Foveal Playground — Plan A: New scenes

Nov 8, 2026 · @Luiz Velho

Question and scope

Question. Does the Greedy Foveal Explorer's behaviour in the Classroom transfer to other static scenes, with every other component held fixed? TR-26-11 (Section 13) lists this as "new scenes, to test whether the baseline's behaviour transfers".

The one row changed. Renderer → *other scenes* (Table 7 in the report, section 15 in `docs/architecture.md`). Matcher (PERFECT / ORACLE), triangulation, 12-mm fusion, the 1° coverage sphere, the 356-gaze lattice, the policy and the stop rule all stay as they are.

In scope. A scene registry so that a scene is a parameter, not a constant; a synthetic scene with a known answer; two to four public scenes of different kinds; a second head pose in Classroom; a cross-scene comparison report.

Out of scope. Changing the policy or fusion to suit a scene (that is a second experiment, after this one); moving heads; natural stereo.

A note on the baseline. TR-26-11 reports the v0 explorer (542 fixations, 322 local and 219 global). `visgraf/foveal-playground` ships the Playground-2 sparse lattice (356 fixations, 315 local, 40 global). This plan compares against the repository's Playground-2 baseline, since that is what students run.

Starting point

The repository baseline in Classroom stops by itself at fixation 356 (315 local, 40 global saccades), with 100 % SEEN, 99.26 % DEPTH and 16,943,073 surfels, in about 7 minutes on an RTX 4090 (MEASURED in f3d-vision).

One consequence to plan around. Under Playground-2 the fixation count is the lattice size in *every* scene. The lattice has no redundant gaze and the stop fires only when no lattice gaze adds unseen area, so all 356 gazes are always fixated. "Fixations to stop" therefore cannot show transfer. What can: the order of fixations (the SEEN / DEPTH growth curves), the local / global split, DEPTH relative to what is visible, map size and time.

Where the code is tied to Classroom. Making a scene a parameter touches these places:

file	what is fixed today	change
`tools/greedy_foveal/run.py`	`BLEND = scenes/classroom/classroom_eye.blend`	take it from the scene record
`tools/greedy_foveal/head_pose.py`	`H0_ORIGIN_W_M = (-0.6, -1.0, 1.2)`, `H0_R_WH`	read the declared pose from the scene record; keep Classroom's values as the self-check case
`tools/active_bootstrap/ab1a_spec.py`	`BLEND`, `VERGENCE_M = 2.10`	`BLEND` unused by the explorer; vergence stays 2.10 m for now (see Design)
`tools/classroom_oracle1_public.py`	`SCENE` path	unused by the explorer; leave as is
`scripts/setup_scene.sh`	Classroom URL, md5, EYE location	`setup_scene.sh <scene>`, driven by the scene record
`scenes/classroom/ASSET_MANIFEST.json`	one manifest	one manifest per scene
`tools/greedy_foveal/compare.py`	`KNOWN_ANSWER` and the signature are Classroom's	known answers per scene, or `check` only for Classroom

What already generalises: `place_eye.py` places an EYE in any `.blend` (`--dry-run` prints the bounds and cameras; `--from-camera` starts from the shot camera); `_assign_instance_ids` numbers every renderable object; `scene_preflight.py` checks the EYE, linked files and OPTIX for whatever scene it opens; and `ab1a_render.build_rehearsal_scene` already builds a synthetic scene from factory start-up.

Design

A scene is a record, not a constant. Each scene gets `scenes/<name>/scene.json`: name; source URL, md5, license and author; the generator `.blend` and the EYE `.blend`; the EYE placement given to `place_eye.py` (location, yaw); and the declared head pose (origin in W, `R_wh`). `run.py explore --scene <name>` (default `classroom`) reads it, starts the render server on that `.blend`, checks the head against that pose, and writes the record's name and digest into `trajectory.json` under a new top-level `scene` key. Scenes are downloaded by the setup script and never committed, as Classroom is now.

