

2018

ISSN 1433-2620 > B 43362 >> 22. Jahrgang >>> www.digitalproduction.com

Publiziert von DETAIL Business Information GmbH

Deutschland € 17,90

Österreich € 19,-

Schweiz sfr 23,-

6

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

NOVEMBER | DEZEMBER 06:2018



Game Engines

Realtime und Interactive –
aber welche taugt für was?

Praxis!

Lighting, ZBrush, Scratch 9,
Gaea, Flame und mehr ...

animago 2018!

Alle Gewinner des
diesjährigen Awards!



Omniburger

360-Grad-Video ist derzeit in aller Munde und in vielen Köpfen, so auch an der Hochschule Hamm-Lippstadt (HSHL). Im Rahmen unserer Projektarbeiten an der HSHL im Studiengang Computervisualistik und Design untersuchten wir zum einen, wie der Workflow mit multiplen Kameras aussieht, und zum anderen, wie die dazugehörigen Stativköpfe aussehen können.

von Olaf Erler und Kevin Otte

Im Ergebnis entwickelten wir so einen Workflow, der für drei bis zwölf Kameras angewendet werden kann, und Köpfe, die mit der entsprechenden Anzahl an Kameras arbeiten. Zu diesem Zweck stellten die betreuenden Professoren Christine Latein (Industriedesign, Lehrgebiet Computervisualistik und Design) und Stefan Albertz (Workflow, Lehrgebiet 3D-Animation und Visual Effects) 3D-Drucker und zwölf GoPro Hero 5 zur Verfügung. Die Entscheidung für die Hero 5 fiel aus zwei Gründen: der kompakte Formfaktor der Kameras und die variable Bildrate. Als Speichermedien nutzten wir die von GoPro empfohlenen SanDisk microSDXC 64GB.

Im Laufe unserer neunmonatigen Projektarbeit, die im Februar 2017 begann, konnten so acht Köpfe entwickelt werden, die sich in der Nachbearbeitung des erzeugten Materials voneinander unterscheiden.

Wie sind wir vorgegangen, um zu unseren Ergebnissen zu gelangen?

Um die Kameraköpfe zu gestalten, müssen vorab einige Aspekte berücksichtigt werden.

Testprozedere

Sowohl in der Entwicklung als auch mit dem fertigen Omniburger gingen wir folgendermaßen vor: Unsere Tests liefen so ab, dass wir unseren Testkopf aufstellten und die Aufnahme durchführten. Das Auslösen aller zwölf Kameras erfolgte bei manchen Tests manuell, bei anderen lösten wir mit der Smart Remote von GoPro aus. Da uns nach ersten Tests bewusst war, dass die manuelle Auslösung nichts nützt, behielten wir die Smart Remote bei. Beide Methoden haben den Nachteil, dass nicht synchron ausgelöst wird. Unsere Testaufnahmen führten wir mit 50 fps durch und konnten bei der Smart Remote eine Latenz von bis zu zwölf Frames feststellen.



Ziel der Designs ist es, mehrere Kameras in einer bestimmten Ausrichtung und Anordnung zu fixieren, um das gewünschte Ergebnis zu erzielen, in unserem Fall die Erzeugung eines zylindrischen oder sphärischen 360-Grad-Videos.

Zu diesem Zweck beschäftigte sich Kevin Otte mit den technischen und ergonomischen Besonderheiten, um funktionale und gut bedienbare Kameraköpfe zu planen. Relevante Punkte waren dabei neben der Befestigung der Kameras die gute Erreichbarkeit der wichtigsten Bedienelemente der Kameras. Hierzu zählen die Tasten „Modus“ und „Auslöser“.

Um den Designprozess anzustoßen, führten wir also erste Tests durch, anfangs mit Stellproben und Holzquadern. Die Ergebnisse dieser Tests führten zu den ersten einfachen Konstruktionen, die drei bis acht Kameras aufnahmen, und sollten so die Grundlage für die Entwicklung der komplexen Köpfe bilden.

Im Gesamten konnte so ein iterativer Gestaltungsprozess entstehen, um sich schrittweise der stereoskopischen 360-Grad-Aufnahme anzunähern. Der auf den Namen Omniburger getaufte Kopf wird von uns auch als Open-Source-Hardware zum Download angeboten. Mit diesem Stativkopf ist es möglich, omnidirektionale stereoskopische 360-Grad-Aufnahmen aufzuzeichnen. Die .stl-Files für den 3D-Druck können Sie auf digitalproduction.com/downloads kostenlos herunterladen.

