

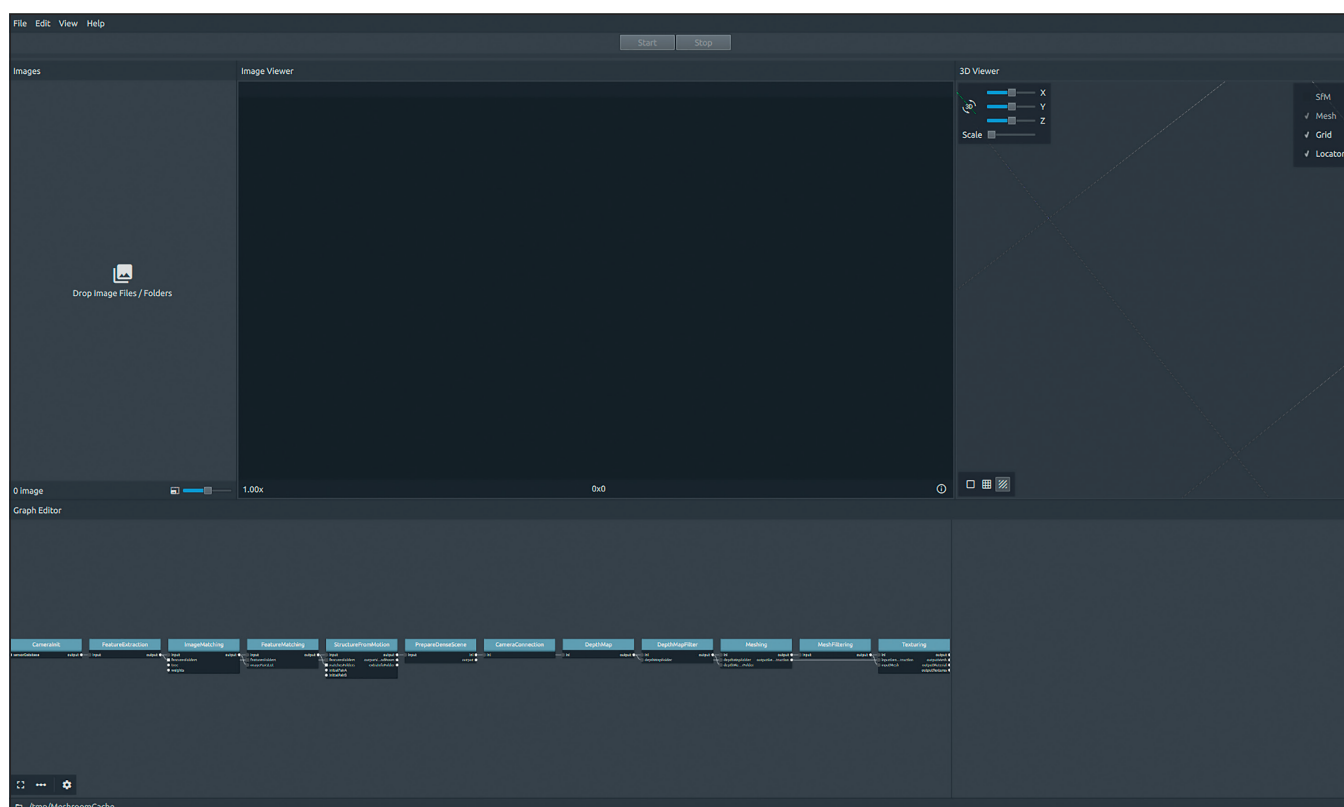


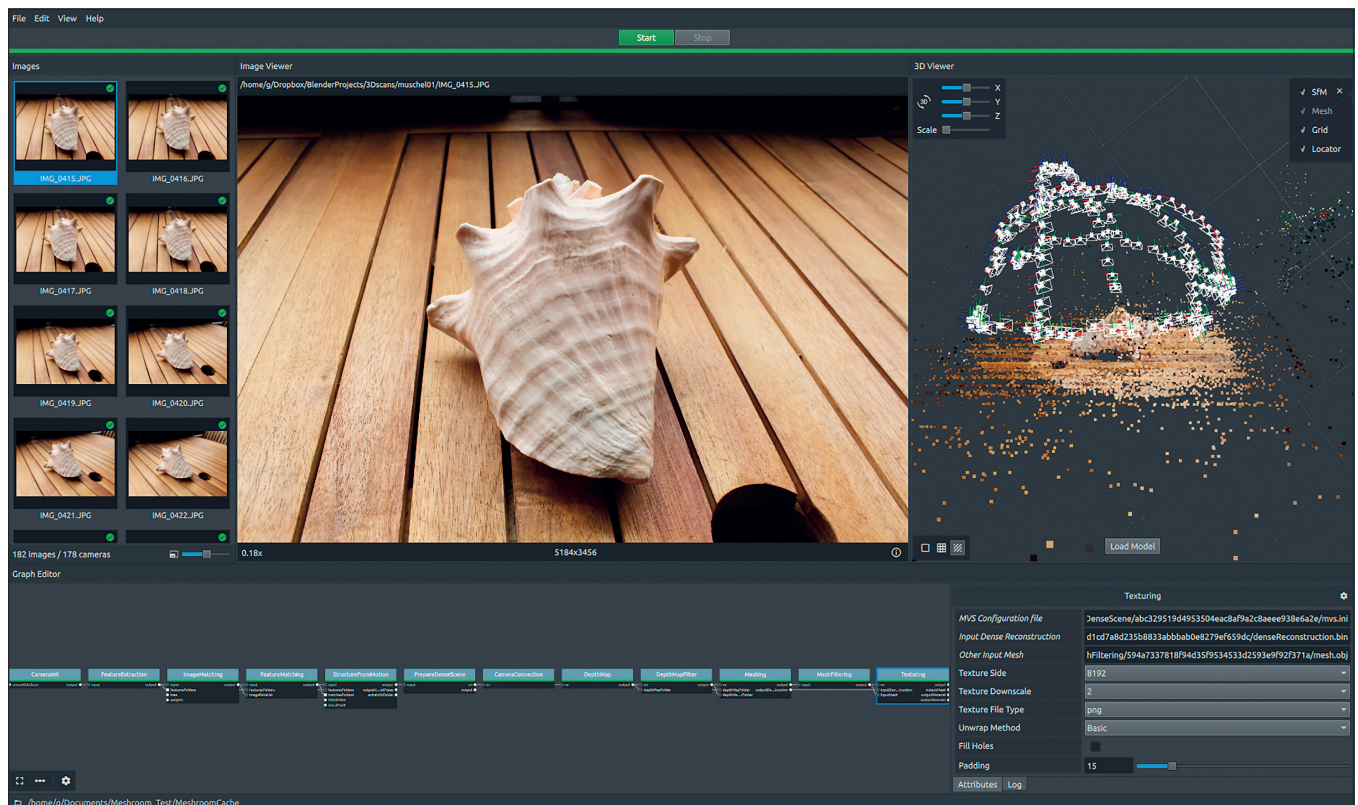
Meshroom – Open-Source-Photogrammetrie auf einen Klick

Es gibt inzwischen zahllose Lösungen für Photogrammetrie bzw. 3D-Scanning, also die Erstellung von dreidimensionalen Objekten aus einer Serie von Fotos. Einige davon sind Open Source und mehr oder weniger aufwendig zu bedienen. Mit Meshroom ist ein neuer Player erschienen, der mit einer einfachen Ein-Klick-Bedienung glänzt und sowohl unter Windows als auch unter Linux funktioniert. Dank Caching und eines Node-basierten Bedienkonzepts gibt es aber trotzdem Möglichkeiten, einzugreifen.

von Gottfried Hofmann

So karg wird man von der Benutzeroberfläche von Meshroom begrüßt. Man könnte auch sagen: aufgeräumt.





Bei Meshroom handelt es sich streng genommen um eine GUI für Alicevision, ein Open-Source-Framework für Photogrammetrie, welches durch die Zusammenarbeit mehrerer Universitäten mit dem VFX-Studio Mikros Image aus Frankreich entstanden ist. Das Projekt wurde auch mit Mitteln der Europäischen Union gefördert. Entwickelt wurden Werkzeuge, die sich gut in bestehende VFX-Workflows integrieren lassen und auch für weitere akademische Forschungen genutzt werden können. Das sollte man im Hinterkopf behalten, wenn man sich die Node-basierte GUI von Meshroom und das Caching-Konzept ansieht. Meshroom an sich lässt sich direkt von bit.ly/meshroom als Binary für Windows und Linux herunterladen. Nach der Installation begrüßt einen die Benutzeroberfläche, die die Komplexität des Unterbaus gekonnt zu verstecken weiß.

