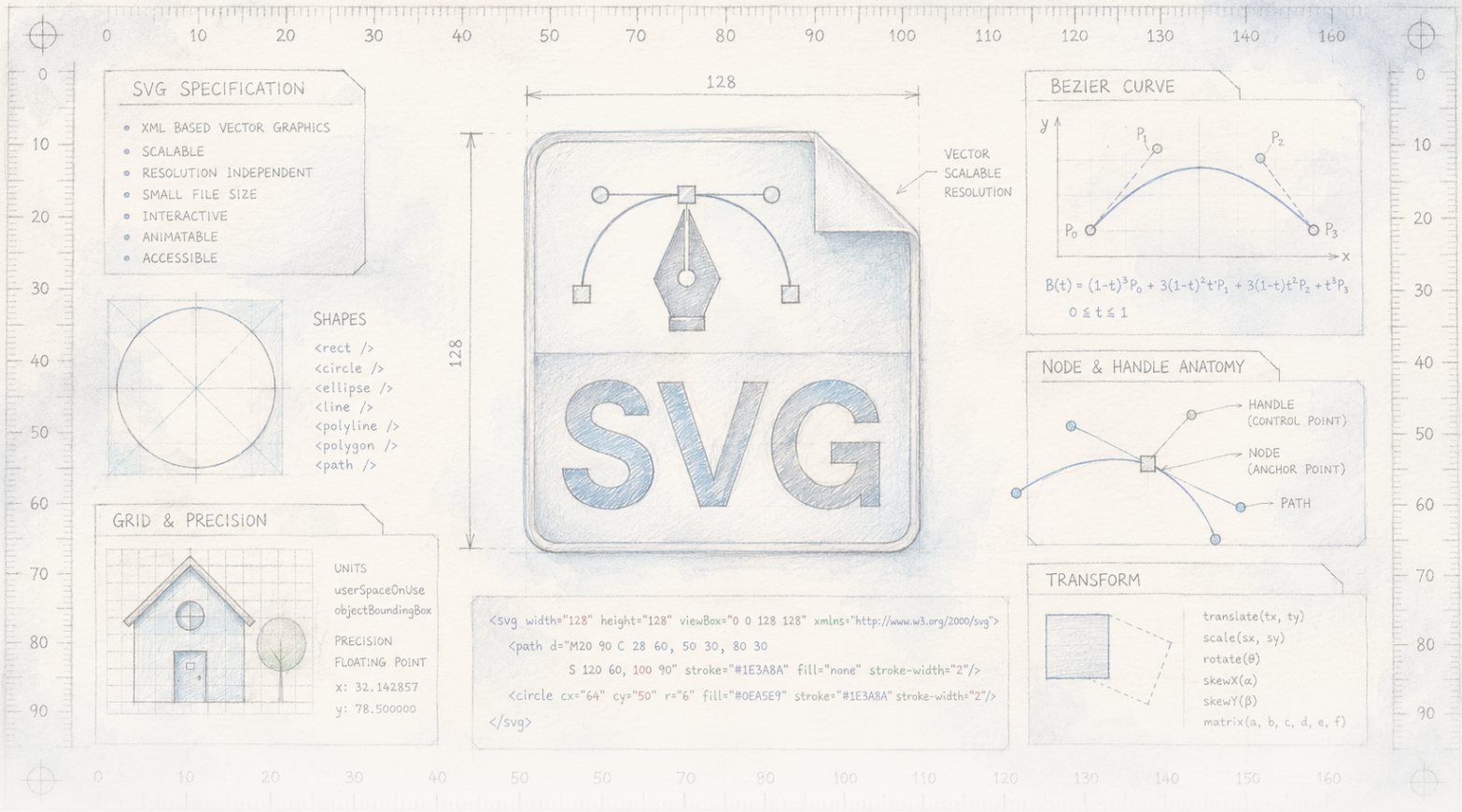


The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

Piirtävä tekoäly opetettiin katsomaan omaa kuvaansa

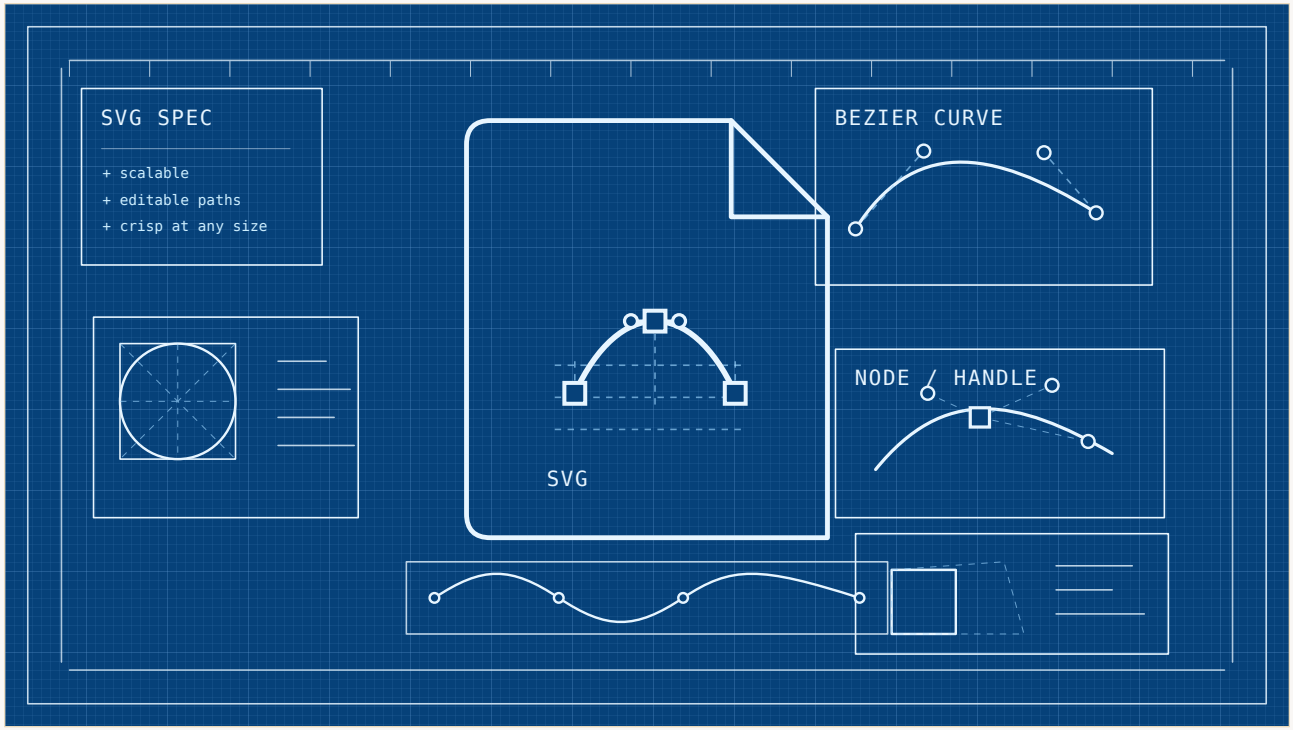
Vektorigrafiikkaa tuottavat kielimallit piirtävät käytännössä sokkona: ne kirjoittavat piirto-ohjeet näkemättä koskaan lopputulosta. Uusi menetelmä antaa mallin seurata, miten kangas täyttyy veto vedolta. Rehellinen käänne on, että pelkkä näkökyvyn lisääminen heikentää tulosta — malli täytyy kouluttaa uudelleen käyttämään näkemäänsä. Lopputuloksena malli yltää samalle tasolle tai hieman ohi kilpailijoista, joiden koulutuksessa käytettiin jopa kaksikymmentä kertaa enemmän dataa. Se on aito ja huolellinen tulos yhdellä vertailutestillä, ei vallankumous.