Verarbeitung der Bilder

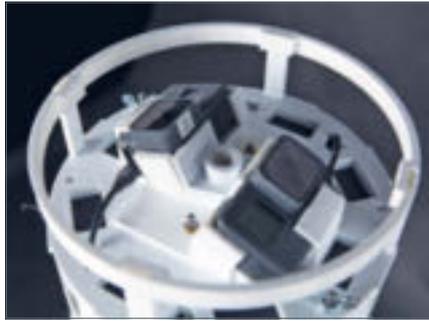
Für die Verarbeitung fiel die Entscheidung auf CaraVR von The Foundry, welches als Plug-in für die Nuke-Familie verfügbar ist. Vorab ist zu erwähnen: Run-and-gun/ Shoot-and-go ist mit dem Omniburger nicht vorgesehen, wodurch die Planung vorab an



Kabelmanagement: Alles geht nach unten.



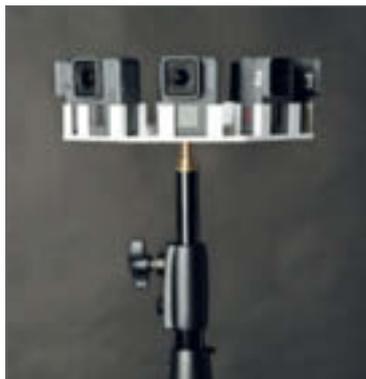
Bausatz ahoi: Wenn das kein überschaubares Rig ist, dann ...



Randsache: Alle Kameras liegen hinter der Kontaktebene – nichts wird zerkratzt, wenn die Kamera abgelegt wird. Und ein weiteres Detail: Die einzelnen Elemente haben Rollstopper, also kleine Ausbuchtungen, die ein Rollen der Rigs verhindern. Klingt komisch, aber wir kennen alle den speziellen Praktikanten ...



Die Nutzung des rückseitigen Touchdisplays spielte keine große Rolle. Ursache dafür ist, dass die omnidirektionale und kompakte Gestaltung des Kopfes das kaum möglich machte. Ein weiterer Aspekt, auf den später eingegangen wird, ist die Planung der Produktion. Im Designprozess wurde weiterhin darauf geachtet, dass der Statusbildschirm (GoPro Vorderseite) während des Betriebs ablesbar, ein integriertes Kabelmanagement vorhanden und dass die vordere Statusleuchte sichtbar war und die Mikrofone nicht verdeckt waren. So konnten wir es realisieren, dass die Kameras auch während der Nutzung mit Strom versorgt werden konnten.



Bedeutung gewinnt. Daher unsere Empfehlung: Nehmen Sie sich mehr Zeit in der Preproduktion, als es sonst der Fall ist. Bei unserem Projekt konnten wir diesen Punkt jedoch außer Acht lassen.

Für unsere Testaufnahmen nutzten wir den Campus Lippstadt der HSHL, da auch unsere Veranstaltungen dort stattfinden. Weiterhin hat der Campus ein Fotostudio, Foyers mit hohen Decken und Grünflächen drumherum – ideale Testumgebungen.

Synchronisation

Um die Synchronisation nach der Aufnahme durchzuführen, setzten wir Synchronisationspunkte durch zweimaliges Klatschen von unterschiedlichen Seiten. Zur Synchronisation nutzten wir Pluraleyes von Red Giant, mitunter aber auch den Multicam-Schnitt von Premiere Pro. Wenn Pluraleyes vorhanden ist, sollte dem der Vorzug gegeben werden. Nachdem die Synchronisation durch-

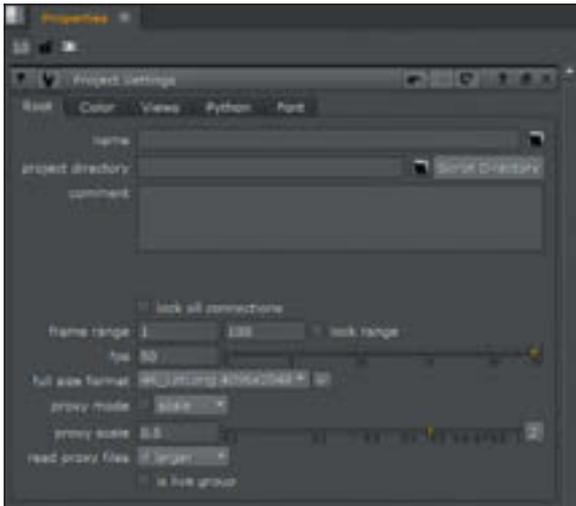
geführt worden war, gaben wir die Clips mit der halben Bitrate aus, da diese für unsere Tests ausreichend war.