Scene families to test. Each family stresses one assumption the baseline makes implicitly.

family	example	what it tests	expectation to check
synthetic known-answer room	a closed box with a few primitives, built by script	the whole pipeline on a scene with an analytic answer	DEPTH ≈ 100 % minus binocular-occlusion slivers next to depth edges
same scene, new vantage	Classroom from 2–3 other head poses	sensitivity to where the head is, nothing else	similar local / global split; different global tail
another interior	a public Blender demo interior of similar scale	transfer at the same metric scale	curves close to Classroom's
close range	a tabletop scene, surfaces at 0.4–1.5 m	vergence fixed at 2.1 m; fine point spacing	DEPTH loss below about 0.35 m (ESTIMATE, below)
open scene	an outdoor or large hall scene with sky	gazes with no hits; ranges past 15 m	lower edge support, more global jumps; DEPTH bounded by the visible fraction
hard materials	glass, mirrors, thin structure (plants, chair legs)	what the oracle calls a surface	holes or spurious depth where Position disagrees with appearance

What is held fixed, and where it is scale-dependent. Everything stays at its baseline value, so that differences come from the scene. Two parameters are tied to Classroom's metric scale. The project records their effect rather than tuning them:

- *Fusion radius 12 mm.* A pixel subtends 1 / 1217.84 rad, about 0.82 mm per metre of range. At 2 m points are 1.6 mm apart; past about 15 m they are more than 12 mm apart, so overlapping fixations stop fusing and the map grows with overlap (ESTIMATE).
- *Vergence 2.10 m* (`ab1a_spec.VERGENCE_M`). The right-eye raster leaves about 8.7° of margin around the 12° core. With a 63-mm baseline the extra disparity of a near point exceeds that margin closer than about 0.35 m, and those points get no correspondence (ESTIMATE).

Limits of the oracle in new scenes. The matcher uses the Position and Object Index passes. Volumes, particles, hair and transmissive surfaces may give Position values that differ from what a camera would match. The hard-materials scene is there to find out, not to be fixed in this project.

Determinism. Each new scene needs its own reproducibility check before any comparison: two identical runs on the same machine must agree bitwise on the control path, as Classroom does.

Work plan

Six milestones on branch `project/new-scenes`. M0 is a refactor and must leave Classroom bitwise unchanged; everything after it is an experiment.

1. **M0 — Scene as a parameter (refactor).**
   - Add `scenes/classroom/scene.json` with today's values; add `head_pose.declared(scene)` and keep the module constants as Classroom's.
   - `run.py explore --scene` (default `classroom`); pass `--scene` to `render_server.py` so its head check uses the same record; record `scene` in `trajectory.json`.
   - Generalise `scripts/setup_scene.sh <scene>` and `scene_preflight.py` to read the record.
   - *Check that can fail:* the 75-fixation smoke run and the full run both PASS `compare.py check`; `head_pose.py` self-test rejects Classroom's pose under another scene's record.
2. **M1 — First-hit reference (exercise E3).** Cross-scene numbers need a denominator: what is visible from the head at all.

- `run.py evaluate` renders an equirectangular Position (+ Object Index) panorama from the head origin at 0.5° (720 × 360), once per scene and head pose, oriented as H0.
- Firewall: `evaluate` verifies `freeze.json` before it opens the reference.
- Metric: solid-angle fraction of first-hit cells with a surfel within 12 mm (`GlobalMap.nearest`, `row_weights`).
- *Check that can fail:* on a synthetic plane the metric returns the known fraction; with a 10° hole cut from the map it drops by the hole's solid angle.

