea contextului. Problema fundamentală: LLM-urile nu pot menține coerența stării pe sute de evenimente, iar corecția post-hoc presupune backtracking prin starea dependentă, cu cost superliniar. Acest lucru a motivat o arhitectură bazată pe separarea responsabilităților.

[0]Acest capitol se bazează pe: **Cudlenco**, Masala, Leordeanu — trimisă spre publicare la CVPR Workshop 2026; arXiv:2604.10383 [8].

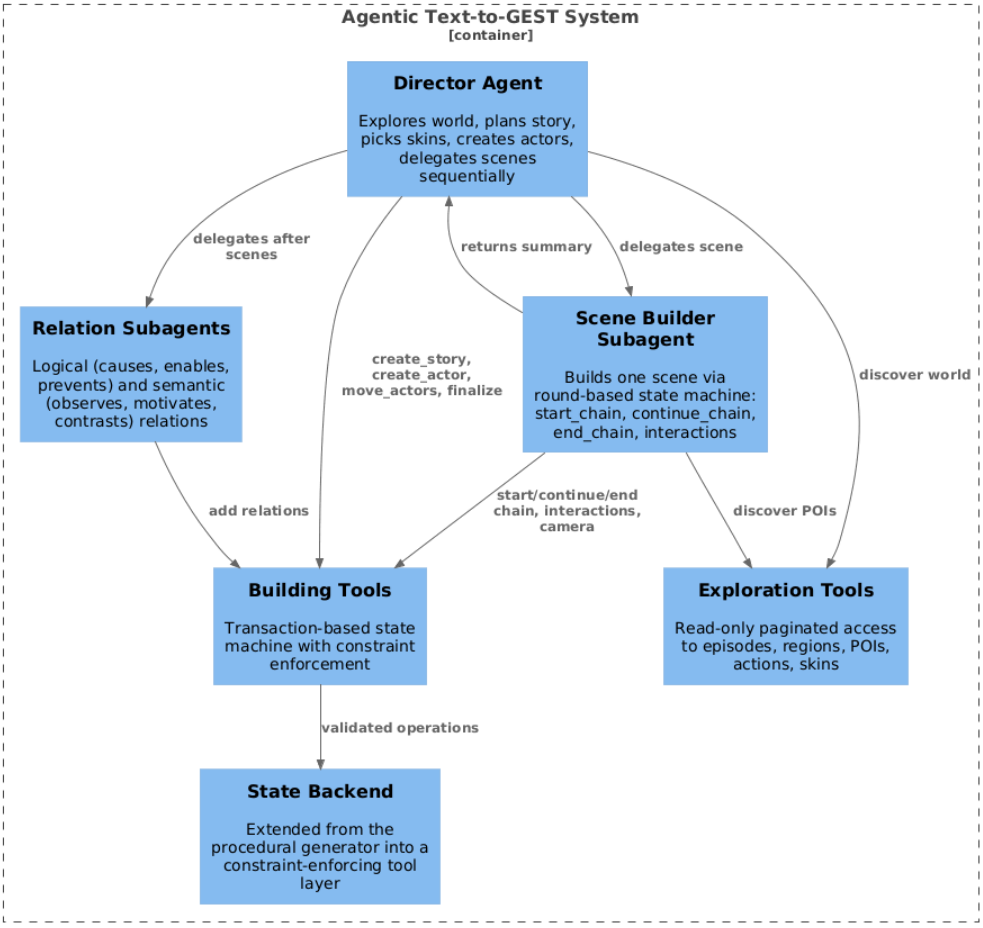


Figura 5.1: **Arhitectura sistemului agentic.** Director Agent planifică povestea și deleagă scenele individuale către Scene Builder Subagent. Ambii agenți interacționează prin Exploration Tools (read-only) și Building Tools (care impun constrângerile), operând asupra unui State Backend. Figură din [8].

5.3 Sistemul Agentic cu Apel de Funcții

Sistemul agentic separă planificarea narativă de impunerea constrângerilor. LLM-ul ia deciziile narative prin raționament în limbaj natural, în timp ce un backend programatic de stare — generatorul procedural din Capitolul 4, refolosit ca strat de tools ce impune constrângerile — gestionează urmărirea stării prin apeluri validate de tools. Sistemul este construit pe LangGraph pentru orchestrarea workflow-ului și pe deepagents pentru delegarea ierarhică între agenți. Atât Directorul, cât și Scene Builder-ul folosesc Claude Haiku[1] ca backend LLM.

Director Agent primește un seed-text și parcurge patru faze: explorare (răsfoirea episoadelor, regiunilor, POI-urilor și skinurilor prin tools read-only paginate), casting (crearea actorilor cu nume, gen, skin și regiune de pornire), construirea scenelor (delegarea fiecărei scene către Scene Builder, însoțită de o descriere narativă a sarcinii) și finalizare (legarea capetelor de scenă prin constrângeri temporale între scene).

