



netidee

PROJEKTE

Images2Mesh Web

Endbericht | Call 15 | Projekt ID 5082

Lizenz: CC-BY

Inhalt

1	Einleitung	3
2	Projektbeschreibung.....	3
3	Verlauf der Arbeitspakete	4
3.1	Arbeitspaket 0 – Detailplanung und Formales	4
3.2	Arbeitspaket 1.1 – Basic Structure-From-Motion.....	4
3.3	Arbeitspaket 1.2 – Basic Reconstruction	5
3.4	Arbeitspaket 1.3 – Basic Coloring	6
3.5	Arbeitspaket 1.4 – Basic All-in-one Webserver	6
3.6	Arbeitspaket 1.5 – Hardware bestellen und einrichten	7
3.7	Arbeitspaket 2 – Family and Friends Testing	7
3.8	Arbeitspaket 3.1 – Optimized Structure-From-Motion	8
3.9	Arbeitspaket 3.2 – Points2Surf.....	8
3.10	Arbeitspaket 3.3 – Microservices	9
3.11	Arbeitspaket 3.4 – Server Security & Data protection.....	10
3.12	Arbeitspaket 4 – Public Testing.....	11
3.13	Arbeitspaket 5.1 – Speed-Up Points2Surf	11
3.14	Arbeitspaket 5.2 – Better Coloring.....	12
3.15	Arbeitspaket 5.3 – Inputs und Outputs im Browser rendern	12
3.16	Arbeitspaket 5.4 – Basic Editing.....	13
3.17	Arbeitspaket 6 – Dokumentation und Formales am Projektende.....	14
4	Umsetzung Förderauflagen	14
5	Liste Projektergebnisse	15
6	Verwertung der Projektergebnisse in der Praxis.....	15
7	Öffentlichkeitsarbeit / Vernetzung	15
8	Eigene Projektwebsite	16
9	Geplante Aktivitäten nach netidee-Projektende	16
10	Anregungen für Weiterentwicklungen durch Dritte	16

1 Einleitung

Wir sind eine Gruppe von Forschenden und Studierenden, die sich an der TU Wien mit Photogrammetrie beschäftigen. Mit unserem Webservice Images2Mesh Web machen wir dieses Forschungsfeld der Allgemeinheit kostenlos zugänglich. Interessierte können einfach Fotos hochladen und etwas später ein 3D-Modell davon runterladen, ganz ohne eigene starke Hardware oder aufwändiges Software-Setup. So können beispielsweise Verkehrsplaner eine Kreuzung digitalisieren und geplante Änderungen veranschaulichen. Eine andere Anwendung ist, Statuen einzuscannen und das digitale Modell als Basis für 3D-Druck zu verwenden. Die Microservice Struktur ermöglicht uns, einzelne Verarbeitungsschritte, wie Structure-from-Motion und Rekonstruktion durch neue Algorithmen zu ersetzen.

2 Projektbeschreibung

Dieses Projekt richtet sich an alle Menschen, die gerne etwas in der echten Welt einscannen und anderen Menschen als 3D-Modell zeigen wollen. Beispielsweise könnte ein Artist unterwegs eine interessante Skulptur finden und diese in ein Spiel einbauen oder als Miniatur ausdrucken. Raumplaner könnten z.B. eine Kreuzung einscannen, nach ihren Plänen verändern und diese damit besser kommunizieren. Museumsbesucher können Fotos von Objekten machen und diese in ein 3D-Modell umwandeln, es Freunden zeigen, oder sogar mit einem 3D-Drucker duplizieren. Künstler können ihre Werke als 3D-Modelle in ihr Online-Portfolio aufnehmen, nicht nur als Fotos.

Das Herzstück dieses Projekts ist eine Photogrammetrie-Pipeline, die Fotos in 3D-Modelle umwandelt. Die wichtigsten Schritte dabei sind Structure-from-Motion (SfM) und Oberflächenrekonstruktion. SfM sucht die gleichen Punkte in verschiedenen Fotos und bringt sie in einen 3D-Kontext. Die Rekonstruktion erzeugt aus diesen Punkten eine geschlossene Oberfläche. Die Ergebnisse können mit verschiedenen weiteren Verarbeitungsschritten weiter verbessert werden. Beispielsweise kann SfM mit genauen intrinsischen Kamera-Parametern verbessert werden. Mit Multi-View-Stereo kann die Punktwolke verdichtet werden und so die Rekonstruktion verbessern.