Workflow

Die GUI ist sauber in Bereiche unterteilt. Zuerst lädt man Fotos per Drag-and-drop in den Bereich bzw. die Kachel ganz links. Dabei muss man unscharfe oder verwackelte Fotos nicht aussortieren, das erledigt später ein Algorithmus automatisch. Man sollte die Fotos auch nicht nachbearbeiten, denn dadurch können die Kamerainformationen in den .exif-Daten verloren gehen, und auf diese ist Meshroom angewiesen. Der Workflow heißt also ganz klar: Fotografieren und dann sofort in Meshroom.

Sollte Meshroom die genutzte Kamera nicht in der mitgelieferten Datenbank finden können, so erscheint eine Fehlermeldung und die Rekonstruktion kann nicht gestartet werden. Die Datenbank wird zwar ständig erweitert, beinhaltet aber natürlich noch nicht jede (Smartphone-)Kamera auf dem Markt. In dem Fall muss sie von Hand um die Sensordaten der eigenen Kamera erweitert werden. Ein anderer Grund, warum das Programm den Dienst verweigert, ist das Fehlen einer CUDA-fähigen Grafikkarte. Da nur ein einziger Schritt im Prozess CUDA voraussetzt, hoffen wir, dass in Zukunft ein Fallback in OpenCL oder direkt für die CPU implementiert wird.

Bevor es losgehen kann, sollte das Projekt noch gespeichert werden. Am besten speichert man jedes Rekonstruktionsprojekt in einen eigenen Ordner, denn Meshroom legt eine Reihe von Unterordnern mit dem Ergebnis und den Zwischenschritten an, die allerdings standardmäßig mit Hashes bzw. zufällig erscheinenden Zahlen- und Buchstabenkombinationen benannt werden. Solange man in dem Ordner sonst nichts hat, ist das aber kein Problem. Alternativ lässt sich der Ort für das finale Mesh inklusive Texturen auch nach eigenen Wünschen festlegen, aber dazu später mehr. In der Praxis ist das Vorgehen mit jedem Projekt in einem separaten Ordner völlig ausreichend.

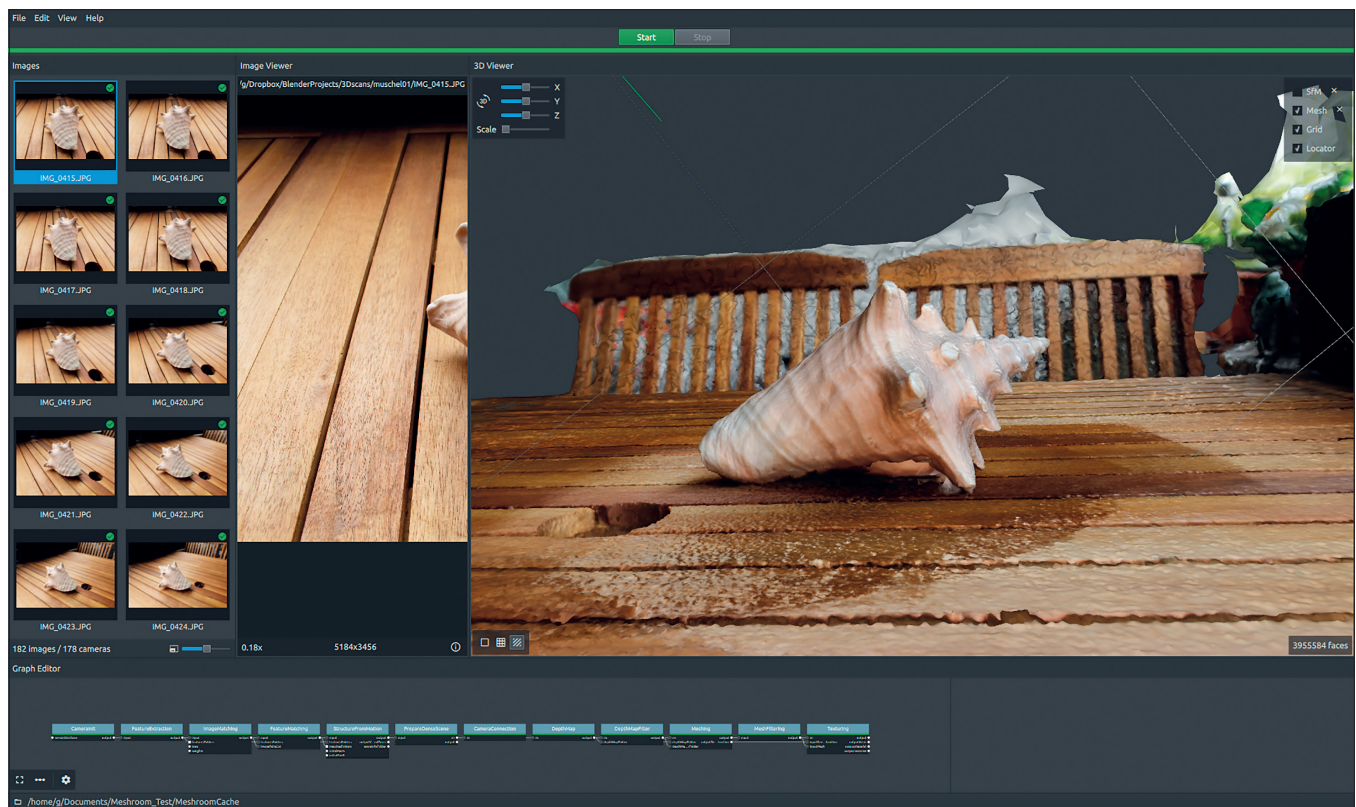
Nach dem Speichern beginnen wir den Prozess mit einem Klick auf Start. Das kann je nach Anzahl der Fotos mehrere Stunden in Anspruch nehmen. Meshroom ist defini-