Scene Builder Subagent primește un context izolat (scenă, episod, regiune, actori,

[1] `https://www.anthropic.com/claude/haiku`

Tabela 5.1: Rate de succes la generare end-to-end. Tabel din [8].

Abordare	Încercări	Reușite	Rată
LLM cu etape	50	0	0%
Procedural aleator	34	25	74%
Hibrid agentic (cu seed)	25	20	80%
Hibrid agentic (autonom)	~20	~10	~50%

descriere narativă) și construiește evenimentele printr-o mașină de stări organizată pe runde. În fiecare rundă: `start_round` returnează stările curente ale actorilor; pentru fiecare actor, `start_chain` returnează acțiunile valide următoare, urmată de apeluri `continue_chain` și de `end_chain` pentru commit; opțional, `do_interaction` creează evenimente sincronizate între doi actori; `end_round` adaugă ordonarea temporală între actori. Scene Builder nu poate crea actori, nu poate muta actori între regiuni și nu poate finaliza GEST-ul — păstrând o separare clară între nivelul de poveste și cel de scenă.

Relation Subagents îmbogățesc opțional graful cu relații logice (causes, enables, prevents, requires) și relații semantice (observes, interrupts, motivates). Acestea populează tipurile de muchii definite în formalismul GEST pe care generarea procedurală nu le produce niciodată.

Impunerea constrângerilor prin tools implementează o mașină de stări tranzacționale cu patru stări (`IDLE`, `STORY_CREATED`, `IN_SCENE`, `IN_ROUND`). Mecanisme principale: urmărirea stării actorilor (postură, obiecte ținute, locație), lanțuri tranzacționale (buffere temporare, commit sau abandon), urmărirea capacității POI-urilor, detecția ciclurilor temporale și regulile de interacțiune. Fiecare GEST produs este executabil prin construcție, deoarece operațiile invalide sunt respinse la nivel de tool.

5.4 Evaluare

Reușita generării (Tabelul 5.1). Pipeline-ul cu etape produce 0% ieșire executabilă. Sistemul agentic obține 80% în regim cu seed și aproximativ 50% în regim complet autonom. Fiecare GEST valid este corect structural prin construcție — eșecurile vin fie din partea LLM-ului (bucle repetitive de planificare), fie din partea motorului, niciodată din încălcarea constrângerilor.

Calitatea narativă. Un juriu format din 3 modele LLM (GPT-5.2 Pro, Gemini 3.1 Pro, Claude Opus 4.6) a comparat poveștile agentice cu cele procedurale. Atunci când narațiunea autorită de Director este comparată cu Video Descrierile din Graf (VDG) procedurale, narațiunile agentice câștigă 79% dintre comparațiile de text și 74% dintre cele de video, confirmând o structură narativă veritabilă, absentă din generarea procedurală.

Fidelitatea narativă. Acoperirea evenimentelor este 1,89/5 în regim autonom și 3,12/5 în regim cu seed. Diferența reflectă mai degrabă constrângeri de platformă (posibilități

cinematografice limitate, control temporal fin, atmosferă și combinarea lină a acțiunilor) decât limitări arhitecturale — sistemul agentic se transferă direct pe orice motor care expune aceste canale.

5.5 Sistemul Agentic ca Realizare Completă a GEST

Pe parcursul acestei teze am construit incremental fiecare direcție a platformei GEST. În Capitolul 3 am dezvoltat conversia text-spre-GEST prin parsare bazată pe reguli, fine-tuning pe GPT-3 și generare zero-shot cu ChatGPT [21] — producând grafuri valide narativ, dar fără nicio garanție că pot fi executate într-o simulare. Am adaptat pipeline-ul de proto-limbaj al lui Masala et al. [22] pentru generarea de text a motorului, adăugând contextul de locație și de gen pentru a permite conversia GEST-spre-text pentru datele motorului. În Capitolul 4 am construit GEST-Engine [7, 9], care execută GEST-uri în mod determinist într-un mediu 3D, și generatorul procedural care produce grafuri executabile la scară — dar doar cu relații temporale și fără coerență narativă. Fiecare dintre aceste componente exercită o parte a formalismului GEST, dar niciuna nu îl exercită în întregime simultan.

Sistemul agentic le aduce împreună. Directorul folosește proprietatea de expansiune ierarhică a GEST pentru a descompune o narațiune din limbaj natural în scene și apoi în evenimente granulare — direcția text-spre-GEST, acum cu garanții de executabilitate. Scene Builder ancorează aceste evenimente în capabilitățile motorului prin tools de explorare și prin registrul de capabilități. Backend-ul de stare impune validitatea prin același generator procedural care produce corpusul GTASA. Relation Subagents populează tipurile de muchii logice și semantice definite în formalismul GEST — aceleași tipuri de relații pe care abordările noastre anterioare de text-spre-GEST le surprindeau la nivel narativ, dar pe care generatorul procedural nu le produce niciodată. GEST-Engine execută apoi graful complet populat, producând video cu adnotări dense. Iar pipeline-ul de generare de text închide bucla înapoi spre limbaj natural.

