



# Images2Mesh Web

Zwischenbericht | Call 15 | Projekt ID 5082

Lizenz: CC-BY

# Inhalt

1	Einleitung.....	3
2	Status der Arbeitspakete.....	3
2.1	Arbeitspaket 0 – Detailplanung und Formales.....	3
2.2	Arbeitspaket 1.1 – Basic Structure-From-Motion.....	3
2.3	Arbeitspaket 1.2 – Basic Reconstruction.....	4
2.4	Arbeitspaket 1.3 – Basic Coloring.....	5
2.5	Arbeitspaket 1.4 – Basic All-in-one Webserver.....	5
2.6	Arbeitspaket 1.5 – Hardware bestellen und einrichten.....	6
2.7	Arbeitspaket 2 – Family and Friends Testing.....	6
2.8	Arbeitspaket 3.1 – Optimized Structure-From-Motion.....	7
2.9	Arbeitspaket 3.2 – Points2Surf.....	7
2.10	Arbeitspaket 3.3 – Microservices.....	7
2.11	Arbeitspaket 3.4 – Server Security & Data protection.....	8
2.12	Arbeitspaket 4 – Public Testing.....	8
2.13	Arbeitspaket 5.1 – Speed-Up Points2Surf.....	8
2.14	Arbeitspaket 5.2 – Better Coloring.....	9
2.15	Arbeitspaket 5.3 – Inputs und Outputs im Browser rendern.....	9
2.16	Arbeitspaket 5.4 – Frontend Editor.....	10
2.17	Arbeitspaket 6 – Dokumentation und Formales am Projektende.....	10
3	Umsetzung Förderauflagen.....	11
4	Zusammenfassung Planaktualisierung.....	11
5	Öffentlichkeitsarbeit/ Vernetzung.....	12
6	Eigene Projektwebsite.....	12

# 1 Einleitung

Mit Images2Mesh Web möchten wir eine ganze Photogrammetrie-Pipeline der Allgemeinheit zugänglich machen. Für möglichst niedrige Einstiegshürden sorgt unser Webservice, d.h. die EndnutzerInnen benötigen weder starke Hardware noch müssen sie Software installieren. Zusätzlich soll es die Microservice-Struktur ermöglichen, dass Forschungsgruppen, die normalerweise nicht die Ressourcen für solche Entwicklungsarbeit haben, ihre Ergebnisse einfacher zur Verfügung stellen können.

Die Basisimplementierung ist fertig. D.h. alle Verarbeitungsschritte sind zumindest mit einfachen Algorithmen implementiert. Die erste Testphase läuft und ermöglicht uns, den weiteren Plan zu verfeinern.

## 2 Status der Arbeitspakete

### 2.1 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.

### 2.2 Arbeitspaket 1.1 – Basic Structure-From-Motion

#### Aufgaben

- Programmiere einen Prozess, 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.

## **2.3 Arbeitspaket 1.2 – Basic Reconstruction**

### **Aufgaben**

- Programmiere einen Prozess, 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**

90% fertig. Es fehlt nur noch etwas Polishing und Error Handling.

### **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 with ca. 100MB, die mehrere Millionen Vertices enthalten.

## 2.4 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.ckdtree`.

### Status

Nicht nötig

### 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.

## 2.5 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

Fertig

### 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.

## **2.6 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**

Einsatzbereit für Testphase, Geld für Erweiterungen ist vorhanden

### **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.

## **2.7 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**

10% fertig

### **Details**

Die 1. Testphase läuft demnächst. Den Zwischenbericht können Sie hier gerade bewundern.

## 2.8 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

5% 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.

## 2.9 Arbeitspaket 3.2 – Points2Surf

### Aufgaben

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

### Status

0% fertig

### Details

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

## 2.10 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

75% 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. Die Erreichbarkeit des Backends wird nun über einen Nginx Reverse Proxy gewährleistet, wodurch Frontend und Backend über die selbe Domain und Port erreicht

werden können. Der nächste Schritt wäre das ganze auf dem Server zu deployen und zu testen.

## 2.11 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.

## 2.12 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

0% fertig

### Details

Erstmal die 1. Testphase erledigen

## 2.13 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...+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+Thingy10k datasets?

### **Status**

10% 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.

## **2.14 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**

5% 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.

## **2.15 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**

80%

## Details

Der Viewer ist fast fertig. Punktwolken und Meshes können angezeigt werden. Kleiner Optimierungen und Verbesserungen sind noch nötig. Beispielsweise fehlt noch shading. Siehe AP 5.4 "Basic Editing" für mehr Details zu den verwendeten Technologien. Wenn wir uns bei AP 5.2 "Better Coloring" für eine Texturing-Lösung entscheiden, muss das auch der Viewer anzeigen können. Zudem muss der Viewer noch hinsichtlich der Benutzerfreundlichkeit getestet werden.