Diese Berechnungen können sehr aufwendig sein und brauchen daher starke Hardware. Mit unserem Photogrammetrie-Webservice übernehmen wir die Berechnungen, so dass Leute mit ganz normalen PCs und sogar Smartphones diese Anwendung nutzen können.

Für alle Schritte gibt es viele mögliche Algorithmen, neuerdings vor allem auf Basis von Deep-Learning. Die Microservice-Struktur unseres Webservices erlaubt es uns, einzelne Schritte relativ einfach auszutauschen.

Andere Forschungsgruppen können unser System einfach kopieren und adaptieren. Damit können die teilweise abstrakten Forschungsergebnisse besser der Öffentlichkeit präsentiert werden.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 0 – Detailplanung und Formales

Aufgaben

- Vertrag unterschrieben, Detailprojektplan (Arbeitsblatt Arbeitspakete) erstellt und abgenommen,
- detaillierte Liste Projektergebnisse mit Lizenz und Ort der öffentlichen Bereitstellung erstellt und abgenommen (Arbeitsblatt Projektergebnisse)
- Projekt-Website in Betrieb & erster Blogeintrag erstellt
- erste Förderrate beantragt; ggf. nach Feedback ergänzt
- Förderrate1 genehmigt

Status

Fertig

Details

Der Geplante Projektstart war im November 2020. Wegen kleinerer Probleme hat sich der Start leider verzögert. Der wichtigste Grund war, dass die TU Verwaltung ein extra Projektkonto gefordert hat, weshalb wir die 1. Förderrate nicht früher beantragen konnten. Insgesamt hat dies zu einer Verzögerung von fast 2 Monaten geführt. Kurz vor Weihnachten und Neujahr anzufangen hätte sich auch nicht mehr ausgezahlt, also haben wir die Implementierung erst mit Anfang Jänner 2021 begonnen. Diese Verzögerung zieht sich durch alle Arbeitspakete und erfordert eine Verlängerung des Projektes.

3.2 Arbeitspaket 1.1 – Basic Structure-From-Motion

Aufgaben

- Programmieren eines Prozesses, der
- n Fotos (als PNG, JPEG, NPY etc.) bekommt
- Structure-From-Motion darauf anwendet und
- eine Punktwolke erzeugt.
- Die genutzte Variante von SFM soll schnell zu implementieren sein, vorhandene Bibliotheken checken, wie z.B. OpenCV. Lizenz checken!

Status

Fertig

Details

Nach kurzer Evaluierung haben wir uns für die Verwendung von COLMAP (<https://colmap.github.io/>, BSD Lizenz) entschieden. COLMAP kombiniert Structure-From-Motion, Multi-View-Stereo, Rekonstruktion und Coloring, wodurch andere Arbeitspakete teilweise überflüssig werden. Dies betrifft AP 1.2 “Basic Surface Reconstruction” und AP 1.3 “Nearest Neighbor Colors”. Im Moment ist noch nicht klar, ob die Rekonstruktion von COLMAP gut genug ist und AP 1.2 “Basic Surface Reconstruction” komplett überflüssig macht. Es ist auch noch nicht klar, ob das Coloring von COLMAP AP 1.3 “Nearest Neighbor Colors” überflüssig macht. Die 1. Testphase wird diese Fragen beantworten. Wahrscheinlich sind die Farben akzeptabel für erste Tests, aber nicht für die finale Anwendung, weshalb AP 5.2 Better Coloring im Plan bleibt.

COLMAP läuft in Step 1 der Microservice-Architektur. Die Inputs sind beliebige Bildformate, der Output ist eine Punktwolke im PLY Format.