Direcția complementară este și ea activă. Abordarea de extragere video-spre-GEST a lui Masala et al. [22] poate recupera grafuri structurate de evenimente din videoclipurile generate, în timp ce pipeline-ul de generare de text convertește GEST-ul executat într-o descriere VDG a ceea ce s-a întâmplat vizual. Pornind de la acel VDG, abordarea de reconstituire a narațiunii folosită în evaluarea noastră demonstrează că un LLM poate deduce intenția la nivel de poveste pornind de la descrierea granulară a execuției — închizând astfel o a doua buclă, de la video înapoi la limbaj natural, la nivel narativ, nu doar la nivelul acțiunilor.

Capitolul următor investighează valoarea practică a adnotărilor dense produse la scara corpusului.

Capitolul 6

Evaluare Experimentală

În acest capitol evaluăm dacă sistemul și datele rezultate sunt utile — nu doar ca artefact de inginerie, ci ca instrument de cercetare pentru comunitatea mai largă a înțelegerii video. Întrebarea centrală este în ce măsură putem folosi modelele explicite de lume oferite de motoarele de joc 3D pentru a genera videoclipuri semnificative semantic, pentru înțelegere video și pentru benchmarking.

6.1 Setul de date GTASA

Am produs GTASA (Ground Truth Annotations for Spatiotemporal Analysis): 938 de videoclipuri cu mai mulți actori, generate procedural, care cuprind 28.200 de clipuri de acțiune din 37 de tipuri de acțiuni și 14,3 ore de filmare, cu 892 de milioane de relații spațiale pereche distribuite pe 1,45 milioane de cadre și 29.579 de alinieri temporale exacte eveniment-cadru. Fiecare poveste are între 7 și 65 de evenimente (medie 29,4) cu 2–6 actori (medie 3,43), desfășurate în 10 episoade din 4 categorii de medii. Corpusul este public[1], alături de codul sursă al GEST-Engine[2] și de sistemul de orchestrare[3].

Tabelul 6.1 compară corpusul GTASA cu seturi de date video și platforme sintetice existente. Per clip, GEST-Engine oferă cu două ordine de mărime mai multe relații spațiale decât Action Genome (31,6K față de 170 per clip). Folosind aceleași descrieri textuale ca prompturi, am generat și videoclipuri corespunzătoare cu VEO 3.1 [14] și WAN 2.2 [28] ca metode neurale de referință pentru evaluarea umană și pentru experimentele de captioning.

[0]Acest capitol se bazează pe: **Cudlenco**, Masala, Leordeanu — trimisă spre publicare la ECCV 2026; arXiv:2604.10385 [9].

[1]`https://huggingface.co/datasets/nnc-001/gtasa-01`

[2]`https://github.com/ncudlenco/mta-sim`

[3]`https://github.com/ncudlenco/multiagent_story_system`

Tabela 6.1: Comparație cu seturi de date video și platforme sintetice. Per clip, GEST-Engine oferă cu două ordine de mărime mai multe relații spațiale decât Action Genome (31,6K față de 170 per clip). Tabel din [9].

Set de date	Clipuri	Ore	Rel. Spațiale	Adn. Temp.	Interacț.
VideoMix22M [2]	22M	1M+	–	–	nu
Kinetics [18]	650K	1,800	–	nivel de clip	nu
Action100M [5]	147M	128K	–	aprox.	nu
CARLA [11]	–	–	–	–	nu
BEDLAM [4]	30K	–	–	–	nu
VirtualHome [25]	2.8K	–	–	nivel de acțiune	nu
VH Act. Genome [26]	5.8K	–	graf de scenă	nivel de acțiune	nu
PVSG [31]	400	–	150K cadre	adnotat	da
GEST [21]	25	–	–	secvențial	nu
Action Genome [17]	10K	82	1.7M	5 cadre/acț.	da
GEST-Engine (al nostru)	28.2K	14.3	**892M**	29.6K concurent	da

6.2 Evaluare Umană a Generării Video

Am colectat 1.020 de adnotări de la 16 evaluatori independenți pe 766 de videoclipuri unice, evaluând validitatea fizică (binar) și alinierea semantică (scală Likert 1–5). GEST-Engine obține 69% validitate fizică față de 18% (VEO) și 13% (WAN) și 4,09/5 aliniere semantică față de 2,50 și 1,75 (Tabelul 6.2). Cele 31% de videoclipuri ale motorului considerate invalide sunt artefacte vizuale ale jocului care stă la baza simulării (defecte de animație, jitter de pathfinding), nu eșecuri ale modelului de lume. Acordul între adnotatori a fost moderat pentru validitate (α = 0,488) și substanțial pentru alinierea semantică (α = 0,636, cu 85,4% dintre dezacorduri în limita unui singur punct Likert).