So sieht Meshroom nach einer erfolgreichen Rekonstruktion aus. Alle Nodes sind im grünen Bereich, und im rechten Fenster sind die Punktwolke sowie die Kamerapositionen visualisiert. Will man noch das fertige Mesh sehen, so muss zuerst auf „Load Model“ geklickt werden. Der Knopf findet sich im Vorschaufenster unten mittig.

tiv nicht das schnellste Programm auf dem Markt. Dafür lässt sich die Arbeit jederzeit pausieren und kann dann später wieder aufgenommen werden. Da alle Daten auf die Platte gecacht werden, ist es auch möglich, das Programm zu beenden und an einem anderen Tag weiterzumachen. Oben wird in einem Fortschrittsbalken über Farbcodierungen angezeigt, wie weit die Berechnungen fortgeschritten sind, unten bei den Nodes kann man noch detaillierter sehen, an welchem Schritt gerade gearbeitet wird. Während der Prozess abläuft, sieht man im rechten Fenster Stück für Stück, welche Informationen gesammelt worden sind, z.B. die Punktwolke oder die Position der Kameras.

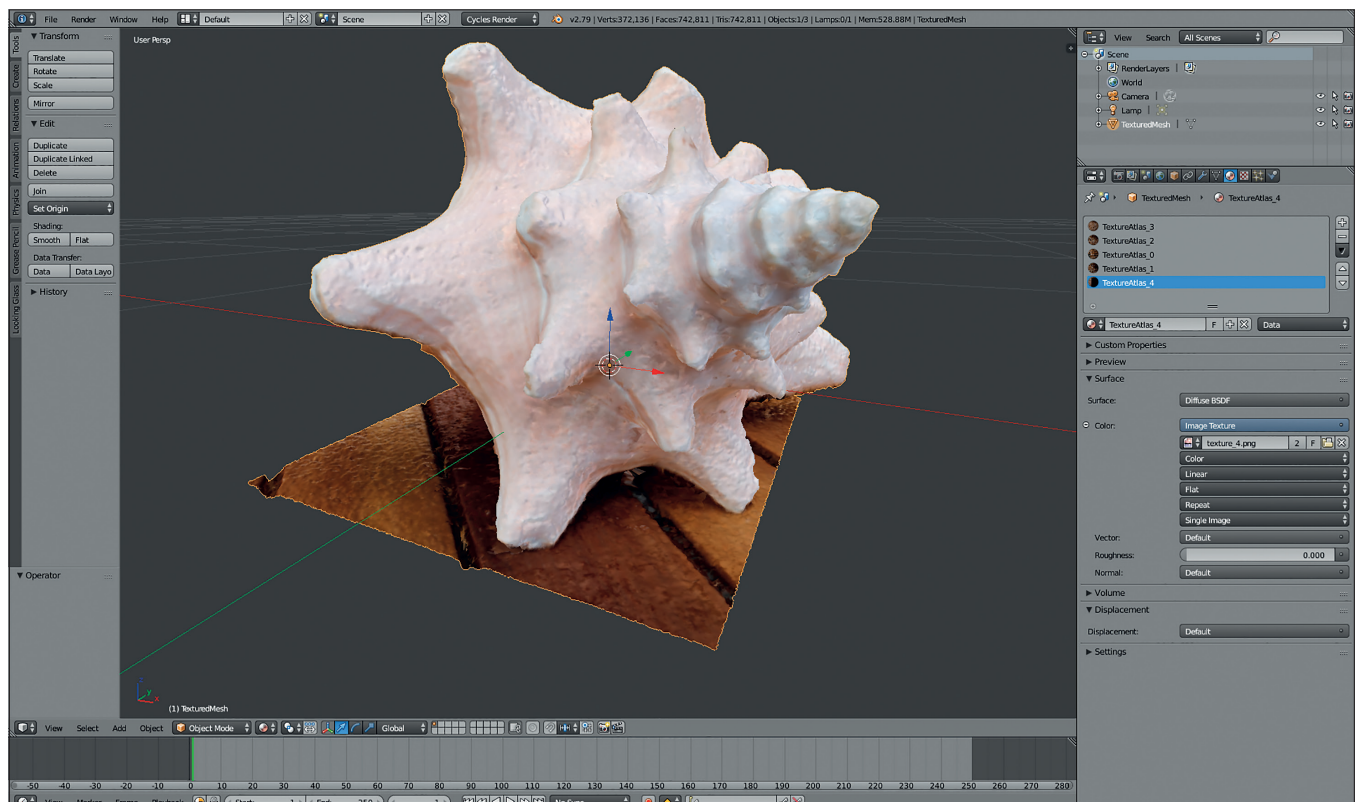
Fertig – was nun?

Wenn die Rekonstruktion abgeschlossen ist, wird noch kein 3D-Modell gezeigt. Das muss erst mit einem Klick auf „Load Model“ geladen werden. Zunächst wird das Modell mit überlagerter Punktwolke angezeigt. Wenn man das Häkchen bei „SfM“ entfernt, verschwindet Letztere zusammen mit den Kameras und man kann ungestört den 3D-Scan betrachten.



