

Two Rulers

The making of two faithful Blender models from two AI concepts



A rangefinder camera and a lever espresso machine started as single GPT Image 2 concepts on Scenario. Agent teams under one Claude Opus 5.5 supervisor rebuilt each in Blender, part by part. Neither team modelled from a drawing: one measured a generated 3D mesh, the other measured the concept image itself.

Drawn views drift

Both builds first asked the image model for orthographic views



Espresso: front, side and top drawn one at a time; the top was rejected



Rangefinder: a hero and six views drawn in one 3840 x 2160 image

For the espresso machine, GPT Image 2 drew front, side and top views one at a time. Every full edit moved something else, and the lever's side plate faced the camera in both the front and the side view. The route was dropped after about 40 minutes and 620 CU.

For the rangefinder, one image kept the style together, but a script that segmented the six views and compared their widths, heights and depths found no consistent sheet among three candidates: up to 32.4 % depth spread, and side views mirrored the wrong way.

Ruler A: a generated mesh

Three image-to-3D models on Scenario, one chosen as a measuring guide



Tripo P2: 240 CU, lever floats off its pivot



Hunyuan 3D 3.1 Pro: chosen, 0.92 overlap with the concept

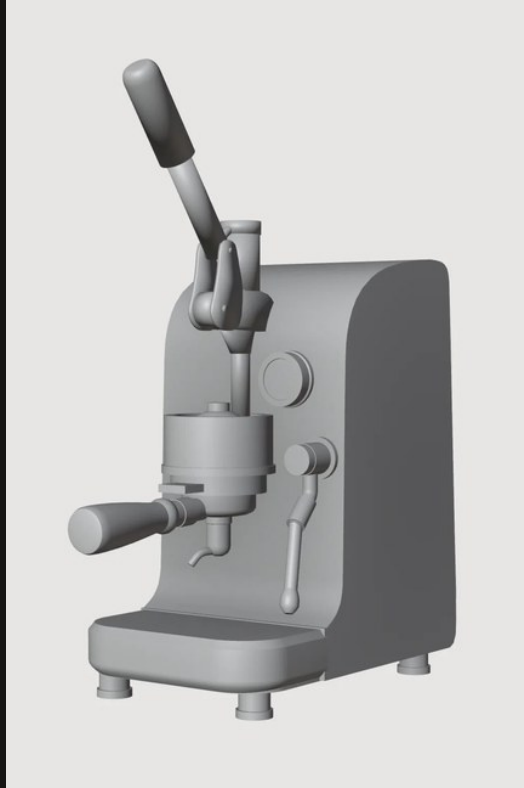


Hitem3D 3.0: densest, but invents parts

The espresso concept ran through Tripo P2, Hunyuan 3D 3.1 Pro and Hitem3D 3.0 at once (770 CU, 11 minutes). The Hunyuan mesh had a connected linkage, a real dial and no invented parts. It was scaled so the tower stands 0.40 m and rendered in fixed orthographic views at 0.3 mm a pixel. The agents measured it and were never allowed to copy, snap to or join its geometry.

