

CloudShadingAI: die technischen Details

Das Training und der Cloud Shadow Service nutzen Python (<https://www.python.org/>).

1. Training (Modellentwicklung)

Das Training dient ausschließlich dazu, ein Modell zur Vorhersage von Movement Vectors (Wolkenbewegungen) zu entwickeln.

Inputdaten

- Wolkenbilder von Europa vom MSG-3 satellite, vorbereitet vom Projektpartner MetGIS
- Meteorologische Daten, vorbereitet von MetGIS (primär Winddaten) (15 Minuten Auflösung)
- Die Wolkenbilder enthalten kodierte Informationen zu: (15 Minuten Auflösung)
 - CTP (Cloud Top Pressure) → daraus ableitbar: Wolkenhöhe
 - COT (Cloud Optical Thickness) → Maß für die optische Tiefe / Lichtdurchlässigkeit

Dataprocessing

- Die Eingangsdaten werden in 18 Layer aufgeteilt
- Für das Training werden die Layers gemerged, was die Robustheit der Bewegungserkennung verbessert
- Die Winddaten werden physikalisch transformiert:
 - Umrechnung von m/s in Pixel pro 15 Minuten
 - Berücksichtigung der Erdkrümmung

Generieren der Trainingslabels (Movement Vectors)

- Zur Bestimmung der tatsächlichen Wolkenbewegung zwischen zwei Zeitpunkten wird OpenCV (<https://opencv.org>) verwendet
- Mithilfe von SIFT (https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html) Keypoint Descriptors werden visuelle Merkmale aus aufeinanderfolgenden Wolkenbildern extrahiert
- Die Descriptor-Distanz dient als Maß für die visuelle Ähnlichkeit zwischen zwei Wolkenpixeln
- Die Winddaten schränken den Suchraum möglicher Matches ein, sodass nur physikalisch plausible Bewegungen berücksichtigt werden
- Das Matching wird bewertet anhand von:
 - Distanz zur erwarteten Windbewegung
 - Descriptor-Distanz (visuelle Ähnlichkeit)
- Ergebnis: Movement Vectors (MVs) als Ground Truth für das Training

Modelltraining

- Die Daten werden in Grids organisiert und zu Gruppen mit ähnlichen topografischen Eigenschaften zusammengefasst
- Für jede Gruppe wird ein eigenes Modell trainiert
 - Input: transformierte Wind Daten
 - Target: Movement-Vektoren
- Das Training erfolgt mit PyTorch (<https://pytorch.org/>)
- Modellarchitektur:
 - Lineares neuronales Netzwerk mit 4 Layern
- Trainiert wird auf dem Supercomputer Leonardo
 - genutzt 1650 Core Stunden → ca. 250 Stunden mit 8 Cores
- Training über fünf Epochen (danach keine signifikante Verbesserung)
- Ergebnis: Modelle, die auf Basis meteorologischer Daten (Wind) Movement Vectors vorhersagen können

2. Cloud Prediction & Shadow Service

Der Service nutzt das trainierte Modell zur Zukunftsvorhersage und Schattenberechnung.

Vorhersagen bis zu sechs Stunden in die Zukunft sind möglich.

Inputdaten im Service

- Aktuelle Wolkendaten:
 - CTP (Cloud Top Pressure)
 - COT (Cloud Optical Thickness)
- Vorhergesagte zukünftige Wetterdaten (insbesondere Windfelder)
- Die tatsächliche Wolkendicke ist nicht direkt bekannt
- Sie wird anhand umliegender Wolkenstrukturen geschätzt

Preprocessing im Service

- Die Daten werden erneut vorverarbeitet (insbesondere Winddaten → Umrechnung in Pixelkoordinaten)
- Die Daten werden in Gruppen aufgeteilt – analog zur Struktur des Trainings

Motion-Vector-Prediction

- Das trainierte Modell berechnet die Movement Vectors für zukünftige Zeitpunkte
- Anschließend erfolgt eine Movement Aggregation
- Die Vektoren werden nicht einfach aufsummiert, sondern physikalisch konsistent kombiniert, um akkurate Multistep-Predictions zu erzeugen
- Der aggregierten Vektoren werden auf das Wolkenbild angewandt

Wolkenschatten-Berechnung

- Aus CTP wird die Wolkenhöhe (in m) abgeleitet
- Die Schattenberechnung erfolgt mittels Raytracing
 - COT/Lichtdurchlässigkeit wird berücksichtigt bei den Wolken
- Zur Performance-Optimierung wird Numba (<https://numba.pydata.org>) eingesetzt
- Der Output sind GeoTIFF-Dateien (Wolkenhöhe und Schatten)
- Die Bereitstellung erfolgt über MapServer (<https://mapserver.org>) an das Frontend

3. Terrain Shadows Service

Dieser Service berechnet Geländeschatten basierend auf globalen Terraindaten.

Daten

- Verwendet werden Terraindaten ASTER Global Digital Elevation Model (GDEM) Version 3
- Die globalen DEM-Daten wurden heruntergeladen und in Tiles aufgeteilt, danach in einem eigenem optimierten Datenbankformat gespeichert
- Die Tiles werden on demand per mmap geladen

Implementierung

- Der Service ist in Zig (<https://ziglang.org>) implementiert
- Die Berechnung erfolgt mittels Raytracing
- Es werden nur Tiles entlang der Sonnenrichtung geladen, die Anzahl ist aus Performancegründen begrenzt
- Die Ausführung erfolgt auf der GPU über Vulkan Compute Shader
- Das System ist für UMA-Architekturen optimiert, wodurch CPU und iGPU auf denselben Speicher zugreifen
- Die Berechnung der Schatten erfolgt auf dem Server und die Schatten werden dem Frontend übermittelt
- Die Simulation ist extrem genau (sehr viel genauer als ähnliche Systeme am Markt), berechnet Schatten pro Pixel und bezieht sehr viele Eigenschaften wie Erdkrümmung, oder Lichtbrechung in der Atmosphäre mit ein

Bei Fragen zum Modell oder Produktanfragen wenden Sie sich an office@luxactive.com