Potrivirea inversă. Ca validare independentă, un alt grup de adnotatori a scris descrieri textuale pornind exclusiv de la fiecare videoclip, fără să vadă promptul original; apoi judecători au stabilit cât de bine se potrivește fiecare descriere cu promptul original folosit ca ancoră. Videoclipurile GEST-Engine obțin cel mai mare scor mediu de potrivire (56,64/100) și cel mai bun rang mediu (1,42/3,00), urmate de VEO 3.1 (42,94, 1,86) și WAN 2.2 (23,44, 2,72) — videoclipurile generate de motor sunt descrise în moduri care recuperează cel mai fidel specificația.

Tabela 6.2: Evaluare umană a calității generării video. Tabel din [9].

Categorie	#Vid.	Valid. (↑)	Sem. (↑)	Rang (↓)
GEST (al nostru)	281	**0,69**	**4,09**	**1,08**
VEO 3.1	255	0,18	2,50	2,04
WAN 2.2	230	0,13	1,75	2,88

6.3 Captioning Video

Am făcut fine-tuning pe două modele de captioning video — VideoLLaMA3 2B [32] și Qwen3-VL 2B [3] — în trei condiții: fine-tuning pe date GEST urmat de fine-tuning pe video real (G), fine-tuning pe date VEO urmat de fine-tuning pe video real (V) și antrenare exclusiv pe date reale (R).

O proprietate distinctă a corpusului GTASA este că fiecare eveniment are un interval exact de cadre. Am exploatat acest lucru pentru a propune o strategie hibridă de eșantionare a cadrelor: pentru fiecare eveniment de acțiune din GEST am ales cadrul din mijloc (cadrul situat la centrul temporal al duratei evenimentului), apoi am completat sloturile rămase (până la maximum 64 de cadre) cu cadre eșantionate uniform la 1 fps. Astfel, fiecare acțiune este reprezentată de cel puțin un cadru, chiar și pentru evenimente scurte pe care eșantionarea uniformă le-ar putea rata complet. Tabelul 6.3 arată efectul: abordarea hibridă îmbunătățește 5 din 7 metrici, cu cea mai mare creștere la CIDEr (+6,47).

Tabela 6.3: Ablație a eșantionării cadrelor pe VideoLLaMA3 2B (condiția G). Tabel din [9].

Metodă	BLEU-4	METEOR	CIDEr	ROUGE-L	SPICE	BERTSc.	BLEURT
Hibrid	**13,62**	**36,36**	**30,25**	**36,95**	**29,72**	90,64	54,13
Uniform	12,73	35,08	23,78	36,64	29,43	**90,65**	**54,59**

Tabelul 6.4 prezintă rezultatele complete pentru captioning. Modelele antrenate prealabil pe date GEST le-au depășit constant pe celelalte, obținând cele mai bune scoruri la șase din șapte metrici automate, în ciuda diferenței mari de domeniu vizual. Antrenarea prealabilă pe VEO afectează performanța comparativ cu antrenarea exclusiv pe date reale (V se clasează ultimul pe VideoLLaMA3, cu 4,00) — datele sintetice realiste vizual, dar nesigure semantic, sunt mai dăunătoare decât absența lor. Un juriu independent format din LLM-uri (GPT-5.2 Pro, Gemini 3.1 Pro, Claude Opus 4.6) a confirmat aceeași ierarhie.

6.4 Sondare Spațio-Temporală

Am sondat patru video-encoders înghețați — V-JEPA 2 [2], VideoMAE V2 [29] (auto-supervizați), Qwen3-VL [3], VideoLLaMA3 [32] (video-encoders ale unor VLM-uri) — pe 11 sarcini de raționament spațio-temporal, folosind ground truth-ul 3D exact din corpus (Tabelul 6.5).

Video-encoders auto-supervizați codifică structura spațială mai bine decât video-encoders ale VLM-urilor. V-JEPA 2 și VideoMAE V2 le depășesc constant pe Qwen3-VL pe aproape toate sarcinile (medie 60,6 și 59,9 față de 52,4), deși Qwen3-VL este antrenat pe date cu ordine de mărime mai mari, sub supraveghere lingvistică. Extindem astfel observația din [12, 27] de la imagini la video-encoders înghețați.

Tabela 6.4: Rezultatele de captioning video. G = GEST apoi Real, R = doar Real, V = VEO apoi Real, H = Referință umană. Metrics Rank este media rangurilor pe cele 7 metrici (H primește rang 1). Jury Rank este media rangurilor date de cele trei LLM-uri de frontieră. All Avg este media între ranguri și juriu. Tabel din [9].

