

Fotogrammetrie API

Ausgangslage:

Ein Benutzer kann ein Video hochladen und wird dann über eine Fortschrittsleiste darüber informiert, wie weit der Prozess fortgeschritten ist. Bei 100 Prozent sollte der Download-Button verfügbar werden, und das fertige Modell sollte im Browser angezeigt werden.

Fähigkeiten und Erfüllungsbeschreibung:

B1: Versionierung mit GitHub

Wir haben die gesamte Entwicklung über GitHub organisiert. Wir haben uns entschieden, direkt auf den Master-Branch zu pushen, da das Projekt klein ist und wir Merge-Anfragen vermeiden wollten.

[Dirk-Vantas/photogrametry-api: Node.js API for a photogrametry service based on nvidia nerf implementation \(github.com\)](https://github.com/Dirk-Vantas/photogrametry-api)

B3: Dynamische Multimedia-Inhalte

Unser Projekt basiert darauf, dass Benutzer ein Video hochladen, das in einzelne Frames aufgeteilt und dann von einem Photogrammetrie-Programm analysiert wird.

Dieses Programm (AliceVision Meshroom) generiert eine sparse Punktwolke durch Parallaxenanalyse von zugehörigen Punkten, die durch Feature-Matching erkannt wurden. Diese Punktwolke wird dann von einem Poisson-Mesher analysiert und in ein Mesh (Polygon-Oberfläche) umgewandelt.

Abschliessen wird das Ausgabemodell vom OBJ-Format in ein GLB-Dateiformat umgewandelt, das im Frontend angezeigt werden kann.

Dann wird dem Benutzer ein Download-Button zur Verfügung gestellt, mit dem er das Modell herunterladen kann, und ein Google Model Viewer-Element zeigt dynamisch das neue Modell zur Vorschau an.

B4: Responsive Webdesign

Die Webseite ist responsiv, und der Download-Button und der Model Viewer passen sich dynamisch der Bildschirmgröße an.

C1: Installationsanleitung und Validierung

Der Quellcode des Frontends wurde vom W3-Validator validiert, und in der Readme auf GitHub finden Sie eine Installationsanleitung sowie eine Projektbeschreibung.

C2: Bibliotheken und Dateiformateinschränkung

Der Upload der Videodatei (in jeglichem von FFmpeg unterstützten Format) wird im Frontend verarbeitet. Im Backend wird das Modell im OBJ-Format ausgegeben und dann in ein für das Frontend brauchbares GLTF-Dateiformat umgewandelt (GLB ist die binäre Version von GLTF). Der Download erfolgt ausschließlich im GLB-Format, und der Google Model Viewer zeigt dies ebenfalls als GLB-Datei an.

C3: Weiterverarbeitung von Multimedia-Inhalten

AliceVision Meshroom gibt nach der Zerlegung des Videos in einzelne Frames durch FFmpeg ein 3D-Modell aus, das als OBJ-Datei exportiert wird. Dies wird dann von einer Bibliothek in ein GLTF-Dateiformat umgewandelt, das im Frontend angezeigt wird, sobald es verfügbar ist.

Das Frontend startet beim Hochladen eine Sequenz, bei der der Download-Button gesperrt und ein Ladebalken angezeigt wird. Im Hintergrund wird der Fortschritt mithilfe von Statusanfragen an die API erfasst, und der aktuelle Status der Video-zu-Modell-Konvertierung wird angezeigt. Nachdem das Frontend die Meldung "Fortschritt 100 %" erhalten hat, wird der Download-Button aktiviert, und das neue Modell wird im Model Viewer angezeigt.

C4: Benutzerfreundlichkeit

Der Benutzer wird daran gehindert, Dateien hochzuladen, die keine Videodateien sind, und es wird sichergestellt, dass der Download-Button nur dann verfügbar ist, wenn das Video erfolgreich in ein Modell umgewandelt wurde.