TinkyVision: Let There Be Sight

Continuous, low-latency vision for any model.

Perslis Research

September 14, 2026 · 04:22 EDT (2026-09-14T08:22Z)

Technical report · **PILOT-READY** · model-agnostic sight · perception & witness — not the control authority

Abstract

The last wave of AI progress made models that can *describe* a picture. It did not make models that can *watch a machine*. A frontier vision model captions a snapshot; it does not observe a running system over time, at low latency, over a channel narrow enough to survive an air-gapped or bandwidth-starved deployment. This report describes **TinkyVision**, a vision layer that gives any model continuous sight by inverting the usual assumptions. Instead of shipping heavy pixels to a multimodal model on demand, TinkyVision samples the screen at a steady **one frame per second**, keeps a fixed **300-frame recycling ring** — five minutes of rolling visual memory that never grows — renders each frame as **terminal art** so that pixels become text any model can read, recovers on-screen text with an OCR pass, and exposes the whole stream through **MCP tools** so the model pulls sight as a first-class tool call inside its normal loop. Each frame carries a hash, so the stream doubles as provenance: you can prove what the model saw and when. The result is coarse but continuous, cheap, and portable — *light vision* that travels anywhere text travels, for exactly the environments where heavy vision cannot go. We describe the design, the streaming path, the MCP interface, and — plainly — the limits.

Honest framing up front: TinkyVision is a **perception and witness** layer, not the system's control authority. It makes a model *see*; it does not decide what the model may *do*. Fidelity is deliberately coarse — layout, state changes, and text, not pixel-perfect detail. Maturity is labelled **PILOT-READY**. Every claim below is written to be checkable.

1 The blind model problem

Ask any model — frontier or local — to operate a live machine and you hit the same wall: it cannot see the machine. It can reason about what *should* be on screen, but it has no eyes on what *is*. The industry's answer has been the vision model: encode an image, feed it to a multimodal transformer, get a caption. That answer is real and powerful, and it is also the wrong shape for operating systems in motion. Four properties make it so:

- **It is a snapshot, not a stream.** A vision call describes one instant. Operating a system is a temporal act — you watch a spinner resolve, a dialog appear, a value settle. Statelessness across calls means the model has no continuity of sight; every frame is a stranger.
- **It needs pixels on the wire.** Images are large. Sending full frames to a cloud vision model assumes bandwidth and an open egress path — precisely what constrained, isolated, or air-gapped environments do not have.
- **It requires the model to *be* multimodal.** A text-only model, a small local model, or a specialised reasoner is simply blind under this scheme. Sight becomes a privilege of a few heavy models.

- **It is heavy and high-latency.** Encode, upload, infer, return — per frame. At the cadence needed to *watch* something, the cost and latency are prohibitive.

The problem is not that vision models are weak. It is that "attach a vision model" answers "can it caption an image?" when the operational question is "can it *keep an eye on* a system, cheaply, continuously, and somewhere the network barely reaches?"

2 Design principles

TinkyVision starts from three commitments, each chosen so that sight becomes a bounded, portable primitive rather than a heavyweight service.

> **P1 — Bounded by construction.** Everything is fixed-size and predictable: a fixed cadence, a fixed ring, a fixed rendering budget. Nothing grows without limit, so cost and latency are known in advance and survive a small machine or a thin pipe.

> **P2 — Model-agnostic.** Sight must not require a multimodal model. If the view arrives as text, then any model — text-only, local, tiny — can read it. Vision becomes a property of the *channel*, not the model.

> **P3 — Provable.** Every frame is hashed and time-ordered, so the stream is not just perception but evidence: a witness record of what was on screen and when.