3. **M2 — Synthetic known-answer scene.**
   - `tools/scenes/build_box_room.py` (Blender) writes a closed room with a few primitives and an EYE; no download.
   - Run the full baseline there; compare DEPTH and first-hit coverage with their analytic values.
   - Add a *range sweep*: a single plane at 0.2, 0.3, 0.5, 1, 2, 5 and 10 m. Measure DEPTH and surfels per fixation against range, testing the 0.35-m and 15-m estimates.
   - *Check that can fail:* the same room with one wall removed must lose the matching solid angle of DEPTH.
4. **M3 — Classroom from other vantage points.** Two or three new head poses chosen with `place_eye.py --dry-run` (for example near the blackboard and at the back of the room). Same scene, same materials: the cleanest test of placement alone.
5. **M4 — Public scenes.** Pick two or three scenes from the families in Design, with licenses that allow download-and-use (CC0 or CC-BY, credited in `scene.json`).
   - For each: a manifest, the preflight, then two identical 75-fixation runs that must agree bitwise on the control path (`compare.py`'s per-fixation digests, run against each other).
   - Then one full run per scene.
6. **M5 — Comparison.** The table and figures in Evaluation, with the commit, command and run directory for every number.

Evaluation

The result is one comparison table, one row per scene and head pose, with Classroom (baseline) as the first row. All numbers are MEASURED from frozen runs. Sky and other no-hit directions are removed from the denominators by M1's reference.

column	definition	source
first-hit fraction	solid angle with a first hit from the head origin, % of 4π	M1 reference
DEPTH / first-hit	final DEPTH ÷ first-hit fraction	`coverage.npz`, reference
first-hit coverage at 12 mm	% of first-hit solid angle with a surfel within 12 mm	`evaluate`
fixations to 90 / 99 % of final DEPTH	where the DEPTH growth curve crosses each level	`depth_history`
local / global saccades	counts; global jumps after the first one	`trajectory.json`
mean global jump	mean angle of global saccades, degrees	`trajectory.json`
surfels, and surfels per steradian of first hit	map size, normalised by what is visible	`final-map.npz`
render s per pair; total wall s	scene cost and run cost	`seconds_sum`

Figures. DEPTH growth curves for all scenes on one axis, normalised by each scene's first-hit fraction; the `overview.png` of each run; and the M2 range sweep (DEPTH and surfels per fixation against range).

How to read it. Transfer holds if the normalised growth curves and the local / global split stay close to Classroom's. The likely failures are named in advance: open scenes losing edge support (more global jumps), close scenes losing DEPTH below the vergence margin, far scenes inflating the map past 15 m.

Controls. M3 (same scene, new vantage) separates placement from scene content. M2's known-answer room checks the pipeline before any real scene is trusted.

Risks and open questions

- **Licenses.** Many demo scenes are CC-BY or CC-BY-SA, not CC0. The setup script downloads them and `scene.json` carries the credit; nothing is redistributed.

- **Scene hygiene.** Demo files often hold animation, keyframed seeds (Classroom keyframes `cycles.seed`, which `pin_seed` repairs), particles, hidden helper objects or missing textures. The preflight and the two-run determinism check catch most of this; budget a day per scene.
- **Render cost.** Heavier scenes may take several times Classroom's 0.75 s per pair. Time a 5-fixation run before a full one.
- **Object Index limit.** `_assign_instance_ids` refuses 32,767 or more renderable objects; very large scenes may hit it.
- **Oracle semantics.** For glass, volumes and hair, it is not settled what the Position pass should count as "the surface". Report what happens; do not fix it here.
- **Cross-machine determinism** is not measured. If students run on different GPUs, compare runs only within one machine.
- *Open question:* should the vergence distance become a scene parameter (for example the median range of the scene)? That changes the measurement, so it would be a separate experiment after this one.

Deliverables and effort

About 6–7 student-weeks and roughly 20 GPU runs, most under 15 minutes each (ESTIMATE).

milestone	deliverable	effort (ESTIMATE)
M0	scene registry, `--scene`, generalised setup and preflight; `compare.py check` PASS	1 week
M1	`run.py evaluate` with the freeze firewall and its known-answer test	1–1.5 weeks
M2	`build_box_room.py`, range sweep, known-answer results	1 week
M3	2–3 Classroom head poses with full runs	0.5 week
M4	2–3 public scenes: records, manifests, determinism checks, full runs	2 weeks
M5	comparison table, normalised growth-curve figure, short report	1 week

M1 is shared infrastructure: Plans B, C and D also need a first-hit evaluation, so it is worth doing first and merging to `main`.