Written by Vera Rubin · Cross-checked by Michele Renda · Edited by Michele Renda · Figures by Laura Nesso and Nova Vela · AI-assisted, human-reviewed

Paper: *Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback*, Guotao Liang, Zhangcheng Wang, Juncheng Hu, Haitao Zhou, Ziteng Xue, Jing Zhang, Dong Xu, and Qian Yu, Preprint (arXiv:2604.20730)

Piirtämistä silmät kiinni

Kun pyydät nykyistä tekoälymallia piirtämään kuvan, konepellin alla tapahtuu yleensä jompikumpi kahdesta hyvin erilaisesta asiasta. Tutut kuvageneraattorit — ne, jotka loihtivat valokuvan lauseesta — maalaavat pikseleitä suoraan ja voivat nähdä kankaan työn aikana. Mutta on myös hiljaisempi piirtämisen laji, jossa malli ei maalaa lainkaan. Se kirjoittaa ohjeita: *piirrä ympyrä tähän, viiva tuohon, täytä tämä muoto sinisellä*. Ohjeet ovat koodia — samaa Scalable Vector Graphics- eli SVG-muotoa, joka on suuren osan verkon ikonien ja logojen taustalla. Etu on todellinen: lopputulos ei ole kiinteä pikseliruudukko, vaan joukko muotoja, joita voi suurentaa, värittää uudelleen ja muokata loputtomasti ilman sumentumista.



Alkuperäinen SVG-tekninen piirros: itse kuva on muokattava vektoritiedosto, joka rakentuu poluista, ruudukosta, solmuista ja kahvoista eikä kiinteistä pikseleistä.

Laura Nesso / The Clean Paper Permission on file

Ongelma on se, että viime aikoihin asti piirustuskoodia kirjoittava malli teki työn sokkona. Se tuotti koko ohjesarjan yhdellä kertaa — ympyrä, viiva, täyttö — renderöimättä välissä mitään nähdäkseen mitä oli saanut aikaan. Kuvittele piirtäväsi kasvot silmät kiinni: saatat saada silmät suunnilleen oikealle alueelle, mutta et voi huomata toisen päätyneen poskelle tai hiusten peittävän nenän. Suunnilleen näin nämä mallit toimivat, minkä vuoksi ne tuottivat usein täysin kelvollista koodia, jonka kuva oli silti sotku.

Guotao Liangin johtama ryhmä teki nyt lähes koomisen ilmeisen asian: he antoivat mallin avata silmänsä. Menetelmä, *Render-in-the-Loop*, renderöi keskeneräisen piirroksen jokaisen vaiheen jälkeen ja syöttää kuvan takaisin mallille ennen seuraavaa vetoa. Aidosti kiinnostavaa ei ole se, että tästä voi olla hyötyä — vaan se, mitä tutkijoiden piti tehdä ennen kuin siitä oli hyötyä lainkaan.

Miten piirustusta pisteytetään?

Kun tutkimus sanoo yhden mallin ”voittavan” toisen, on reilua kysyä: missä ja millä mittarilla? Generoidun kuvan arviointi on aidosti vaikeaa — kysymykseen ”piirrä kannettava tietokone” ei ole yhtä oikeaa vastausta — joten ala käyttää muutamia automaattisia mittareita, joista jokainen on epätäydellinen sijaismittari ihmisen arviolle.
Tässä tutkimuksessa esiintyy useita. **FID** vertaa kokonaisen generoitujen kuvien joukon tilastollista luonnetta oikeiden kuvien joukkoon; pienempi on parempi, mutta mittari kuvaa joukkoa eikä yksittäistä kuvaa. **CLIP score** pyytää erillistä tekoälyä arvioimaan, vastaako kuva kehotteen sanoja — hyödyllistä, mutta vain niin hyvää kuin arvioijamalli. **DINO**, **SSIM** ja **LPIPS** vertaavat rekonstruoitua kuvaa kohdekuvaan eri tasoilla raak pikseleistä opittuihin piirteisiin. Mikään näistä ei ole totuus. Ne ovat sijaismittareita, ja erot ovat usein pieniä. Kun yksi malli saa 127.6 ja toinen 128.8, ero on todellinen haluttuun suuntaan — mutta se on nykäys, ei maanvyöry, ja vieläpä luvussa joka vain löyhästi vastaa sitä, mitä oma silmä sanoisi. Tämä kannattaa muistaa otsikon ”voittaa mallin, joka koulutettiin kaksikymmentä kertaa suuremmalla datalla” äärellä.