3 How it works: the mechanism we are showing

3.1 One frame per second

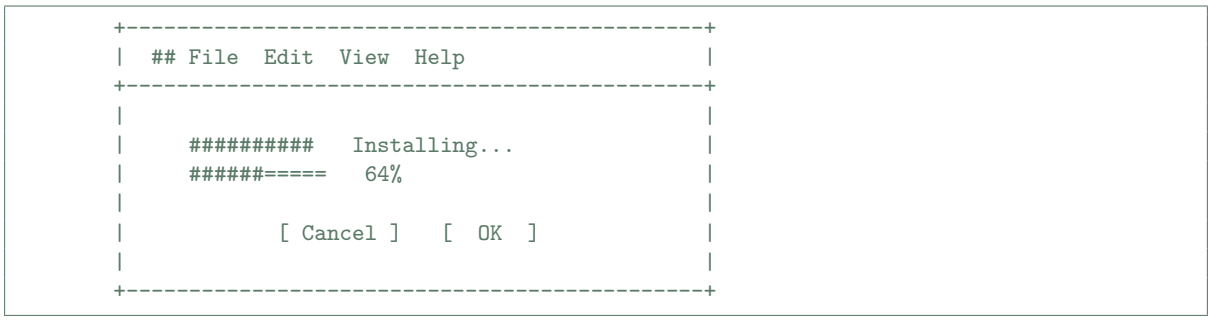
TinkyVision samples the target — a desktop, an application window, a VM guest — at a steady one frame per second. This is a deliberate cadence, not a limitation. One frame per second is enough to perceive the state changes that matter in an interface (a window opening, a value updating, a step completing) without flooding the model's context or the wire with redundant near-identical frames. The cadence is predictable, so a downstream consumer knows exactly how much data arrives per minute and can budget for it.

3.2 A 300-frame recycling ring

Frames land in a fixed ring of 300 slots. At one frame per second, that is **five minutes of rolling visual history** — and it *never grows*. Slot 301 overwrites slot 1. Storage is constant; there is no cleanup job, no unbounded log, no disk to fill. The model gains a short-term visual memory it can scroll back through: "what did the screen look like thirty seconds ago?" is a lookup, not a re-capture. Bounded memory is what lets the same design run on a workstation and on a constrained node without changing its footprint.

3.3 Terminal art — turning pixels into sight any model can read

This is the step that ends the blindness. Each frame is rendered as **terminal art**: a compact grid of characters (ASCII/ANSI) that preserves the layout, structure, and gross visual state of the screen. A frame stops being a blob of pixels and becomes a block of *text*. A model with no vision encoder at all can now "see" the screen, because the screen has arrived in the one modality every model already reads.



Illustrative terminal-art frame. A text-only model reads this as characters and perceives: a window with a menu bar, an install in progress at ∼64%, and two buttons — enough to know the state and decide the next action.

Terminal art is lossy on purpose. It carries *structure and state* — where things are, what changed, what phase the interface is in — not photographic detail. For operating a system, structure and state are usually what you need. Where exact on-screen text matters, an **OCR pass** over the same frame recovers the literal characters, and an accessibility read (where available) recovers named elements. Terminal art for the shape of the screen, OCR and accessibility for its words: together they compose functional sight.

3.4 Per-frame provenance

Every frame is hashed as it enters the ring. The hash plus the frame's timestamp turn the stream into an acquisition record: a verifiable answer to "what did the model see, and exactly when?" Perception and evidence are the same artifact. In a setting where a model's observations inform a consequential action, that receipt is the difference between "the model says it saw X" and "here is the hashed frame, timestamped, that shows X."

4 MCP tooling: vision as a first-class tool call

A stream is only sight if the model can actually look. TinkyVision exposes the ring through **MCP (Model Context Protocol) tools**, so looking is a normal tool call inside the model's existing reason-act loop — not an out-of-band pipeline the model has to be wired into. The model pulls exactly the view it needs, when it needs it:

Capability	What the model gets
Current frame	The latest captured frame of the target surface.
Terminal-art frame	The frame rendered as text — sight for a non-vision model.
OCR of a frame	The literal on-screen text recovered from the pixels.
Frame history	Scroll back through the ring — the last N seconds of the screen.
Find text / element	Locate a string or a named control on the current surface.
Accessibility read	Named, structured elements where the platform exposes them.

Because vision arrives as tool results, it composes with everything else the model does. A step in a plan becomes: *look (terminal-art frame) → read (OCR) → decide → act → look again.* The perception loop and the action loop are the same loop. And because each look is a bounded, text-shaped payload, the latency of "checking the screen" is the latency of a small tool call — not of encoding and shipping an image to a remote vision service.