3.3 Arbeitspaket 1.2 – Basic Reconstruction

Aufgaben

- Programmieren eines Prozesses, der
- eine Punktwolke (als NPY, LAS etc.) bekommt
- Screened Poisson Surface Reconstruction (PyMeshlab) darauf anwendet,
- ein Mesh erzeugt. Mesh als STL oder PLY speichern.

Status

Fertig

Details

Screened Poisson Surface Reconstruction ist praktisch der Standard für Oberflächenrekonstruktion aus Punktwolken. Die einfachste Art es zu verwenden, ist mit PyMeshlab (<https://github.com/cnr-isti-vclab/PyMeshLab>, GNU GPL 3).

Diese Implementierung von Screened Poisson Surface Reconstruction (SPSR) übernimmt Farben als Vertex-Colors in das rekonstruierte Mesh. Vor allem in Bereichen mit wenig Details führt dies zu sehr verwaschenen Farben. Diese Coloring-Methode ist definitiv nicht gut genug für die finale Anwendung. Daher ist AP 1.3 “Nearest Neighbor Colors” überflüssig, nicht jedoch AP 5.2 “Better Coloring”. Da COLMAP (siehe AP 1.1 “Basic Structure-From-Motion”) auch eine Rekonstruktion beinhaltet, ist ein kurzer Vergleich nötig. COLMAP verwendet Poisson Surface Reconstruction (Vorgänger von SPSR). Die Variante von COLMAP ist etwas ungenauer, dafür aber dank CUDA-Optimierung schneller und durch andere Boundary-Conditions besser für lückenhafte Punktwolken geeignet. Für die finale Anwendung wird die Rekonstruktion durch unser Points2Surf ersetzt, d.h. AP 3.2 “Points2Surf” und AP 5.1 “Speed-up Points2Surf” bleiben im Plan. SPSR ist in Schritt 2 implementiert. Inputs sind die Punktwolken von COLMAP und Outputs sind Meshes mit ca. 100MB, die mehrere Millionen Vertices enthalten.

3.4 Arbeitspaket 1.3 – Basic Coloring

Aufgaben

- Programmiere einen Prozess, der
- eine Punktwolke (als NPY, LAS etc.) bekommt
- ein Mesh (als STL oder PLY) bekommt
- für jeden Vertex im Mesh den nächsten Punkt in der Punktwolke sucht
- die Farbe vom Punkt zum Vertex überträgt und
- das gefärbte Mesh speichert.
- Nearest-neighbor search mit spatial acceleration structure, z.B. scipy.ckdtdtree.

Status

100%

Details

Wie in AP 1.1 “Basic Structure-From-Motion” und AP 1.2 “Basic Reconstruction” beschrieben, sind einfache Coloring Methoden bereits in SFM und SPSR inkludiert. Wir müssen es also nicht selbst implementieren.

Die Qualität reicht allerdings nur für die 1. Testphase. Für die finale Version bleibt AP 5.2 “Better Coloring” erhalten.

3.5 Arbeitspaket 1.4 – Basic All-in-one Webserver

Aufgaben

- Programmiere einen Webservice
- mit minimalem Web-Interface
- Upload für Fotos (Datenmenge limitieren, um DOS zu verhindern)
- Lasse die Fotos durch die Pipeline (1.1-1.3) laufen
- Gefärbtes Mesh zum Download anbieten
- Webservice Set-up dokumentieren, damit alle im Team testen können
- Überlastung verhindern durch z.B. eine einfache Warteliste
- Checke Django, Heroku, (Nvidia) Docker

Status

100%

Details

Einige Teile dieses AP wurden übersprungen, weil sie mit Microservices (siehe AP 3.3) auch sehr schnell implementiert werden konnten, die finale Lösung also schon schnell genug erreichbar war. Das Frontend ist größtenteils im Rahmen von AP 5.3 und AP 5.4 (Rendering und Editing) entstanden.