Four gates on the espresso machine

Three agents (body, brew, lever) and an independent reviewer



G1 blockout: 0.924



G2 primary forms: 0.945



G3 detail and review: 0.945

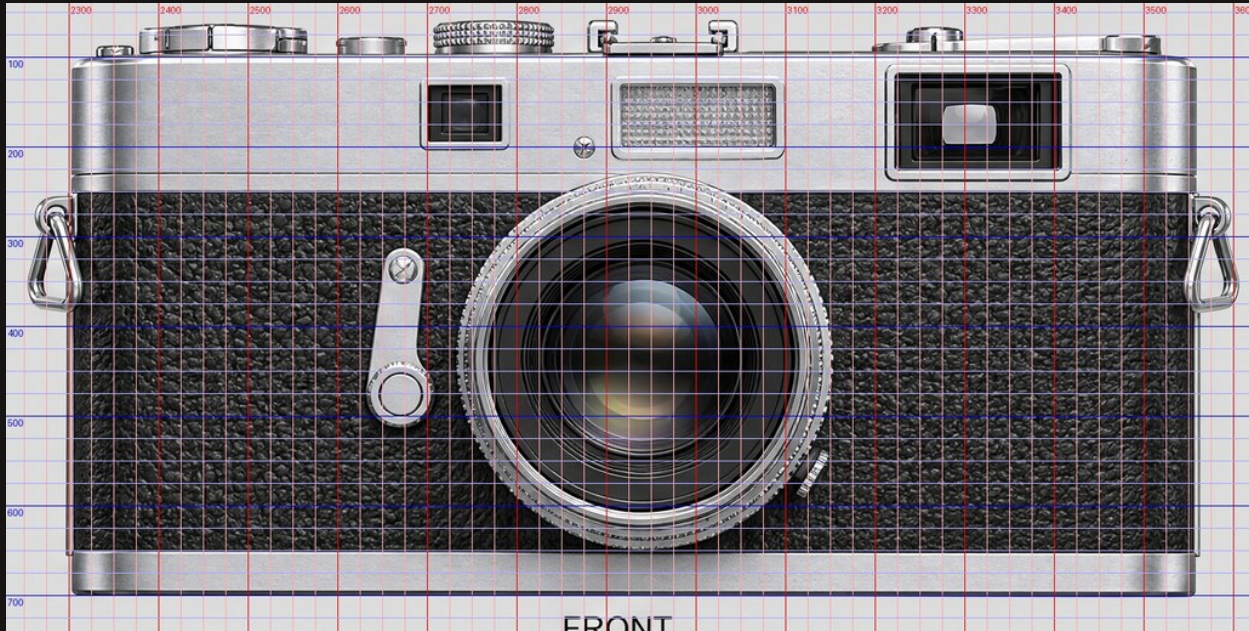


G4 look in Cycles: 0.945

Each gate scored five orthographic views against the mesh and the concept through a camera fitted to it (overlap 0.935 for the fit). Where the concept and the mesh disagreed, on the feet, handle and lever reach, the concept won. The reviewer audited 66 objects: 0 blockers, 5 majors, 10 minors, all fixed and re-gated. Four gates took 3 h 15 on the clock and 5 h 12 of agent time.

Ruler B: the concept itself

No mesh generator, not even as a measuring proxy



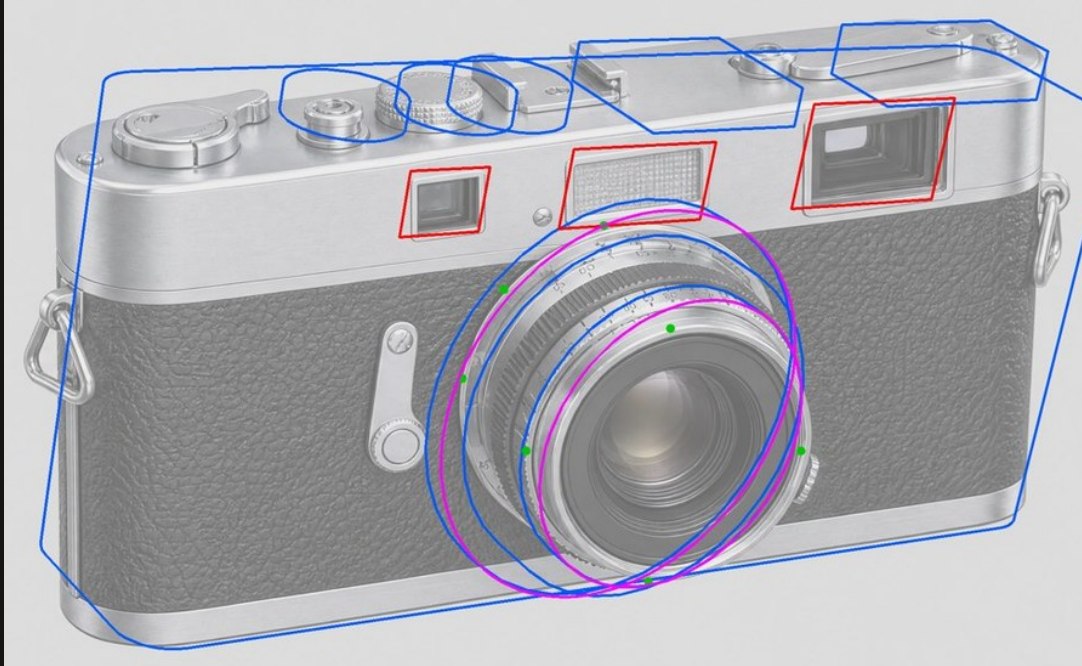
The sheet's own front view puts the windows, lens and dials about 22 mm left of the hero.

