Engineer

GDAL

et bibliotek for å lese og konvertere geodata

Hva det er

GDAL, kort for Geospatial Data Abstraction Library og ofte skrevet GDAL/OGR, er programvare med åpen kildekode som åpner kartdata lagret i ett format og skriver dem ut i et annet, og som projiserer dem om underveis når du ber om det. Prosjektets egen dokumentasjon kaller det et oversetterbibliotek, en «translator library».

Romlige data kommer i to former. Rasterdata er et rutenett av celler: et satellittbilde, et flyfoto, en terrengmodell. Vektordata er punkter, linjer og polygoner med egenskaper knyttet til seg: adresser, veier, eiendomsgrenser. GDAL gir programmet som kaller det, én abstrakt modell for hver form, og bak modellen sitter en driver per format. GeoTIFF, JPEG 2000 og NetCDF er drivere på rastersiden. Shapefile, GeoJSON, GeoPackage, KML og tabeller i PostgreSQL er drivere på vektorsiden, og det finnes mange flere av begge slag. Et program skrevet mot GDAL leser alle sammen gjennom de samme kallene.

Vektorhalvdelen het i mange år OGR, og de to halvdelene hadde hver sin driverstruktur helt til GDAL 2.0 slo dem sammen. Den historien er grunnen til at vektorprogrammene fortsatt bærer det gamle prefikset: `ogrinfo` beskriver et vektordatasett, og `ogr2ogr` konverterer et. Arbeidet med biblioteket begynte sent i 1998 med Frank Warmerdam. Det er utgitt under en MIT-lignende lisens og er et prosjekt under Open Source Geospatial Foundation, OSGeo.

Du kan bruke det på tre måter. Den første er kommandolinjen, med programmer som `gdalinfo`, `gdal_translate` og `gdalwarp` for raster og `ogrinfo` og `ogr2ogr` for vektor. Siden versjon 3.11 finnes det også én samlet `gdal` -kommando med underkommandoer som `gdal raster` og `gdal vector`, som prosjektet fortsatt merker som foreløpig. Den andre er kode, gjennom bindinger for Python og andre språk. Den tredje er inne i annen programvare: QGIS leser for eksempel

vektorformatene sine gjennom GDAL-driverne og tilbyr GDAL-programmene i Processing-verktøykassen. En typisk konvertering får plass på én linje. Her blir en shapefil til en GeoPackage og projiseres samtidig om til UTM-sonen som brukes over hele Danmark:

```
ogr2ogr -f GPKG -t_srs EPSG:25832 parcels.gpkg parcels.shp
```

I skrivende stund er gjeldende versjon 3.13.3. Få verktøy konkurrerer på dette laget. ArcGIS har sine egne konverteringsverktøy, og enkelte formater har små lesere for seg selv, men et team som må ta imot det en kunde sender, ender gjerne opp med GDAL i stakken, ofte gjennom et bibliotek som bygger på det.

Bruk begrepet i en beslutning

Bruk «Hva er GDAL?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Alt som skjer på et sted — fartøy, kjøretøy, tomter, sensorer — lagt på et kart du kan søke i, måle på og dele.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Gjorde en geodatapipeline 50 ganger raskere — 50x raskere pipelines for geografiske data

Gjorde en geodatapipeline 50 ganger raskere Ta kontakt →

Hvorfor det er viktig

Alle geospatiale systemer har en grense der data utenfra kommer inn: en kommunes shapefiler, en landmålers GeoTIFF-filer, en partners GeoJSON-eksport, en KML-fil noen har tegnet i en nettleser. Beslutningen GDAL påvirker, er hvem som står vakt ved den grensen. Det kan være en person som klikker seg gjennom et skrivebordsverktøy, eller et skript som gjør det samme hver gang. GDAL er det som gjør skriptet billig å skrive.

En konvertering skrevet som en `ogr2ogr` - eller `gdalwarp` -kommando kan leses i en kodegjennomgang, legges i versjonskontroll og kjøres på nytt når neste leveranse kommer. En konvertering gjort for hånd blir gjort én gang, av den som husker stegene. For data som kommer i bolker, måned etter måned, avgjør den forskjellen om kvaliteten holder.

Lisensen teller også. Det er ingen avgift per bruker og ingen leverandør som må godkjenne noe, og det samme verktøyet kjører på en bærbar maskin, på en byggserver og inne i en container.

Den tredje grunnen er rekkevidden inn i lagringen. GDALs virtuelle filsystemer lar en driver lese data der de ligger. Med prefikset `/vsicurl/` leser den en fil over HTTP ved å be om byteområder, så de fleste drivere tar inn et eksternt datasett bit for bit uten å laste ned hele, og `/vsis3/` gjør det samme mot Amazon S3. Sett det sammen med en Cloud Optimized GeoTIFF, som er delt opp i fliser, har sine egne oversiktsbilder og er lagt opp for å holde antallet forespørsler nede, så kan et raster på mange gigabyte ligge i objektlagring og likevel leses ett vindu om gangen.

Kostnadene er også reelle. GDAL er et stort bibliotek i maskinkode med mange valgfrie avhengigheter, og hvilke drivere et gitt bygg inneholder, avhenger av hvordan det ble kompilert. To maskiner som begge «har GDAL», kan derfor åpne ulike formater. Og biblioteket oversetter trofast, noe som også betyr at det trofast tar med seg hver eneste begrensning i formatet det blir bedt om å skrive.

Slik ser det ut i praksis

GDAL dukker opp i to prosjekter på dette nettstedet, i to ulike roller.