3.6 Arbeitspaket 1.5 – Hardware bestellen und einrichten

Aufgaben

- Zusammen mit den Technikern des CG Instituts die Hardware besprechen. Ca. Ryzen 7, 2 x RTX 3070, 64 GB RAM.
- Domain für den Webservice absprechen und von den Technikern einrichten lassen. Es wird vielleicht eine Sub-Domain wie images2mesh.cg.tuwien.ac.at
- Bei Lieferschwierigkeiten evtl. einen alten Rechner temporär verwenden.
- SSH-Zugang für Projektmitarbeiter

Status

100%

Details

Die Kosten waren zuerst deutlich zu niedrig eingeschätzt. Daher haben wir, mit Netidee abgesprochen, etwas Geld umgeplant. Jetzt sind insgesamt 6000€ für Hardware eingeplant.

Am Anfang des Projekts waren die Preise für Grafikkarten noch extrem, zum Teil mehr als das doppelte des Listenpreises. Zum Glück haben wir noch keine GPU gebraucht. Als diese dann im August nötig wurde, hatten sich die Preise wieder fast normalisiert. Geld für eine 2. RTX 3090 ist vorhanden.

3.7 Arbeitspaket 2 – Family and Friends Testing

Aufgaben

- Wenn der Server im Testbetrieb mit den einfachen Algorithmen läuft, erste Tests mit Bekannten und Verwandten machen.
- Klären, ob die Ergebnisse die Erwartungen erfüllen und warum. Tests mit verschiedenen Mengen von Fotos, verschiedene Qualität, Lichtverhältnisse, Objekte etc. Richtlinien für nötige Anzahl Fotos suchen. Gefundene Probleme entweder gleich beheben oder in den Workplan eintragen.
- Kontakt zu Künstlern, Museen etc. herstellen und testen lassen.
- Zwischenbericht schreiben und 2. Rate beantragen

Status

75% fertig

Details

Die 1. Testphase ist abgeschlossen. Der Zwischenbericht und die 2. Rate sind erledigt.

Bisher haben hauptsächlich Freunde, KollegInnen und Familienmitglieder getestet. Kontakt zu Künstlern und Museen ist bisher eher eingeschränkt.

Die wichtigste Erkenntnis ist, dass die Vertex-Colors nur bei sehr detaillierten Meshes akzeptable Farben liefern. Wenn wir Farben durch Texturing einbauen, können wir die Meshes verkleinern. Damit würden wir Datenmengen reduzieren und Rendering beschleunigen, was gerade für Smartphones wichtig ist.

3.8 Arbeitspaket 3.1 – Optimized Structure-From-Motion

Aufgaben

- Genauere Recherche, um die beste Variante von SFM für die in den Tests gefundenen Bedingungen auszuwählen.
- Implementierungsaufwand abwägen.
- Implementieren, evtl. in C++, sonst wahrscheinlich in Python.

Status

90% fertig

Details

Es gibt einige neue Methoden, die Deep Learning verwenden. Diese neuen Methoden wirken vielversprechend, ihre Praxistauglichkeit ist aber unklar. Mehr Recherche ist nötig. COLMAP braucht ca. 1 Stunde für 100 Bilder, was zu lang ist. Wir brauchen eine andere Lösung.

Umfangreiche Tests und Vergleiche haben gezeigt, dass NeRF-basierte Methoden noch keine brauchbaren Ergebnisse liefern. Verwandte, Geometrie-basierte Methoden wie NeuS und Neuralangelo haben viel Potential für die Zukunft, sind allerdings zum Teil viel zu langsam und/oder haben inkompatible Lizenzen.

Die aktuelle Version von AliceVision / Meshroom erzeugt sehr gute Ergebnisse in akzeptabler Zeit. Die Umstellung von COLMAP zu Meshroom macht das Setup einfacher. Meshroom ist modular, wodurch wir in Zukunft einfacher einzelne Schritte ersetzen können. Für Enduser wird der Prozess verständlicher durch Auswahl einer Bounding Box für die Rekonstruktion. Diese Umstellung läuft und ist fast abgeschlossen.

3.9 Arbeitspaket 3.2 – Points2Surf

Aufgaben

- Oberflächenrekonstruktion durch verbesserte Version von Points2Surf ersetzen
- Einfache installation als Pip/Conda package

Status

75% fertig

Details

Points2Surf wird bald durch unsere aktuelle Forschung ersetzt. Die Optimierungen werden zu AP 5.1. Speed-Up P2S verschoben.