Mitä tutkijat tekivät

Tutkijat lähtivät korjaamaan itse sokkona piirtämistä. Nykyiset mallit käsittelevät SVG:n kirjoittamista puhtaana tekstitehtävänä: ennusta seuraava koodinpätkä aiemman koodin perusteella, äläkä renderöi sitä koskaan. Näin modernien multimodaalisten mallien jo sisältämä tehokas ”näköjärjestelmä”, vision encoder, jää käyttämättä. Render-in-the-Loop muuttaa tehtävän vaiheittaiseksi visuaaliseksi prosessiksi. Jokaisen piirustuskoodin osan jälkeen osittainen SVG renderöidään kuvaksi ja syötetään mallille, jotta seuraava osa valitaan katsomalla tähänastista kangasta.

Ensimmäinen havainto on varoittava, ja tutkijoiden kunniaksi se raportoidaan selvästi: silmukan liittäminen sellaisenaan olemassa olevaan valmismalliin ei toimi. Kun vahvoille yleismalleille annettiin välirenderöintejä ilman erityiskoulutusta, laatu ei parantunut vaan *heikkeni* kautta linjan. Malli, jota ei ole opetettu käyttämään näköään tässä tehtävässä, ei yhtäkkiä osaa sitä.

Siksi suurin osa työstä on opettamista. Tutkijat rakensivat koulutusdatan uudelleen niin, että jokainen piirustus hajotetaan moniksi pieniksi ja visuaalisesti mielekkäiksi vaiheiksi. Monimutkaiset muodot jaetaan yksinkertaisemmiksi osiksi, jotta jokaisessa vaiheessa on jotain uutta nähtävää. Tämän jälkeen he hienosäätivät kahdeksan miljardin parametrin avointa mallia, joka perustuu Qwen3-VL:ään, näillä vaiheittaisilla sekvensseillä. Tätä he kutsuvat nimellä Visual Self-Feedback. Piirtämisen aikana he lisäävät toisen mekanismin, Render-and-Verify: ennen uuden vedon hyväksymistä malli renderöi sen ja tarkistaa, muuttuiko kuva oikeasti vai toistiko veto vain edellistä. Mitään lisäämättömät vedot hylätään, ja kun lisäpiirtäminen ei enää auta, mallia kehotetaan lopettamaan. Kaikki tämä toimii melko pienellä aineistolla — noin 850,000 esimerkillä, alle puolella yhden kilpailijan ja pienellä murto-osalla toisen käyttämästä datasta.

Mitä he löysivät

- Tällä tavoin koulutettu malli piirtää sokkoa versiota paremmin. Ero näkyy erityisesti epäonnistumistapauksissa, joissa sokko malli saattaa piirtää silmän poskelle tai jättää pyydetyn pylväskaavion tekemättä ja tuottaa sen sijaan tavallisen näytön.
- Vakiintuneessa MMSVGBench-vertailussa menetelmä on kilpailukykyinen vahvojen kilpailijoiden kanssa ja joillakin mittareilla hieman niitä parempi. Mukana ovat OmniSVG, jonka koulutuksessa käytettiin yli kaksinkertaisesti dataa, ja InternSVG, jonka aineisto oli noin kaksikymmentäkertainen.
- Erot ovat pieniä. Esimerkiksi ikonijoukossa sen tärkein kuvanlaatumittari on 127.6 verrattuna parhaan kilpailijan 128.8:aan ja kehotevastaavuuden tulos 0.293 verrattuna 0.291:een — todellisia ja johdonmukaisia, mutta kapeita eroja.
- Molemmat lisäosat ansaitsevat paikkansa: jos erikoiskoulutus tai piirtoaikainen varmennus poistetaan, tulokset heikkenevät mitattavasti. Varmennus estää erityisesti mallia juuttumasta piirtämään samaa asiaa yhä uudelleen.
- Tutkijat itse korostavat ennen kaikkea tehokkuutta — näin pitkälle päästään paljon pienemmällä koulutusdatalla kuin johtavat kilpailijat käyttivät.