	VideoLLaMA3 2B				Qwen3-VL 2B			
Metrică	H	G	R	V	H	G	R	V
BLEU-4	–	**13,62**	13,49	11,48	–	10,70	**11,66**	8,51
METEOR	–	**36,36**	34,02	32,71	–	**29,68**	29,40	27,59
CIDEr	–	**30,25**	29,32	21,89	–	21,46	**25,21**	17,31
ROUGE-L	–	**36,95**	35,28	34,33	–	**30,96**	29,63	28,93
SPICE	–	**29,72**	28,05	26,44	–	**25,06**	23,98	23,63
BERTScore	–	90,64	**90,67**	90,31	–	**89,99**	89,82	89,80
BLEURT	–	**54,13**	53,20	52,99	–	49,20	49,17	**51,15**
Metrics Rank (↓)	**1,00**	**2,14**	2,86	4,00	**1,00**	**2,43**	2,86	3,71
Jury Rank (↓)	**2,02**	2,51	2,52	2,94	**1,45**	2,78	2,92	2,86
All Avg (↓)	**1,77**	**2,42**	2,61	3,21	**1,34**	**2,69**	2,90	3,07

Cea mai mare diferență între video-encoders apare la sarcinile dinamice. La Prezența mișcării diferența dintre V-JEPA 2 (88,6%) și Qwen3-VL (63,1%) este de 25,5 puncte; la Apropiere/Depărtare diferența este de 13,8 puncte. Sarcinile care necesită integrarea schimbării spațiale în timp sunt cele la care video-encoders auto-supervizați se desprind cel mai clar de cele ale VLM-urilor.

VideoLLaMA3 excelează la numărarea actorilor. Pe cele 8 sarcini comune, VideoLLaMA3 se plasează între video-encoders auto-supervizați și Qwen3-VL, dar le depășește dramatic pe toate la Numărul de actori (61,0% față de 37,3% pentru V-JEPA 2). Numărarea este un mod cunoscut de eșec al video-encoders supervizați prin limbaj [23].

Determinarea direcției între două entități rămâne dificilă. Scorurile la Direcția pereche sunt doar marginal peste baseline-ul aleator de 12,5% pentru toate video-encoders (15–17%), indicând că raționamentul direcțional fin între entități este slab reprezentat în video-encoders actuali.

6.5 Studiu de caz: Punctele Slabe ale VLM-urilor

Am evaluat dacă VLM-urile de frontieră pot detecta videoclipuri fizic invalide, folosind 94 de videoclipuri pe care toți adnotatorii au fost de acord (45 valide, 49 invalide; 49 GEST, 25 VEO, 20 WAN). GPT-5.2 Pro, Gemini 3.1 Pro și Claude Opus 4.6 au primit aceleași instrucțiuni de validitate folosite în studiul uman (Tabelul 6.6).

Toate cele trei modele se situează în jurul a 56% acuratețe — abia peste cei 47,9% pe care i-ar obține un clasificator banal care declară totul ca valid. Eșecul este sistematic, nu specific unui model: votarea cu majoritate între cele trei modele dă doar 58,51%. Cel mai elocvent este comportamentul calitativ: atunci când li se cere să descrie aceste videoclipuri

Tabela 6.5: Acuratețe echilibrată (%) pe sarcinile de sondare spațio-temporală. VL3* = parțial (8/11). Tabel din [9].

Categorie	Sarcină	Aleat.	V-JEPA 2	VMAE V2	Qwen3	VL3*
Scenă	Nr. actori	20,0	37,3	33,0	32,2	**61,0**
	Pres. mișcare	50,0	**88,6**	87,6	63,1	–
Entitate	Pres. entit.	50,0	**66,2**	63,0	56,2	–
	Dist. cameră	33,3	57,9	**58,2**	57,4	57,8
	Schimb. unghi	50,0	**75,5**	75,3	62,4	60,1
	Apropie/Depărt.	50,0	**70,9**	68,9	57,1	54,0
Pereche	Ord. adâncime	50,0	61,3	**63,0**	59,4	–
	Direcție per.	12,5	**17,4**	16,4	15,1	15,7
	Distanță per.	33,3	40,0	**41,9**	36,1	40,1
	Mișcare rel.	50,0	67,8	**69,4**	61,5	62,8
Temporal	Limita evenim.	50,0	**84,1**	81,7	75,6	78,7
Medie (toate 11)			**60,6**	59,9	52,4	–
Medie (8 comune)			**56,4**	55,6	49,7	53,8

invalide, modelele ignoră cu totul inconsistențele și narează evenimentele în mod natural, chiar și atunci când obiectele se teleportează sau actorii își schimbă identitatea în mijlocul scenei. Nu doar că nu reușesc să detecteze inconsistențele — nu reprezintă deloc tipul de permanență a obiectelor și de urmărire a identității care ar face detecția posibilă.

6.6 Discuții

O temă recurentă este că coerența semantică contează mai mult decât realismul vizual. Videoclipurile generate de motorul nostru — în ciuda graficii din 2004 — depășesc generatoarele neurale de top în ceea ce privește fidelitatea semantică, oferă date de pretraining mai eficiente decât videoclipurile fotorealiste produse de VEO și permit sarcini de sondare diagnostice pe care niciun set de date din lumea reală nu le poate susține. Aceasta arată că modelele explicite de lume, însoțite de reprezentări structurate, oferă o valoare complementară realismului vizual al generării neurale.