Die verbesserte Version unserer Rekonstruktion hat leider unerwartet lange gedauert. Sie ist nun endlich im Review Prozess eines wissenschaftlichen Journals. Der Code ist schon gesäubert und bereit. Sobald positive Reviews kommen, werden wir alles auf öffentlich stellen und die Rekonstruktion einbauen.

3.10 Arbeitspaket 3.3 – Microservices

Aufgaben

- Server umbauen, damit die einzelnen Pipeline-Schritte getrennt laufen können. Z.B. je ein Microservice (Docker-Container) pro Schritt.
- Mehrere Prozesse gleichzeitig erlauben, aber nur 1 Rekonstruktion pro Grafikkarte.
- Upload von Punktwolken erlauben (SFM überspringen).
- Warteschlange mit Schätzung der Wartezeit.
- Fortschritt und Position in Warteschlange auf Website anzeigen.

Status

90% fertig

Details

Die Schritte der Pipeline können aktuell in unabhängigen Docker-Containern voneinander ausgeführt auf einen Rechner ausgeführt werden. Bei Bedarf können auch mehr Worker-Container gespawnt werden, um parallel zu Arbeiten und die Hardware des Servers besser zu nutzen.

Für schnelleres Testen haben wir Continuous Integration (CI) via Github Actions implementiert, was das Repo auf dem Live-System pullt und deployed. Leider, haben mehrere Male winzige Änderungen am Code, wie z.B. Änderungen in der Nginx Config, das Live-System komplett lahmgelegt. Deshalb haben wir CI für den Develop-Branch Priorität gegeben. Dadurch können wir den Develop-Branch in wenigen Minuten testen, bevor wir ein Live-Deploy machen.

Allerdings hat dies Probleme mit HTTPS Zertifikaten und den zwei Web-Servern verursacht. Dies wurde erzeugt aus einem Mix von einzelner physischen Maschine mit mehreren GPUs, einzelner Web-Server mit zwei URLs und dazugehörigen HTTPS Zertifikaten. Generell haben die Zertifikate lange Probleme beim Setup der Entwicklungsumgebung verursacht.

Die Lösung dieses Problems ist eine Vereinfachung mit der Einschränkung, dass die Development-URL und das RQ-Dashboard (<https://netidee.cg.tuwien.ac.at/>) kein gültiges Zertifikat haben. D.h. nur das offizielle Live-System (<https://pix2model.cg.tuwien.ac.at/>) hat ein gültiges HTTPS Zertifikat.

Folgende Punkte sind noch offen, schränken aber nicht die Funktionalität ein:

- Warteschlange und geschätzte Wartezeit anzeigen
- Upload von Punktwolken erlauben
- Skalierung auf viele GPUs und / oder Server

3.11 Arbeitspaket 3.4 – Server Security & Data protection

Aufgaben

- DOS verhindern, indem häufige IP-Adressen blockiert werden.
- Admin-Zugang, um einzelne Prozesse beenden zu können und Parameter (z.B. Größen-Limits und Grid-Resolution) für einzelne Nutzer ändern zu können.
- Login mit temporären Login-Daten. Login-Daten und alle anderen Daten auf Knopfdruck sofort von Server löschen, sonst nach 1 Tag.
- Auf Wunsch Email senden, wenn das Ergebnis bereit zum Download ist.
- Server-Telemetrie und Email, wenn der Server nicht mehr erreichbar ist (status-checker Prozess auf anderen CG Server?).

Status

0% fertig

Details

Logins für normale Rekonstruktionen werden nicht nötig sein. Accounts mit besonderen Rechten wären ein guter Schritt in Richtung Monetarisierung.

Unser Backend-Entwickler ist mit Arbeitspaket 3.3. (Microservices und CI) ausgelastet gewesen. Ein weiterer Entwickler ist nicht in Reichweite gewesen. Die Aufgaben in diesem Arbeitspaket sind nicht kritisch für die normale Nutzung des Systems, weshalb wir ihnen niedrige Priorität gegeben haben.