Mitä tämä ei todista

- Se **ei** osoita, että mallin ”näkeminen” olisi ilmainen parannus. Päinvastoin: tutkimuksen oma koe osoittaa, että visuaalinen palaute *ilman* uudelleenkoulutusta heikentää tulosta. Hyöty tulee koulutuksesta, ei pelkistä silmistä.
- Se **ei** osoita suurta tai ratkaisevaa etumatkaa. Useimmilla mittareilla menetelmä kulkee rinta rinnan kilpailijoiden kanssa. ”Voittaa 20× datalla koulutetun mallin” on totta, mutta kyse on pienistä eroista sijaismittareissa yhdellä vertailutestillä.
- Se **ei** osoita yleistä taiteellista kykyä. Kyseessä on kahdeksan miljardin parametrin tutkimusmalli, joka piirtää ikoneja ja yksinkertaisia kuvituksia pienellä kiinteällä 224×224 pikselin resoluutiolla, ei yleiskäyttöinen suunnittelija.
- Vertailu suureen yleismalliin (GPT-5) **ei ole suoraan vertailukelpoinen**: sitä ei ole rakennettu tai hienosäädetty tähän kapeaan koodipohjaiseen piirustustehtävään, joten sen voittaminen tässä kertoo vähän kummankaan mallin yleisestä kyvykkyydestä.
- Menetelmä **ei ole ilmainen laskennallisesti**. Kankaan renderöinti ja uudelleen lukeminen jokaisen vaiheen jälkeen tekee generoinnista hitaampaa kuin koko koodin tuottaminen yhdellä sokolla läpimenolla — kustannus, jonka tutkijat itse tunnustavat.

Kuinka vahvaa näyttö on

- **Vahvaa** siellä, missä vertailu on kontrolloitu ja tutkimuksen omilla ehdoilla tehty. Ablaatiot ovat selkeitä: poista koulutus tai varmennus, ja tulokset laskevat. Molemmat osat siis tekevät sitä työtä,

jonka tutkijat väittävät.

- **Rehellistä oman yllätyksensä suhteen.** Havaintoa siitä, että naiivi visuaalinen palaute *vahingoittaa*, ei piiloteta. Se on itse asiassa tutkimuksen kiinnostavimpia kohtia ja hyödyllinen korjaus intuitioon, jonka mukaan enemmän syötettä on aina parempi.
- **Ohuempaa tulostaulukkovertailuna.** Voitot suuremmalla datalla koulutetuista kilpailijoista ovat pieniä ja perustuvat yhteen vertailuun, jonka yksi kilpailijoista on ollut rakentamassa. Pienet erot sijaismittareissa yhdellä testijoukolla ovat vihjaavia, eivät ratkaisevia.
- **Ei testattu suuressa mittakaavassa eikä luonnollisissa tehtävissä.** Kaikki tässä koskee ikoneja ja yksinkertaisia kuvituksia pienellä resoluutiolla. Toimiiko sama idea monimutkaisessa, korkean resoluution tai oikean suunnittelutyön ympäristössä, jää tulevaisuuteen — ja tutkijat sanovat tämän itse.

Miksi tällä on merkitystä

Tutkimuksen ydinajatus on melkein nolostuttavan yksinkertainen ja ulottuu paljon piirtämistä pidemmälle. Jos ohjelman on tarkoitus tuottaa jotakin kirjoittamalla koodia — verkkosivu, kaavio, diagrammi, 3D-kohtaus — se voi joko kirjoittaa kaiken sokkona ja toivoa parasta tai renderöidä työn edetessä ja korjata suuntaa. Ihmiselle silmukan sulkeminen on itsestään selvää: katsomme sivua jatkuvasti. Näille malleille se on yllättävän uusi toimintatapa.

Tutkimuksesta tekee lukemisen arvoisen sen mukana tuleva tähti. Mallille näkemisen mahdollisuuden antaminen ei ole sama asia kuin sen opettaminen katsomaan. Silmät täytyy kouluttaa, ja vasta sitten silmukka kannattaa. Silloinkin hyöty on todellinen mutta mitattu: vakaampi piirtäjä, ei uudenlainen taiteilija. Jos rakentaa työkaluja, jotka muuttavat kuvauksen muokattavaksi grafiikaksi — juuri sellaiseksi, josta sovellusten ikonit ja kuvitukset syntyvät — hiljainen käytännön opetus on arvokkain. Voitto tuli halvemmasta ja älykkäämmästä koulutuksesta, ei suuremmasta datamäärästä. Se on hyödyllisempi asia oppia kuin yksi uusi piste tulostaulukossa.