Meshroom nachdem das 3D-Modell geladen und die Anzeige für die Punktwolke sowie die Kamerapositionen ausgeschaltet wurden. Beim Beispiel-Scan ist sehr viel der Umgebung enthalten. Um das zu entfernen, muss das Modell in ein dediziertes 3D-Programm wie z.B. Blender geladen werden.

Der Beispiel-Scan in Blender geladen und nachbearbeitet. Meshroom hat in diesem Fall die Texturen der Muschel auf fünf Dateien aufgeteilt. Auch diese Entscheidung trifft Meshroom automatisch. Man kann die Anzahl der Texturen, deren Format und Auflösung allerdings auch selbst im Texture Node von Meshroom definieren.



Dabei wird einem schnell auffallen, dass der Betrachter auch wirklich nur zum Betrachten gedacht ist. Werkzeuge wie eine Maskierung für nicht benötigte Bereiche fehlen komplett. Die Nachbearbeitung sollte stattdessen in einem dedizierten 3D-Programm geschehen. Dafür speichert Meshroom das Ergebnis als .obj-Datei mit einer oder mehreren Texturen. Aber wo befinden sich die? Hierbei hilft ein Blick auf den letzten Node im Tree, der sich Texturing nennt.

Nodes

Klickt man auf einen Node, so sieht man im Fenster rechts des Node Trees dessen Einstellungen. Beim Node Texturing finden sich in den Einstellungen ganz unten „Output Folder“, „Output Mesh“ und „Output Textures“. Hier erfährt man, wo Meshroom das Ergebnis hingespeichert hat. Wenn man für das Projekt, wie im Artikel weiter vorne vorgeschlagen, einen eigenen Ordner angelegt hat, dann ist es leicht zu finden: Projektordner > MeshroomCache > Texturing > Wirre-Zahlen-Buchstaben-Kombination.