Tabela 6.6: Performanța VLM-urilor de frontieră la detecția validității videoclipurilor. Tabel din [9].

Model	Acuratețe	Precizie	Recall	F1
gpt-5.2-pro	56,38	**68,94**	54,54	44,84
gemini-3.1-pro	56,38	59,76	57,44	54,09
claude-opus-4.6	55,32	60,21	56,60	51,62
Juriu (maj.)	**58,51**	60,23	**59,21**	**57,70**

Capitolul 7

Concluzii și Direcții Viitoare

7.1 Sinteza Contribuțiilor

Această teză a investigat înțelegerea vizuală din perspectivă biologică și computațională, producând cinci contribuții:

Semnalele creierului și înțelegerea vizuală [10]. Înregistrările EEG din timpul recunoașterii vizuale au arătat că electrozii frontali asociați procesării cognitive de nivel superior contribuie mai mult la acuratețea clasificării decât electrozii occipitali plasați deasupra cortexului vizual. Combinarea EEG cu features vizuale profunde a îmbunătățit recunoașterea de la 91% la peste 97%, confirmând că semnalele cerebrale poartă informație complementară. Această observație ne-a motivat să căutăm reprezentări explicite și conceptuale ale conținutului vizual.

Reprezentarea GEST [21]. Graful Evenimentelor în Spațiu și Timp descompune scenele vizuale și descrierile textuale în evenimente conectate prin relații temporale, spațiale și semantice. GEST a depășit metricile consacrate de similaritate a textelor prin potrivire de grafuri, iar videoclipurile generate prin GEST au fost preferate de evaluatorii umani în 87% din cazuri în raport cu metodele neurale de referință.

Motorul GEST [7, 9]. Un sistem complet pentru generarea de videoclipuri cu mai mulți actori și adnotări dense la nivel de cadru, fără cost suplimentar. Motorul implementează orchestrarea temporală prin algebra intervalelor a lui Allen, cuplată cu satisfacerea constrângerilor în stil Floyd-Warshall, un sistem de chain-ID pentru identitatea obiectelor și colectarea multi-modală a artefactelor (RGB, segmentare la nivel de instanță, grafuri de relații spațiale, alinieri eveniment-cadru, descrieri în limbaj natural). Sistemul cuprinde aproximativ 100.000 de linii de cod, scrise de un singur cercetător pe parcursul a șase ani.

Generarea agentică text-spre-GEST [8]. Un sistem bazat pe LLM care separă planificarea narativă de impunerea constrângerilor, obținând o rată de succes end-to-end de 80% față de 0% pentru variantele bazate exclusiv pe LLM organizat în etape. Narațiunile agentice câștigă 79% dintre comparațiile de text și 74% dintre cele de video în raport cu metodele procedurale de referință. Cu Relation Subagents care populează tipurile de muchii

logice și semantice, aceasta este prima abordare care exercită întreaga capacitate expresivă a formalismului GEST.

Validarea experimentală [9]. Corpusul GTASA (938 de videoclipuri, 892 de milioane de relații spațiale) a evidențiat trei rezultate: videoclipurile generate de motor obțin 69% validitate fizică față de 18%/13% pentru VEO/WAN; pretraining-ul pe GEST depășește VEO și antrenarea exclusiv pe date reale pentru captioning video; iar video-encoders auto-supervizate codifică structura spațială semnificativ mai bine decât encoders vizuale ale VLM-urilor, în timp ce VLM-urile de frontieră obțin rezultate apropiate de șansă (56%) la detecția videoclipurilor fizic invalide.

7.2 Direcții Viitoare

Mai multe direcții extind această lucrare. Portarea pe GTA V prin FiveM ar reduce semnificativ diferența de domeniu vizual, păstrând în același timp garanțiile semantice. Modul de cameră cinematic, implementat la nivel de infrastructură, are nevoie de validare end-to-end. Rata de succes de aproximativ 50% în regim autonom a sistemului agentic poate fi îmbunătățită prin strategii mai bune de variație și prin detecția buclelor. Folosirea videoclipurilor generate prin GEST ca stimuli pentru colectarea de date EEG ar închide bucla înapoi spre fundamentele biologice ale înțelegerii vizuale explorate în Capitolul 2.