## 2.16 Arbeitspaket 5.4 – Frontend Editor

### Aufgaben

Basic editing von Punktwolke und/oder Mesh:

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

### Status

80%

### Details

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

1. Beschreibung des Projektes - Implementiert
2. Upload Bilder, um den Meshing Prozess zu starten - Implentiert
3. Upload von Punktwolken, um den Meshing Prozess zu starten - Mock-up
4. Download der bereitgestellten Meshes über ID - Mock-up
5. 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:

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

## 2.17 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

5%

### Details

- Dokumentation für Server aufsetzen teilweise vorhanden

## 3 Umsetzung Förderauflagen

Keine Auflagen

## 4 Zusammenfassung Planaktualisierung

### Gehalt:

TU Wien Bedingungen: Honorarnoten sind nur möglich, wenn kein Beschäftigungsverhältnis besteht. Das heißt für die TU-Angestellten Philipp Erler, Stefan Ohrhallinger und Michael Wimmer, dass Einkommen aus diesem Projekt als Sonderzahlungen behandelt werden müssen. Daher fällt hier der Arbeitgeberanteil an. Wir erhöhen also den Stundensatz für die Abrechnung um 30% (Netidee Richtlinie für Arbeitgeberanteil). Die geplanten Arbeitsstunden werden entsprechend reduziert. Diese Stundenreduktion dürfte kein Problem für den Erfolg des Projekts darstellen.

### Aufgaben:

- AP 1.5. Hardware: Geld verschoben innerhalb des AP 1.5. Bereits akzeptiert, siehe Email am 2021-02-08.
- AP 3.2 und AP 5.1 Points2Surf: Points2Surf wird durch unser neues Forschungsergebnis ersetzt. Einige Aufgaben fallen weg, andere kommen hinzu.
- AP 1.3 und AP 5.2 Coloring: “Basic Coloring” fällt weg und “Better Coloring” wird ausgebaut.
- AP 3.3 Microservices:
  - In der Zukunft möchten wir die Möglichkeit haben, das ganze Server-übergreifend zu betreiben (also z.B. Worker auf anderen Servern zu hosten). Dazu müssten wir aktuell noch die hochgeladenen Bilder auf diese Worker-Server übertragen werden. Es gibt sicher eine bessere Lösung dafür.
  - Continuous Integration, damit wir neue Commits automatisch auf dem Server deployen können.
  - Aufteilen von Tasks auf verfügbare GPUs und CPUs
- AP 3.4 Server Security & Data Protection:
  - Priority-System und Admin-Kontrolle für Tasks
- Passwort für Zugriff auf Bilder, Punktwolken und Meshes
  - Server Telemetrie, Notification on Crash

### **Arbeitszeit und Projektdauer:**

Die Verzögerung am Anfang zieht sich durch das ganze Projekt. Alle Beteiligten können nur neben ihrer Hauptbeschäftigung am Projekt arbeiten, was die Planung extrem schwierig macht. Die bisherigen Erfahrungen zeigen, dass eine Verlängerung nötig ist.

Bisher gab es wenig Beratungsbedarf, daher bleibt mehr Arbeitszeit für Implementierung. Die Zwischenergebnisse des Projekts (minimal working Prototype) sind voraussichtlich bis Ende 2021 bereit und für die Allgemeinheit nutzbar, allerdings noch nicht mit der vollen Qualität und Geschwindigkeit.

## 5 Öffentlichkeitsarbeit/ Vernetzung

Noch keine. Während und nach der 1. Testphase nehmen wir Kontakt auf mit Testern und möglichen Interessenten. Zu den Interessenten zählen wir u.a. Museen, MA 41 (Stadtvermessung), StadtplanerInnen, ArchitektInnen, KünstlerInnen und MaklerInnen. Zu diesen Gruppen haben wir private und berufliche Kontakte.

Wir nutzen die Synergie mit unserer aktuellen Forschung (Verbesserungen für Points2Surf), indem wir z.B. bei wissenschaftlichen Konferenzen diesen Webservice für Demos verwenden.

Die TU Wien, die Fakultät Informatik und das CG-Institut hat Kanäle auf Social Media, die wir eventuell nutzen können, um dieses Projekt zu bewerben.

Zusätzlich gibt es in der wissenschaftlichen Gemeinschaft Kanäle wie z.B. 2d3d.ai, die für eine Verbreitung geeignet wären.

## 6 Eigene Projektwebsite

Unser Webservice wird auf <https://netidee.cg.tuwien.ac.at/> erreichbar sein. Eventuell ändern wir die URL noch, wenn wir einen besseren Namen haben.

Der Source Code ist bald auf <https://github.com/ErlorPhilipp/Images2Mesh> verfügbar. Im Moment ist das Repository noch auf private gestellt, kann also nur von Teammitgliedern eingesehen werden. Spätestens zum Projektende stellen wir es dann auf public.