Nun kann man das Mesh mit einem 3D-Programm der Wahl öffnen und weiterbearbeiten. Für Maya existiert zudem ein Plug-in, das neben dem Mesh an sich auch die Kamerapositionen und Point Cloud importiert inklusive der Information, welches Bild die Kamera gemacht hat. Das wiederum kann zum Texture Painting eingesetzt werden, während die Point Cloud für Korrekturen des Meshes genutzt werden kann. Weiterhin existiert ein OFX-Plug-in für Nuke und Konsorten, mit dem die Kameraposition eines Bildes aus einem Meshroom-Scan rekonstruiert werden kann.

Scan verfeinern

Die Tatsache, dass sämtliche Zwischenergebnisse auf Platte geschrieben und dort belassen werden, erlaubt nicht nur das Pausieren und Wiederaufnehmen eines Scans oder den Import der Daten in Maya mittels Plug-in. Meshroom bringt auch Nettigkeiten mit wie die Möglichkeit, einem bereits abgeschlossenen Scan weitere Bilder hinzuzufügen. In dem Fall werden automatisch neue Nodes an der richtigen Stelle erzeugt und eingefügt, und Meshroom integriert die neuen Informationen in die bestehenden Daten. Wenn also einem Scan ein paar Details fehlen, kann man an den entsprechenden Stellen weitere Fotos machen, und Meshroom schließt dann hoffentlich die Lücken. Das Ganze kommt allerdings zu einem Preis.

Ein Meshroom-Projekt kann leicht mehrere Gbyte groß werden.

Zukunftsmusik

Meshroom ist mit dem Node-basierten Konzept inklusive Cache-System so aufgebaut, dass man sich über verschiedene Wege mit eigenen Algorithmen in die Pipeline einklinken kann. Es gibt sogar eine Schnittstelle für Matlab. Das macht es für Forschung und Entwicklung besonders interessant. Daher verwundert es nicht, wenn bereits einige Erweiterungen geplant sind. Am interessantesten ist sicherlich die Möglichkeit, einen 3D-Scan eines Films zu erstellen und diese Daten später für das Kamera-Tracking einsetzen zu können. Leider sind die Funktionen noch nicht fertig, es klingt nach einem interessanten Einsatzgebiet für das Caching-Konzept von Meshroom. Weiterhin ist ein Plug-in geplant, das interessante Frames aus einem Video extrahiert (z.B. Drohnenaufnahmen) und diese fürs 3D-Scanning einsetzt oder anders ausgedrückt: Meshroom soll Unterstützung für Videos erhalten.

Ein anderes Forschungsprojekt dreht sich derzeit darum, wie man aus einem 3D-Scan die Albedo- bzw. Farbinformation des Objekts extrahieren kann oder anders ausgedrückt: Wie würde ein Objekt ohne Schatten oder Verfärbungen durch Licht aussehen? Denn um Licht und Schatten kümmert sich später die 3D-Software, in einem Scan hingegen sind sie eher störend.

Fazit

Meshroom ist ausgesprochen einfach zu bedienen, bietet gute Ergebnisse und kann mit interessanten Konzepten aufwarten. Außerdem funktioniert es nicht nur unter Windows, sondern auch unter Linux. Leider ist die Software im Vergleich zu Konkurrenten wie z.B. Reality Capture ausgesprochen langsam. In unseren Tests war die Geschwindigkeit noch annehmbar, solange die Anzahl der Bilder im zweistelligen Bereich blieb.

> ei



Gottfried Hofmann ist Diplom-Informatiker und bietet seit mehreren Jahren professionellen Support sowie Schulungen für die freie 3D-Software Blender an. Als freischaffender Autor schreibt er für Fach- und Computerzeitschriften. Er hat zahlreiche Blender-Tutorials verfasst, u.a. für CG Tuts+ und CG Cookie. Weiterhin betreibt er die Webseite www.Blender-Diplom.com, auf der Blender-Tutorials in deutscher und englischer Sprache zur Verfügung stehen und Schulungen gebucht werden können.

Alicevision

DP: We are currently here at FMX with the team from Mikros that created Alicevision. A few days ago, they released Alicevision's main program that is Meshroom, a full-featured stand-alone photogrammetry solution. Why are you releasing it as open-source?

Benoit Maujean: Since the beginning of the project (2010), we have wanted it to be open-source and able to be integrated within other software. We chose the Mozilla Public License v2, so if you make modifications to the public code, you push it back to the repository. But we want to maximize the partnership with both academic institutes and industry without any problem.