Die Benachrichtigung per Email wäre wichtig für Usability und braucht eher wenig Aufwand, weshalb wir dies gern noch implementieren würden.

3.12 Arbeitspaket 4 – Public Testing

Aufgaben

- Fokus auf Usability-Testing. Lasst eure Oma euren Opa einscannen und rekonstruieren
- Testen, ob alle Algorithmen funktionieren, wie gewünscht.
- Ist das Coloring ausreichend gut?
- Testen, ob das Mesh gut genug für 3D-Druck ist.
- Performance Tests

Status

75% fertig

Details

Die beiden Testphasen (Arbeitspaket 2 und 4) hatten einen fließenden Übergang und lassen sich daher nicht wirklich trennen. Wie bei Arbeitspaket 2 beschrieben, ist die wichtigste Erkenntnis, dass Texturing nötig ist. Externe Tests müssen noch ausgebaut sein.

3.13 Arbeitspaket 5.1 – Speed-Up Points2Surf

Aufgaben

- Points2Surf ist relativ langsam.
- Mit iterativem Verfeinern wie MISE in Occupancy Networks müsste es schneller und genauer werden.
- Low-level Optimierungen wie Custom CUDA Kernels.
- Punktwolke vor Verarbeitung auf $-0.5 \dots +0.5$ skalieren
- das erzeugte Mesh auf die ursprüngliche Größe der Punktwolke skalieren
- sehr große Punktwolken vorher down-samplen auf ca. 50k Punkte
- Neu trainieren mit ABC+Famous+Thing10k datasets?

Status

90% fertig

Details

Dieses AP hat Überschneidungen mit unserer aktuellen Forschung. Wir kombinieren dieses Netidee-Projekt und unser aktuelles Forschungsprojekt, um Synergien zu nutzen. Manche Engineering-lastige Aufgaben sind im Forschungsprojekt nicht vorgesehen, haben aber einen großen Wert für die Endnutzer. Solche Aufgaben sind in diesem AP eingeplant.

Bei der Umsetzung von AP 3.2 (Points2Surf2 / bessere Rekonstruktion) wurde auch die Geschwindigkeit deutlich verbessert, von ca. 10 Minuten auf 1 Minute pro Objekt. Weitere Low-Level Verbesserungen sind möglich, aber aufwändig und nicht kritisch für die Nutzbarkeit des Systems.

3.14 Arbeitspaket 5.2 – Better Coloring

Aufgaben

- Nearest Neighbor Vertex Colors (NNVC) ist sehr einfach und wird bei kleineren Grid-Resolutions schlechte Ergebnisse liefern. Mit Philipp, Stefan und/oder Michi absprechen, wie man das verbessern kann. Mögliche Lösungen:
- Sub-division des Meshes und dann NNVC
- automatisches UV-Mapping und Erzeugung von Texturen aus den Punktwolken.
- Texturen aus den Fotos?

Status

90% fertig

Details

Einfache NNVC reichen nur für die 1. Testphase (siehe AP 1.3 “Basic Coloring”). Die Qualität ist vor allem in niedrig aufgelösten Bereichen (z.B. grobe Hierarchieebenen von SPSR) schlecht. Ein Subdivision-Ansatz könnte dieses Problem teilweise lösen. Da wir eine Nutzung auf schwacher Hardware (z.B. Smartphones) anstreben, wäre eine Texturing-basierte Lösung deutlich besser. Das Mapping von Bildern zu Texturen ist allerdings relativ kompliziert. Eventuell können wir die nötigen Informationen aus COLMAP extrahieren, müssten aber wahrscheinlich den Code ändern. Alternativ können wir vielleicht auch spezielle Forschungsergebnisse verwenden, die Praxistauglichkeit müssen wir aber erst herausfinden.

Die Testphasen haben das Problem mit Vertex-Colors bestätigt. Deshalb haben wir dem Texturing hohe Priorität gegeben. Allerdings ist dieses Arbeitspaket unerwartet langwierig gewesen. Wir haben lange versucht, das Texturing von OpenMVG in unserer COLMAP Pipeline nach der Rekonstruktion einzubauen. Leider hat das Setup mit extra Docker-Image nie funktioniert.