O direcție deosebit de promițătoare este folosirea GEST-urilor ca modalitate de condiționare pentru generatoarele neurale de video. Modelele actuale text-spre-video primesc un singur prompt textual ca intrare și trebuie să deducă singure întreaga dispunere spațială, ordonarea temporală și coordonarea dintre actori — ceea ce, după cum arată experimentele noastre, nu reușesc adesea să facă. Un GEST oferă tocmai informația structurată care lipsește din text: ce actori există, ce fac, în ce ordine și cu ce constrângeri temporale. Avem în vedere un pipeline în două etape: GEST-Engine generează un videoclip corect semantic, dar învechit vizual, iar un model neural (de exemplu, o rețea de translatare video-spre-video sau un model de difuzie condiționat de GEST) îl transformă în filmare fotorealistă păstrând conținutul semantic. Corpusul GTASA, cu tripletele lui de GEST-video-text împerecheate, oferă un set de antrenare natural pentru această abordare. Această direcție ar combina fidelitatea semantică a modelelor explicite de lume cu realismul vizual al generării neurale și ar contribui la explicabilitatea videoclipurilor generate prin AI: fiindcă specificația GEST este lizibilă pentru oameni, iar fiecare cadru poate fi urmărit până la un eveniment specific din graf, conținutul generat ar deveni interpretabil și auditabil într-un mod în care generarea pur neurală nu este.

7.3 Considerații Finale

Această teză a pornit de la întrebarea cum funcționează înțelegerea vizuală — atât în sistemele biologice, cât și în cele computaționale — și a ajuns la un sistem pentru generarea, adnotarea și evaluarea conținutului vizual prin reprezentări structurate de evenimente. Drumul de la citirea semnalelor cerebrale din Capitolul 2 la generarea de videoclipuri narative cu mai mulți actori din Capitolul 4 poate părea ocolit, dar urmează un singur fir: convingerea că înțelegerea vizuală, în esență, este despre evenimente — cine ce a făcut, unde, când și de ce — nu despre pixeli.

Reprezentarea GEST întruchipează această convingere. Plasând evenimentele și relațiile dintre ele în centru, în loc de cadre și features, GEST oferă un nivel de descriere la care vederea computațională și limbajul natural se întâlnesc firesc, la care generarea și înțelegerea împart același vocabular și la care diferența dintre ceea ce o mașină poate produce și ceea ce poate explica se reduce la zero. Fiecare videoclip pe care îl produce GEST-Engine este complet explicat de graful care l-a generat; fiecare adnotare este exactă pentru că provine din modelul lumii, nu dintr-un efort de etichetare ulterior.

Credem că această abordare — modele explicite de lume cu specificații structurate de evenimente — va deveni din ce în ce mai importantă pe măsură ce domeniul trece de la generarea de clipuri scurte la producerea de narațiuni lungi, coerente, cu mai mulți actori. Generatoarele neurale vor continua să se îmbunătățească în privința realismului vizual, dar fidelitatea semantică va necesita exact acel tip de structură formală pe care o oferă reprezentări precum GEST. Cele două paradigme nu sunt concurente, ci complementare: modelele explicite generează datele structurate și garanțiile semantice, iar modelele implicite generează realismul vizual și bogăția perceptuală.

Instrumentele pe care le-am construit — reprezentarea, motorul, generatorul agentic, corpusul și metodologia experimentală — sunt publice. Sperăm că vor servi nu doar ca infrastructură de cercetare pentru comunitatea înțelegerii video, ci și ca fundament pentru viziunea pe termen lung de a apropia percepția, limbajul și gândirea prin reprezentări vizuale structurate.

Bibliografie

[1] J. F. Allen, „Maintaining knowledge about temporal intervals”, în *Communications of the ACM* 26.11 (1983), pp. 832–843.

[2] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus et al., „V-jepa 2: Self-supervised video models enable understanding, prediction and planning”, în *arXiv preprint arXiv:2506.09985* (2025).

[3] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge et al., „Qwen3-vl technical report”, în *arXiv preprint arXiv:2511.21631* (2025).

[4] M. J. Black, P. Patel, J. Tesch și J. Yang, „Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion”, în *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8726–8737.

[5] D. Chen, T. Kasarla, Y. Bang, M. Shukor, W. Chung, J. Yu, A. Bolourchi, T. Moutakanni și P. Fung, „Action100M: A Large-scale Video Action Dataset”, în *arXiv preprint arXiv:2601.10592* (2026).

[6] S. Chen, X. He, L. Guo, X. Zhu, W. Wang, J. Tang și J. Liu, „Valor: Vision-audio-language omni-perception pretraining model and dataset”, în *arXiv preprint arXiv:2304.08345* (2023).

[7] N. Cudlenco, M. Masala și M. Leordeanu, „[Tiny Paper] GEST-Engine: Controllable Multi-Actor Video Synthesis with Perfect Spatiotemporal Annotations”, în *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026, URL: https://openreview.net/forum?id=uUofPYVMZH.

[8] N. Cudlenco, M. Masala și M. Leordeanu, *Agentic Video Generation: From Text to Executable Event Graphs via Tool-Constrained LLM Planning*, Submitted to the CVPR 2026 Workshop on Agentic AI for Visual Media, 2026, arXiv: 2604.10383 [cs.CV], URL: https://arxiv.org/abs/2604.10383.