DP: It seems that your program is really good for educational use because the user has the option of simply pushing a button, but you also literally expose the different steps of the internal workflow by making use of a node graph. The user can even optimize this way for example by using a custom base mesh. Did you have educational usage in mind?

Fabien Castan: Yes, on the website we try to explain the high-level notion for each step and make reference to the publications we are using. We are working with four different research labs. They have the ambition to be able to create student projects on top of it because all the relevant stuff is implemented and they can combine it to create some specific research environment.

Benoit Maujean: They can even use Matlab to test their algorithms. It might not be optimized, but they can test their algorithms in the context of the global pipeline. They can add their nodes in Meshroom and see how the algorithm can fit with the rest of the pipeline because everything else is constant before and after. We are doing that with our PHD, who is testing algorithms regarding the analysis of the materials and lighting of the scene. He is able to test it in the context of global reconstruction.



The main demo image of Alicevision which is the workhorse behind Meshroom.

Fabien Castan: And it is also the idea of having a tweakable pipeline. Advanced users have access to low-level parameters to see how the program reacts and analyse it. And you can even interact with it directly in the UI and compare the results directly without going to the command line.

Yann Lanthony: It is important to say that at first we did not have Meshroom as it is now. When we were working on an algorithm and wanted to test the impact of parameters, we had to manually write command lines, make sure to write an output, change parameters, etc. You know you get mixed up with all those terminals on your screen, you get confused which one is which. When we first had the really very first version of Meshroom which basically was just the graph editor for ourselves it changed things rapidly. We said goodbye to the terminals. Meshroom is a development tool for us as well.

DP: For a casual user there is just one button and for the experts there is the extremely finely grained control of every step and parameter. TDs and larger studios seem to be in the target market for this tool.

Fabien Castan: We are TDs at Mikros, so that is our everyday job. People in produc-

tion use the one-button-click, some more advanced users tweak things on their own. But when people have trouble on a dataset that does not work as expected, we can analyse the console and analyse how Meshroom reacts. That allows us to improve the algorithms by just seeing that there is a problem on a certain part and that a parameter is critical, because on this dataset it reacts in a certain way and changes the impact. So, for instance, we might want to compute this setting automatically for this kind of use-case.

Yann Lanthony: A recent example of this is a series of shots of actors' faces on set. The one-button-click result was not so great because of the subtle movements of the actors' faces during the shooting, which resulted in offsets during reconstruction. We were able to adjust some parameters to make sure we got rid of a maximum of those outliers by applying advanced denoising.

Benoit Maujean: Regarding the TDs – the fact that it is all Python behind the scenes is very useful. The best example is the integration with render farms. That was one of the first things that was expected, the fact that you can use other workstations around you or the render farm computers to parallelize reconstruction.

Fabien Castan: The submission to render farms can be customized by TDs as well because there is an abstraction layer. We already have people working with us who are using different render farms.

DP: Meshroom currently takes photos as sources, and in the future it will also ingest LIDAR data. How about video?

Yann Lanthony: We currently have a software that is able to extract interesting keyframes from videos, such as frames that are not too close together and not too blurry. It is not integrated into Meshroom yet, so we currently run this process in the background. It is not yet available in the high-level UI. It is ready, but it is not yet fully integrated.

DP: In the presentation you showed an example use-case where you first 3D-scanned a movie set and then used that data for the camera solve in a tracking shot. The result was a shot where the camera moved through the point cloud of the previous reconstruction. A process that opens up a lot of possibilities.

Fabien Castan: Yes, that is pretty interesting, but not production-ready yet. Currently, the part that is production-ready is the

AliceVision

reconstruction of the movie set. But we are working on both an offline solution for high-quality camera tracking as well as real-time camera tracking to make augmented reality on the shooting set for previz. The offline camera tracking can rely on natural features and markers, but the previz solution requires markers. We have developed a new marker type called CCTag which provides a nice robustness to challenging conditions regarding motion blur, depth of field, low lighting, and partial occlusions.

DP: You have close partnerships with academia ...

Benoit Maujean: Yes, we are working with people in research labs, but the objective is also to test in the production context. To be

robust and to be used in the everyday life of movie production.

Fabien Castan: Currently, we have partnerships with academics and hardware manufacturers, but we are also very open and

interested in collaborations with other industries. Photogrammetry is not limited to the postproduction area but can be used for many things. It is also the foundation of augmented reality, both in realtime and offline.