Wenn wir den Umbau von COLMAP zu AliceVision / Meshroom fertig haben, ist Texturing automatisch dabei. Das Texturing von Meshroom ist zwar relativ langsam, aber robust und hat gute Qualität. Es hat viele Parameter, wodurch wir es etwas beschleunigen können.

3.15 Arbeitspaket 5.3 – Inputs und Outputs im Browser rendern

Aufgaben

- Hochgeladene Fotos anzeigen, Punktwolken und Meshes (mit WebGL) anzeigen.

- Extra Viewer, mit dem man bereits heruntergeladene Meshes anzeigen kann.

Status

95%

Details

Der Viewer ist fertig. Punktwolken und Meshes können angezeigt werden. Kleiner Optimierungen und Verbesserungen sind implementiert. Siehe AP 5.4 “Basic Editing” für mehr Details zu den verwendeten Technologien.

Für AP 5.2 “Better Coloring” ist schon Texturing implementiert. Es muss dann nur noch aktiviert werden.

3.16 Arbeitspaket 5.4 – Basic Editing

Aufgaben

Basic editing von Punktwolke und/oder Mesh:

- cropping, verschieben, skalieren, rotieren, zurücksetzen
- in Web-Interface using WebGL
- Bounding-Box Längen anzeigen

Status

95%

Details

Das Frontend ist eine React-Anwendung und setzt sich aus 5 Bereichen zusammen:

- Beschreibung des Projektes - Implementiert
- Upload Bilder, um den Meshing Prozess zu starten - Implentiert
- Upload von Punktwolken, um den Meshing Prozess zu starten - Implentiert
- Download der bereitgestellten Meshes über ID - Implentiert
- Viewer / Editor - Implementiert

Der Viewer / Edit Bereich integriert Three.js. Der Nutzer kann die ply Modelle, die nach dem Upload der Bilder / der Punktwolken entstanden sind, hochladen und editieren. Mögliche Funktionalitäten sind:

- Croppen (um bspw. den Untergund wegzuscheiden, auf welchem das zu rekonstruierende Modell fotografiert wurde) - Implementiert
- Basis Transformationen: Translation, Skalierung, Rotation – Implementiert
- Bounding Box Längen anzeigen –Implementiert

Für die Umstellung zur AliceVision / Meshroom Pipeline müssen wir die vorhandenen Features etwas adaptieren. Der Punktwolken Editor wählt dann eine Bounding Box für die Rekonstruktion aus, der Editor fürs 3D-Modell ist dann nur noch ein Viewer.

3.17 Arbeitspaket 6 – Dokumentation und Formales am Projektende

Aufgaben

- APN: Dokumentation und Formales am Projektende
- Die geplanten Projektergebnisse (siehe Arbeitsblatt "Projektergebnisse") sind erstellt/funktionsfähig und ausreichend dokumentiert;
- Projekt-Website wurde ein letztes Mal aktualisiert: Projektergebnisse sind unter Angabe der open source bzw. creative commons Lizenz der Öffentlichkeit zur Verfügung gestellt,
- Projektendbericht und Endabrechnung sind abgenommen; abschließender Projektblogeintrag; letzte Förderrate beantragt

Status

100%

Details

Erledigt:

- Repository ist public: <https://github.com/ErlorPhilipp/Pix2Model>
- Live-System läuft: <https://pix2model.cg.tuwien.ac.at/>
- Beschreibung für Enduser im Live-System/Info
- Dokumentation für Server Setup im Repo
- Dokumentation für Development Setup im Repo
- Veröffentlichungsfähiger Einseiter