[9] N. Cudlenco, M. Masala și M. Leordeanu, *GTASA: Ground Truth Annotations for Spatiotemporal Analysis, Evaluation and Training of Video Models*, Submitted to the European Conference on Computer Vision (ECCV 2026), 2026, arXiv: 2604.10385 [cs.CV], URL: https://arxiv.org/abs/2604.10385.

[10] N. Cudlenco, N. Popescu și M. Leordeanu, „Reading into the mind's eye: Boosting automatic visual recognition with EEG signals", în *Neurocomputing* 386 (2020), pp. 281–292.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez și V. Koltun, „CARLA: An open urban driving simulator", în *Conference on robot learning*, PMLR, 2017, pp. 1–16.

[12] M. El Banani, A. Raj, K.-K. Maninis, A. Kar, Y. Li, M. Rubinstein, D. Sun, L. Guibas, J. Johnson și V. Jampani, „Probing the 3d awareness of visual foundation models", în *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21795–21806.

[13] R. W. Floyd, „Algorithm 97: shortest path", în *Communications of the ACM* 5.6 (1962), pp. 345–345.

[14] Google, „Veo 3 Model Card", în (2025), Accessed: March 04, 2026, URL: `https://storage.googleapis.com/deepmind-media/Model-Cards/Veo-3-Model-Card.pdf`.

[15] W. Hong, M. Ding, W. Zheng, X. Liu și J. Tang, „Cogvideo: Large-scale pretraining for text-to-video generation via transformers", în *arXiv preprint arXiv:2205.15868* (2022).

[16] A. G. Huth, W. A. De Heer, T. L. Griffiths, F. E. Theunissen și J. L. Gallant, „Natural speech reveals the semantic maps that tile human cerebral cortex", în *Nature* 532.7600 (2016), pp. 453–458.

[17] J. Ji, R. Krishna, L. Fei-Fei și J. C. Niebles, „Action genome: Actions as compositions of spatio-temporal scene graphs", în *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10236–10247.

[18] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev et al., „The kinetics human action video dataset", în *arXiv preprint arXiv:1705.06950* (2017).

[19] L. Khachatryan, A. Movsisyan, V. Tadevosyan, R. Henschel, Z. Wang, S. Navasardyan și H. Shi, „Text2video-zero: Text-to-image diffusion models are zero-shot video generators", în *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 15954–15964.

[20] M. Leordeanu și M. Hebert, „A spectral technique for correspondence problems using pairwise constraints", în *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, 1482–1489 Vol. 2, DOI: 10.1109/ICCV.2005.20.

[21] M. Masala, N. Cudlenco, T. Rebedea și M. Leordeanu, „Explaining vision and language through graphs of events in space and time", în *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2826–2831.

[22] M. Masala și M. Leordeanu, „From Vision To Language through Graph of Events in Space and Time: An Explainable Self-supervised Approach", în *arXiv preprint arXiv:2507.04815* (2025).

[23] R. Paiss, A. Ephrat, O. Tov, S. Zada, I. Mosseri, M. Irani și T. Dekel, „Teaching clip to count to ten", în *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 3170–3180.

[24] S. F. Popham, A. G. Huth, N. Y. Bilenko, F. Deniz, J. S. Gao, A. O. Nunez-Elizalde și J. L. Gallant, „Visual and linguistic semantic representations are aligned at the border of human visual cortex", în *Nature neuroscience* 24.11 (2021), pp. 1628–1636.

[25] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler și A. Torralba, „Virtualhome: Simulating household activities via programs", în *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8494–8502.

[26] Y. Qiu, Y. Nagasaki, K. Hara, H. Kataoka, R. Suzuki, K. Iwata și Y. Satoh, „Virtualhome action genome: A simulated spatio-temporal scene graph dataset with consistent relationship labels", în *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3351–3360.

[27] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. Iyer, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang et al., „Cambrian-1: A fully open, vision-centric exploration of multimodal llms", în *Advances in Neural Information Processing Systems* 37 (2024), pp. 87310–87356.

[28] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang et al., „Wan: Open and advanced large-scale video generative models", în *arXiv preprint arXiv:2503.20314* (2025).

[29] L. Wang, B. Huang, Z. Zhao, Z. Tong, Y. He, Y. Wang, Y. Wang și Y. Qiao, „Videomae v2: Scaling video masked autoencoders with dual masking", în *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 14549–14560.

[30] R. Wang, J. Yan și X. Yang, „Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching", în *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[31] J. Yang, W. Peng, X. Li, Z. Guo, L. Chen, B. Li, Z. Ma, K. Zhou, W. Zhang, C. C. Loy et al., „Panoptic video scene graph generation", în *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18675–18685.

[32] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li et al., „Videollama 3: Frontier multimodal foundation models for image and video understanding", în *arXiv preprint arXiv:2501.13106* (2025).