Benoit Maujean
is CTO and R&D Manager
at Mikros Image.



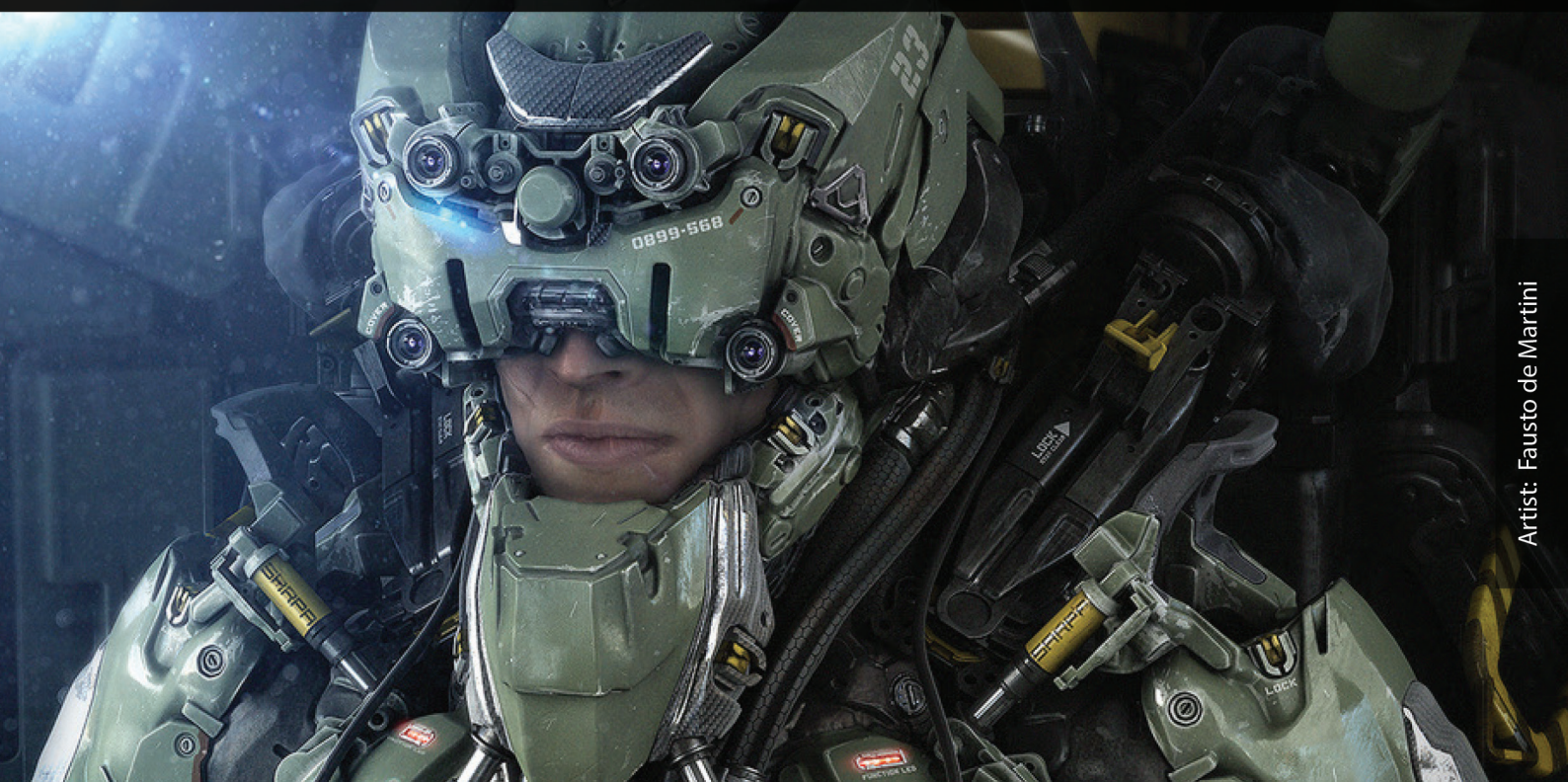
Fabien Castan
is R&D Engineer
at Mikros Image.



Yann Lanthony
is R&D Engineer
at Mikros Image.

Anzeige

CGSOCIETY



Artist: Fausto de Martini

Networking Platform for Professional Digital Artists
www.cgsociety.org