4 Umsetzung Förderauflagen

Keine Auflagen

5 Liste Projektergebnisse

1	Projektzwischenbericht und Projektendbericht	CC-BY-3.0	https://netidee/images2mesh-web
2	Entwickler_innen-DOKUMENTATION	MIT License	https://github.com/ErlerPhilipp/Pix2Model/blob/main/README.md
3	Anwender_innen-DOKUMENTATION	MIT License	Info-Tab im Live-System https://pix2model.cg.tuwien.ac.at/
4	Veröffentlichungsfähiger Einseiter	CC-BY-3.0	https://netidee/images2mesh-web
5	Dokumentation Externkommunikation	CC-BY-3.0	Im Endbericht auf https://netidee/images2mesh-web
6	Source Code Server, Photogrammetrie	MIT License	https://www.netidee.at/images2mesh-web https://github.com/ErlerPhilipp/Pix2Model

6 Verwertung der Projektergebnisse in der Praxis

Aus Datenschutz- und Speicherplatzgründen löschen wir hochgeladene Fotos und die erzeugten 3D-Modelle nach 7 Tagen. Wir sammeln keine Statistiken und aktuelle Daten können nur mit gesicherter Verbindung (SSH) direkt auf dem Server gefunden werden. Daher sind Aussagen über die Nutzung schwierig.

Auf dem Server tauchen hin und wieder Daten von externen Nutzern auf. Das Github Repository zieht langsam Aufmerksamkeit auf sich. Kollegen an der TU Wien und beim VRVis nutzen unser System gelegentlich für schnelle Tests. Der Family-and-Friends Test hat gezeigt, dass auch normale Leute ohne Vorwissen das System nutzen können.

7 Öffentlichkeitsarbeit / Vernetzung

Bisher haben wir noch nicht viel Öffentlichkeitsarbeit gemacht. Bei Rechenzeiten von ca. 1 Stunde pro Szene wäre unser Server schnell überlastet gewesen. Mit den abschließenden Änderungen (AliceVision / Meshroom) wird dieses Problem entschärft.

Wir haben Kontakte gesammelt, u.a. zu einer Gruppe an der Universität für angewandte Kunst Wien und Architekten an der TU Wien. Von diesen Partnern erwarten wir einige Tests und weitere Verbreitung, was allerdings erst nach Projektende passieren dürfte.

Wir werden unseren Webservice auch im Rahmen wissenschaftlicher Konferenzen und Publikationen weiter bewerben.

8 Eigene Projektwebsite

Live-System: <https://pix2model.cg.tuwien.ac.at/>

Repository: <https://github.com/ErlerPhilipp/Pix2Model>

Allgemeine Informationen: <https://www.cg.tuwien.ac.at/research/projects/Images2Mesh-Web>

9 Geplante Aktivitäten nach netidee-Projektende

Im Rahmen von Forschungsprojekten bleiben wir auf dem aktuellen Stand im Bereich Photogrammetrie und können immer wieder neue Algorithmen in das System einbauen. Neben Abschlussarbeiten kann unser System auch bei Praktika weiterentwickelt werden. Eventuell beantragen wir ein Folgeprojekt bei Netidee, um den Studierenden weitere Anreize bieten zu können. Der Fokus liegt vor allem auf einem UI-Update und neuen Algorithmen für die Photogrammetrie-Pipeline.

Bei wissenschaftlichen Konferenzen können wir unseren Photogrammetrie Webservice bewerben. Möglicherweise können wir auch eine wissenschaftliche Publikation dazu schreiben.

Das geplante System ist für Tests von Privatpersonen und Organisationen gedacht. Wenn genug Nachfrage für größere Projekte besteht, wäre auch eine Ausgründung und Weiterentwicklung denkbar. Die Gründungsunterstützung der TU Wien, i2c, wäre dafür ein Ansprechpartner.

10 Anregungen für Weiterentwicklungen durch Dritte

Es gibt einige neue Algorithmen, die in die Photogrammetrie-Pipeline eingebaut werden könnten. Am aussichtsreichsten dürften NeuS-basierte Neuronale Netze sein, wie z.B. Neuralangelo.

Wenn eine kommerzielle Nutzung gewünscht wird, sollte der Server um Accounts mit speziellen Rechten erweitert werden. Diese Rechte könnten folgendes beinhalten: Vorreihung in Warteschlange, mehr Input-Fotos, aufwändigere Algorithmen und Parameter für bessere Qualität.