The rangefinder brief allowed Blender modelling only. So the hero three-quarter view of the chosen sheet became the only judge, and the orthographic views were kept as style references for sides the hero cannot show.

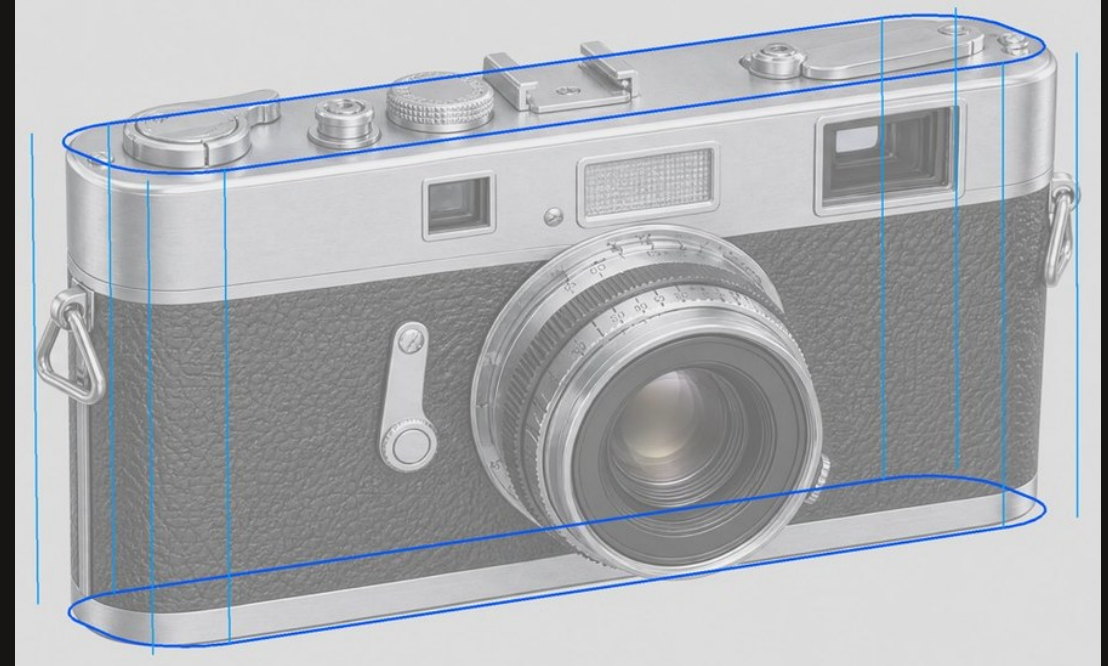
The decision mattered: once the hero was measured, the sheet's front view turned out to place the rangefinder window 22.8 mm, the illuminator 22.0 mm and the lens axis 23.3 mm away from where the hero draws them. A model built from it would have been a faithful copy of a different camera.