Fra 2015 til 2022, som teamleder for GIS-programvareutvikling, gikk leveransene på mer enn tretti GIS-prosjekter, fra utvikling via produksjonssetting til drift. Under dem lå PostgreSQL med PostGIS for de romlige dataene, og GDAL/OGR var verktøyet for å flytte dem mellom formater: shapefiler, GeoJSON, GeoTIFF, KML og vektorfliser. Ved siden av GDAL sto QGIS og JOSM for selve dataarbeidet, og kartene på toppen ble bygget med Mapbox GL og Leaflet. Arbeidet bak den listen med formater spente fra digital kartlegging i 2D og 3D via LiDAR-behandling, georektifisering og vektorisering til innendørs kartlegging og navigasjon, og hvert prosjekt fikk driftsstøtte i produksjon etter at det var levert.

Fra 2022 til 2024, som dataingeniør på et storskala prosjekt for geospatial analyse, var GDAL ett av verktøyene for å finne og rette feil. Romlige feil ble diagnostisert og rettet med GDAL, QGIS og ArcGIS, og fordi de samme vaskeoppgavene kom tilbake gang på gang, ble de automatisert som Bash-skript. PostGIS bar de romlige spørringene og indeksene, prosedyrer i PL/pgSQL og Transact-SQL transformerte dataene inne i PostgreSQL og SQL Server, og

det rensede resultatet matet interaktive visualiseringer bygget med Mapbox. Over 1 000 problemer ble løst, og arbeidet bidro til 35 % kortere kjøretid for romlige spørringer.

Hvor det går galt

GDAL gjør nøyaktig det du ber om. Det meste av trøbbelet begynner når du ber om noe som er litt annerledes enn det du mente.

Punktene havner på feil kontinent. Siden GDAL 3.0 følger biblioteket den aksefølgen som instansen bak et koordinatsystem har fastsatt, og for EPSG:4326, det velkjente systemet med bredde- og lengdegrad, betyr det breddegrad først. Kode skrevet for GDAL 2 sender lengdegrad først, og det gjør GeoJSON også. Bland de to, så byttes hvert par om: et punkt i Oslo dukker opp igjen i Arabiahavet. Bestem én rekkefølge i kanten av programmet og skriv den ned. I kode som vil ha lengdegrad først, er den dokumenterte bryteren `SetAxisMappingStrategy(OAMS_TRADITIONAL_GIS_ORDER)` på det romlige referansesystemet.

Kolonnenavn blir kuttet, og datoer mister klokkeslettet. En shapefil begrenser navnene på egenskapene til 10 tegn. GDAL forkorter lengre navn mens det skriver, og når to forkortede navn kolliderer, kutter det dem til 8 tegn og legger til et løpenummer, slik at `abcdefghijkl` blir `abcdefgh_1`. Formatet lagrer datoer uten klokkeslett. GDAL anbefaler å holde `.shp`- og `.dbf`-filer på høyst 2 GB av hensyn til samvirket med annen programvare. Æ, ø og å eller kyrillisk tekst kan bli til støy hvis kodingen feiltolkes når `.cpg`-filen som angir tegntabellen, mangler. Skriv GeoPackage der mottakeren godtar det. Der en shapefil er påkrevd, gir du kolonnene nye navn med vilje før konverteringen.

Et eksternt raster lastes ned i sin helhet. Med en vanlig `https://`-adresse uten prefiks velger GDAL HTTP-pseudodriveren sin, som trekker hele filen inn i minnet før noen driver leser den. Det passer for en liten fil og er feil valg for et landsdekkende ortofoto. Sett `/vsicurl/` foran adressen, slik at driveren bare henter byteområdene den trenger, og lagre store raster som Cloud Optimized GeoTIFF, slik at de områdene blir få. La så filen være i fred: å redigere en COG på stedet ødelegger oppbygningen, og siden GDAL 3.8 blir et forsøk på å åpne en COG for oppdatering avvist, med mindre det bruddet er eksplisitt tillatt.

Tallene i en CSV kommer fram som tekst. CSV-driveren leser hvert felt som en streng så lenge den ikke får beskjed om noe annet, så en kolonne med folketall sorteres alfabetisk, og en sum lenger ned i løypa blir til en typefeil. Sett åpningsvalget `AUTODETECT_TYPE=YES`, eller legg en `.csvt` -fil ved siden av CSV-filen som angir typen til hver kolonne.

En oppgradering knekker Python-importen. `gdal` -pakken på PyPI bygges mot den GDAL-versjonen som allerede er installert på maskinen, så de to versjonene må stemme overens. Prosjektets installasjonsveiledning låser den ene til den andre:

```
pip install gdal[numpy]=="$(gdal-config --version).*"
```

Les mer

- GDAL documentation — prosjektets egen beskrivelse av oversetterbiblioteket, driverne og lisensen.
- Programs — alle kommandolinjeverktøyene, fra `gdalinfo` og `ogr2ogr` til den nyere `gdal` -kommandoen.
- GDAL Virtual File Systems — `/vsicurl/`, `/vsis3/` og de andre prefiksene for å lese data der de ligger.
- OGR Coordinate Reference Systems and Coordinate Transformation tutorial — med avsnittet om aksefølge og hva som endret seg i GDAL 3.0.
- COG – Cloud Optimized GeoTIFF generator — filoppbygningen, oversiktsbildene og hvorfor redigering på stedet ødelegger dem.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er GDAL?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/gdal/