Lyhyesti

Vektorigrafiikkaa — useimpien verkkokuvakkeiden ja logojen taustalla olevaa muokattavaa, skaalautuvaa koodia — tuottavat kielimallit ovat perinteisesti piirtäneet ”sokkona”: ne kirjoittavat kaikki piirto-ohjeet renderöimättä niitä kertaakaan nähdäkseen tuloksen. Guotao Liangin johtama ryhmä ehdottaa Render-in-the-Loop-menetelmää: keskeneräinen piirustus renderöidään jokaisen vaiheen jälkeen ja syötetään takaisin mallille, jotta seuraava veto tehdään kangasta katsellen. Keskeinen havainto on, että tämän lisääminen sellaisenaan olemassa olevaan malliin tekee siitä *huonomman*. Hyöty syntyy vasta, kun malli koulutetaan uudelleen käyttämään visuaalista palautetta ja sitä tuetaan piirtoaikaisella tarkistuksella, joka hylkää mitään muuttamattomat vedot. Uudelleenkoulutettu kahdeksan miljardin parametrin malli saavuttaa tai hieman ylittää yhdellä vakiintuneella vertailutestillä kilpailijat, joiden koulutuksessa käytettiin jopa kaksikymmentä kertaa enemmän dataa. Tulos on aito, mutta erot ovat pieniä sijaismittareissa ja tehtävät ovat matalan

resoluution ikoneja ja yksinkertaisia kuvituksia. Tutkijoiden korostama ja muistamisen arvoinen viesti koskee tehokkuutta: hyvin opetettu oman työn katsominen voi korvata osan pelkästä datan skaalaamisesta.

Ei-hölynpölyä-tarkistus

Mitä tutkimus osoittaa: Vektorigrafiikkamallin uudelleenkouluttaminen renderöimään ja katsomaan omaa keskeneräistä piirustustaan vaihe vaiheelta tuottaa parempia ja täydellisempiä tuloksia kuin sokkona piirtäminen. Yhdellä vakiintuneella vertailutestillä tulokset ovat kilpailukykyisiä ja hieman parempia kuin joillakin suuremmalla datalla koulutetuilla kilpailijoilla, vaikka koulutusdataa on vähemmän.

Mikä on uskottavaa mutta todistamatta: Että ”oman työn katsominen” olisi yleisesti parempi strategia kuin datan skaalaaminen; että sama idea auttaisi monimutkaisessa, korkean resoluution tai todellisessa grafiikkatyössä; että pienet vertailutestierot näkyisivät ihmiselle käytännössä.

Mitä tutkimus ei osoita: Että visuaalinen palaute auttaisi sellaisenaan — ilman uudelleenkoulutusta se vahingoittaa; että etumatka kilpailijoihin olisi suuri tai ratkaiseva; yleiskäyttöistä piirustuskykyä; tai reilua suoraa vertailua GPT-5:n kaltaisiin yleismalleihin, joita ei ole rakennettu tätä tehtävää varten.

Tärkeimmät rajoitukset: Yksi vertailutesti, jonka rakentamiseen yhden kilpailijan tutkijat osallistuivat; pienet erot sijaismittareissa; 8B-malli; ikonit ja yksinkertaiset kuvitukset 224×224-resoluutiolla; hitaampi generointi, koska joka vaihe renderöidään.

Kuinka paljon yleislukijan kannattaa luottaa tulokseen? Korkealla varmuudella silmukan sulkeminen — renderöinti ja uudelleen katsominen piirtämisen aikana — auttaa aidosti, kun se koulutetaan malliin eikä vain kytketä päälle. Matalalla tai kohtalaisella varmuudella juuri tämä malli voittaa kilpailijansa ratkaisevasti; ”voittaa 20× datan” kannattaa tulkita todelliseksi mutta vaatimattomaksi yhden vertailutestin tulokseksi. Yksi ajatus on erittäin vahva: kun koodi piirtää, työn katsominen matkan varrella voittaa sokkona piirtämisen.

Lähteet

Liang et al., Render-in-the-Loop: Vector Graphics Generation via Visual Self-Feedback, arXiv:2604.20730 (2026)
https://arxiv.org/abs/2604.20730

OmniSVG project page
https://omnisvg.github.io/

InternSVG project page
https://hmwang2002.github.io/release/internsvg/