Foveal Playground — Plan D: Natural segmentation and semantics

Nov 8, 2026 · @Luiz Velho

Question and scope

Question. Can the explorer build an object-level description of the scene from its own sensory data — segments that persist across fixations, and optionally labels — good enough for attention to use, without any oracle? TR-26-11 (Section 13) lists "natural segmentation and semantics, giving attention an object-level vocabulary without oracles".

The row changed. Segmentation / semantics: today *none in control; oracle labels for figures only* → natural segmentation, then object-aware attention. Renderer, oracle *correspondence*, triangulation and fusion stay fixed. The matcher stays an oracle: this plan replaces the oracle *labels*, not the oracle geometry.

Two parts, in order.

1. *Segmentation as a measurement.* Produce segments per fixation, lift them to the map and keep their identity across fixations. Evaluate against oracle labels after the run is frozen. The policy is untouched, so every gaze stays as in the baseline.
2. *Object-aware attention.* Only once part 1 is good enough: policies that use segments (finish an object before leaving it; stop when no object is open; look for objects of a named kind).

Out of scope. Natural stereo; moving the head; training new models. Pretrained models may be used, as optional dependencies.

A note on the baseline. This plan starts from the repository's Playground-2 baseline (356 fixations), not the v0 run reported in TR-26-11.

Starting point

The oracle labels today. `classroom_oracle1_render._assign_instance_ids` gives each renderable MESH, CURVE, SURFACE, FONT or META object in the scene its own Blender `pass_index`, 1, 2, 3, … in name order. Every point and surfel carries the Object Index of its pixel as `instance_id_oracle`. It is used only for the segmentation PLY and panel F of `overview.png`; no control step reads it (architecture, section 13).

A question to settle first. Object Index 0 is "geometry without a catalogue Object Index", and the explorer has a special correspondence branch for it. Classroom's chairs, desks, lamps and books come from linked libraries. If they are placed as collection instances, they are not in the renderable-object list and their surfels may all carry index 0. In that case the oracle cannot tell one chair from another, and it is not yet a usable ground truth for instance segmentation. This needs to be measured before anything is built on it.

What one fixation offers as sensory input.

input	shape	today	note
`rgb_L`	640 × 640 × 3, of which the 256 × 256 core is reconstructed	loaded	not bitwise reproducible under Cycles OPTIX
`rgb_R`	640 × 640 × 3	in the EXR, not loaded	needed for stereo-consistent segments
points `xyz`	up to 65,536 per fixation, in H0, with their core pixel	computed	deterministic; gives range and normals on the core grid
`core_valid`	256 × 256 bool	computed	where geometry was measured
the persistent map	about 17 M surfels at the end	computed	lets segments be tied across fixations

What the map already does for identity. `GlobalMap.fuse` associates each new point with an existing surfel within 12 mm. That association is exactly what is needed to say "this segment in fixation 120 is the same surface as that segment in fixation 37". Today `fuse` computes it and discards it.

Design

Three layers, each replaceable.

1. *Per-fixation segmenter.* `segment(obs, calib) -> labels`: a 256 × 256 int array over the core, −1 where `core_valid` is false. Implementations, from simplest to heaviest:
   - **G (geometric):** cut the core grid at range jumps and normal changes computed from `xyz`, then connected components. Deterministic, no new dependencies.

- **A (appearance):** superpixels on `rgb_L` (Felzenszwalb or SLIC), merged with G's depth edges.
- **L (learned, optional):** a pretrained class-agnostic segmenter on `rgb_L`; and, for semantics, an open-vocabulary labeller giving each segment a text label and score.

2. *Lifting and identity.* Each point inherits its pixel's segment. `GlobalMap.fuse(…, return_assign=True)` also returns the surfel index each point was fused into, without changing the map. Two segments from different fixations are merged (union-find) when they share at least k surfels and a minimum fraction of either segment. Each surfel keeps the final segment id; labels from L are kept as votes per segment.
3. *Map-level 3-D segmentation (post-hoc baseline).* On the frozen map: extract large planes (walls, floor, ceiling, desk tops) by RANSAC, then cluster the rest at about 2–3 cm. It shows what is possible when identity is solved globally rather than fixation by fixation.

The truth firewall. The segmenter receives a stripped observation: `rgb_L`, `rgb_R`, the core `xyz` grid, `core_valid` and the calibration. Position and Object Index are not in it. The test is fail-capable: pass a dict that raises on any oracle key and run every segmenter on it. Oracle labels are read only by `evaluate`, after `freeze.json` is verified, as in exercise E3.

Determinism. G is bitwise reproducible. A and L read RGB, which Cycles OPTIX does not reproduce run to run. That is harmless while segmentation is only measured. Once segments steer gaze (part 2), a policy that uses A or L loses bitwise reproduction. Either use G in control, or replace the bitwise check with a stated tolerance (same segments up to a boundary-pixel fraction).

Attention that uses segments (part 2). Each variant is one row change in the local or global policy, run first in Plan C's simulator:

id	idea	change
A1	finish the object	among eligible local candidates, prefer those whose footprint holds the open boundary of the current gaze's segment
A2	object-level stop	stop when no segment has an open boundary toward an unvisited lattice gaze

id	idea	change
A3	task-driven search	given a label ("chair"), rank gazes by the expected number of new segments of that label; report fixations until every instance is found

A3 is the Yarbus-style case from the report's background: the same scene, a different question, a different scanpath.