Solve the concept's camera

Read the picture before optimising it



Free optimisation cheated: sheared body, rings on the wrong ellipses

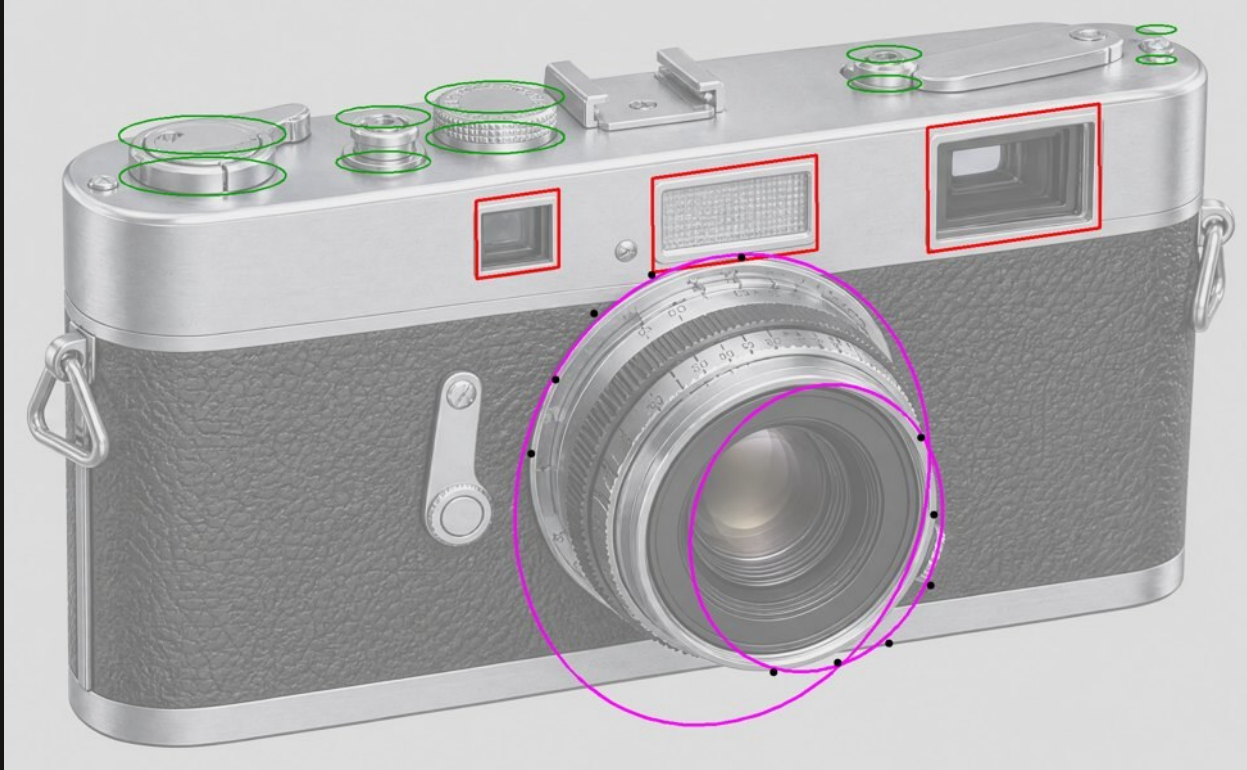


Solved: body outline overlap 0.993

A search over camera and layout together bent the geometry to fit the picture. The hero itself gave the answer: both ends of the body have the same height in pixels and the verticals stay vertical, so it is near axonometric. That fixed azimuth 29.9, elevation 13.6 and about 12.5 px per mm, and a local search on the body outline alone reached 0.993.

Millimetres, then four part groups

One layout file, a blackout at 0.973, four agents in parallel



Windows back-projected onto the front plane, controls onto the top plane, rings onto the lens axis.

Every visible feature was back-projected onto the plane it sits on and written to one millimetre file, LAYOUT.json. A blackout of plain primitives, scored from the fitted Blender camera, already overlapped the concept at 0.973 before any detail.

Four agents then built the body, the lens, the top controls and the small fittings in parallel, one script and one file each. When their parts collided, the supervisor settled the conflict in LAYOUT.json and every agent rebuilt. The final 73 parts have 0 intersecting triangles.

Reviewed from the concept's camera

Three rounds, region by region



Round 1: the knurl dominated and the depth-of-field ring had no face



Final: chrome DOF ring with numerals, 150-tooth knurl, chrome focus tab

Each round cropped the same regions from the concept and from a Cycles render through the concept camera: lens, top controls, front windows and timer lever. The written review went back to the owning agent. The lens glass was solved by elimination: four test renders showed the studio leaking through the elements, and hiding the lights from transmission gave the deep black of the concept.

Same rule, two results

Concept on the left, Blender model on the right



Espresso: 67 parts, final overlap 0.945



Rangefinder: 73 parts, final overlap 0.967

Overlap is the silhouette IoU from each concept's own fitted camera, excess counted against it. The numbers are not a contest: the objects, the concepts and the rules differ. What both builds share is the rule that made them faithful: a reference that cannot drift, a camera fitted to the concept, and every gate scored from that camera.

Rebuild your own concept

The route both builds share

- 1 Generate the concept with Scenario. If you want orthographic views, draw them in one image and check them against each other with a script before trusting them.
- 2 Choose a ruler that cannot drift: a mesh from a Scenario image-to-3D model used only for measuring, or the concept image itself with a camera solved from it.
- 3 Write every dimension to one shared file in millimetres. Builders read it; they never re-measure.
- 4 Gate every stage through the concept's fitted camera: blockout, primary forms, detail, look. Rebuild a failed part instead of patching it.
- 5 Use the agent skill from this exploration to run the same route through Scenario MCP.

[Open the Two Rulers exploration](#)

[Connect your agent to Scenario MCP](#)

[Generate concepts and 3D meshes on Scenario](#)