The model no longer waits for a picture to be described to it. It asks to look, and looks — a character grid at a time.

5 Low-latency streaming where there would be no vision at all

The sharpest claim of this report is not that TinkyVision sees *better* than a frontier vision model. It is that TinkyVision sees *where a frontier vision model cannot go.*

Consider the payload. A full screenshot is hundreds of kilobytes to megabytes of image. A terminal-art frame is a few kilobytes of text — often less. That difference is not a marginal optimisation; it is a change of category. Text-shaped frames stream over channels that cannot carry images at all: a terminal session, a serial line, a text-only relay, a narrow or intermittent link. Where you could never sustain a video feed to a cloud vision endpoint, you can sustain a one-frame-per-second character stream indefinitely, because each frame is small and the ring is bounded.

This is what we mean by **light vision**: sight sized to survive the transport, not sight that assumes the transport is generous. The cadence is fixed, the frames are text, the memory is bounded — so the bandwidth and latency are predictable and low, and they stay that way for as long as the stream runs.

6 Air-tight environments: where light vision is exactly what is needed

The design pays off most in the environments that break the usual approach:

- **Air-gapped and isolated systems.** No open path to a cloud vision model, and often no graphical egress at all. Terminal art travels wherever a character can — including out of a locked-down box through a text channel that was never meant to carry a picture.

- **Bandwidth-starved and intermittent links.** Remote, embedded, or degraded connectivity where an image feed is a non-starter but a few kilobytes per second is fine.
- **Legacy and headless surfaces.** Old guests and machines whose only reliable channel is a terminal. If the model can read text, it can watch them.
- **Constrained and local models.** Deployments where the operating model is small, local, or text-only by design. Under TinkyVision they are not blind — sight rides the channel, so it does not depend on the model being multimodal.

In each case the alternative is not "worse vision." It is *no vision* — a model reasoning about a screen it cannot see. Light vision turns that into a model that watches, at a cost the environment can actually pay.

7 Honest limits

Sight this cheap and this portable comes with real trade-offs, and we state them plainly rather than let the demo imply otherwise.

- **Coarse fidelity.** Terminal art conveys structure and state, not photographic detail. Fine graphics, exact colours, and dense imagery are flattened. Where literal text matters, OCR recovers it — with OCR's own error rate — and where structure matters, accessibility reads help; but this is not a pixel-perfect inspection tool.
- **One frame per second misses sub-second events.** The cadence is chosen for bounded cost. Events faster than the sampling interval — a flash, a transient toast — can fall between frames. The ring gives you five minutes of history, not a high-speed capture.
- **Perception, not authority.** TinkyVision tells a model what is on screen. It deliberately does not decide what the model may do about it. Action gating, admission, and control belong to separate layers; conflating "the model can see" with "the model may act" would be a category error.
- **Provenance binds acquisition, not intent.** A per-frame hash proves what was captured and when. It does not, by itself, prove why an action followed. It is a witness record, and should be cited as one.
- **Maturity: PILOT-READY.** The capture, ring, terminal-art rendering, and MCP interface are working and in use; they are not presented as an enterprise-hardened, fully qualified product. Claims here are scoped to what has been built and observed.

8 Conclusion: let there be sight

A model that cannot see a running system is not a weak operator; it is a blind one, reasoning about a world it cannot observe. The reflex fix — bolt on a vision model — answers a different question than the operational one, and it answers it in exactly the environments that can least afford the pixels, the egress, and the multimodal requirement.

TinkyVision takes the other road. It makes *time* the primary axis — a steady one-frame-per-second stream with five minutes of bounded memory — and it makes *text* the medium, rendering the screen as terminal art so that sight rides the channel instead of the model. Delivered as

MCP tool calls and stamped with per-frame provenance, it turns “look at the screen” into a cheap, continuous, verifiable act available to any model, anywhere text can travel. Not sharper sight than a frontier vision model — *portable, continuous, provable* sight, in the places heavy vision never reaches.

The model was blind. Let there be sight.

TinkyVision technical report · Perslis Research · generated September 14, 2026 · 04:22 EDT (2026-09-14T08:22Z) · PILOT-READY · every claim labelled by how it was earned.