Work plan

Six milestones on branch `project/segmentation`. M0–M4 leave the trajectory exactly as the baseline; only M5 changes gaze.

1. **M0 — Audit the oracle labels.**
   - From a baseline `final-map.npz`: the fraction of surfels with index 0; surfels per catalogue object; which Blender objects (by name) the indices are.
   - If the furniture is index 0 (collection instances), build *evaluation-only* labels: a separate after-freeze render of the run's gazes with per-instance ids (for example a Cryptomatte object pass, or a copy of the scene with instances made real and numbered). The control render is not touched.
   - *Deliverable:* a label catalogue and a one-page note on what an "object" is in each scene.
2. **M1 — Interfaces and firewall (refactor).**
   - `segmentation.py` with the `segment` API and the stripped observation; `load_observation` also loads `rgb_R`; `fuse(…, return_assign=True)`.
   - *Checks that can fail:* the oracle-key guard test; a GPU smoke run and full run still PASS `compare.py check`, since nothing in control changed.
3. **M2 — Geometric segmenter G, lifting and identity.**
   - Develop offline on a saved run (Plan B's replay or Plan C's observation cache).
   - *Known answer:* on a synthetic scene of a box on a table in front of a wall, G must find three segments in a fixation that sees all three, and union-find must keep the box one segment across the fixations that see it.
4. **M3 — Map-level 3-D baseline.** Planes plus clustering on the frozen map; evaluated the same way. The gap between M2 and M3 shows how much is lost to fixation-by-fixation identity.

5. **M4 — Appearance and learned segmenters (optional dependencies).** A, then L, kept in `requirements-seg.txt` so the base install stays numpy + Pillow. Add open-vocabulary labels per segment and measure label accuracy on the catalogue's object names.
6. **M5 — Object-aware attention.** A1 and A2 in Plan C's simulator using G (deterministic), then A3 with the best labeller. One or two GPU confirmation runs, `checks` with the new known answers, and the report.

Evaluation

Segmentation is scored on the surfels of the frozen map against M0's labels, weighted by surfel count and, separately, by solid angle (so that near objects do not dominate). Attention is scored against the Playground-2 baseline in the simulator.

Segmentation (part 1).

measure	definition
ARI, AMI	agreement between natural segment ids and object labels over all labelled surfels
matched IoU	mean IoU of each object with its best natural segment (Hungarian matching)
over- / under-segmentation	objects split into more than one segment; segments covering more than one object
identity failures	objects whose surfels end in several final segments although each fixation segmented them correctly
boundary F-score	on the cyclopean sphere at 0.5°, natural boundaries against label boundaries
label accuracy (M4)	share of matched segments whose text label names the object's kind
cost	seconds per fixation for segmenting and lifting

Report per object kind as well (walls and floor, desks, chairs, small objects): planes are easy, thin chair legs are not.

Attention (part 2).

measure	definition
object switches	how often consecutive fixations land on different objects
fixations to complete an object	from first to last fixation that adds surfels to it
first-hit coverage at the stop	for A2, against fixations used (Plan C's measure)
fixations to find all instances	for A3, per label, against the baseline's order

Figures. The natural-segmentation PLY beside the oracle one; segment maps of a few fixations over their RGB; for A3, the scanpath for two different labels on the same scene.

Risks and open questions

- **No usable ground truth (M0).** If furniture shares index 0 and an evaluation-only label render is not possible, instance-level scores cannot be computed. Fall back to plane / non-plane and object-kind scores, and say so.
- **What counts as an object** is a modelling choice: a desk with its drawer, a chair's legs, books on a shelf. Fix the granularity in M0, before any score is computed.
- **The 12° fovea sees parts, not objects.** A chair seldom fits in one fixation. Identity across fixations (union-find through the map) will carry most of the burden; expect identity failures to dominate the error.
- **RGB nondeterminism** (A, L) breaks bitwise reproduction once segments steer gaze. Keep G for control unless the report states a tolerance.
- **Heavy dependencies and GPU sharing.** Learned models need PyTorch and GPU memory beside Blender on the same card. Keep them optional, load them once per run, and time them.
- **Over-merging through the map.** Two touching objects share surfels along their contact, so the merge thresholds (k, fraction) must be tested on the synthetic box-on-table scene before Classroom.
- *Open question:* should segment identity eventually feed back into fusion (never fuse across a segment boundary)? That changes fusion, so it would be its own experiment, after Plan B.

Deliverables and effort

About 8 student-weeks (ESTIMATE). It splits naturally into two projects: D1, segmentation (M0–M4), and D2, attention (M5), which needs Plan C's simulator.

milestone	deliverable	effort (ESTIMATE)
M0	label audit, evaluation-only labels if needed, granularity note	1 week
M1	`segmentation.py`, stripped observation, guard test, `return_assign`, `compare.py check` PASS	0.5–1 week
M2	segmenter G, lifting, union-find identity, synthetic known answer, scores	2 weeks
M3	map-level planes + clustering baseline, scores	1 week
M4	segmenters A and L, open-vocabulary labels, scores (optional)	1.5 weeks
M5	A1, A2, A3 in the simulator; GPU confirmation; report	1.5–2 weeks

Dependencies: M2 and M5 run fastest on Plan C's observation cache; the evaluation reuses the freeze firewall from Plan A's M1.