Die Verarbeitung sollte nun in CaraVR beginnen. Während der Projektarbeit verwendeten wir Nuke X 10.5.2 mit CaraVR 1.0v4. Nachdem unsere Testclips synchronisiert und exportiert waren, wurden sie in Nuke X importiert. Das Projektformat wurde auf „LatLong“ eingestellt und die korrekte Bildrate angegeben. Im Anschluss daran wurde der C_CameraSolver erzeugt. Dieser dient dem Auflösen der Kameraanordnung und erzeugt den Prestitch. Der C_CameraSolver verwendet als Preset Einstellungen für die Hero 4. Die Einstellungen müssen überschrieben werden. Der erste zu ändernde Parameter ist „Rig Size“ auf dem Reiter „Cameras“. Der Durchmesser des Omniburgen beträgt 19 cm und wir gehen davon aus, dass der Sensor etwas weiter zur Kopfmittle platziert ist. Als nutzbare Werte können 18 cm und 18,5 cm gesetzt werden. Anschließend deaktivierten wir die Checkbox „Match in overlap region“, CaraVR sucht dann außerhalb des Überlappungsbereichs der Kameras. Das bedeutet längere Rechenzeit, sorgt aber für einen stabileren Prestitch. Sobald diese beiden Einstellungen erfolgt sind, kann das Keyframeintervall auf dem Reiter „C_CameraSolver“ angepasst werden.

Das Keyframeintervall gibt an, zu welchen Keys die Featureerkennung erfolgt. Die Auflösung des Prestitches funktioniert mit einem Keyframe, aber in der Praxis zeigten sich Werte zwischen 50 und 125 Frames als gut nutzbar, um einen soliden Prestitch zu erzeugen. Zu berücksichtigen sind hier die 50 fps, mit denen die Basisvideos erzeugt wurden. Bei einer Änderung der Framerate sollte das Intervall im Ein- bis Zweisekundenbereich liegen.

Nachdem das Intervall eingestellt und mit einem Klick auf „Key All“ erzeugt worden ist, kann der Prestitch mit Betätigung des Buttons „Match“ durchgeführt werden.

Das Matching hängt von der verwendeten GPU ab. Während der Projektarbeit standen die Quadro K4200, GTX 1080 und GTX 1080 Ti von Nvidia zur Verfügung. Letztere lieferte hier am schnellsten ein Ergebnis. Nach erfolgreichem Matching sollte der Prestitch mit „Solve“ aufgelöst werden. Sofern der Prestitch auf dem Kopf steht, wird er um 180 Grad in Z-Richtung rotiert, dies erfolgt auf dem Reiter „Settings“. Als nächstes kann die Ausrichtung der Horizontlinie erfolgen. Dafür kann bei aktiviertem C_CameraSolver in den Viewer geklickt und mit betätigen von Strg+Alt+LMB verschoben



werden. Sollte eine Kamera angeklickt werden, so ist diese hervorgehoben durch einen gelben Kreis um die Kameranummer. Hier ist etwas Vorsicht geboten.

C_ColourMatcher

Im nächsten Schritt wird ein C_Colour-Matcher Node erzeugt. Wie der Namen erahnen lässt, soll dieser dafür sorgen, ein

möglichst homogenes Bild zu erzeugen. Während unserer Tests stellten wir die Kameras auf den ProTune-Modus mit 5.600 K, ISO 400 und der Bequemlichkeit halber auf automatische Belichtung. Wir untersuchen derzeit, wie die Belichtung am besten eingestellt werden kann. Gerade bei Außenaufnahmen ist dies ein nicht zu vernachlässigender Parameter.

Bei unseren Omniburgetests, die wir im Herbst des letzten Jahres durchführten, sollte eine Belichtungszeit von 1/50 ausreichen.

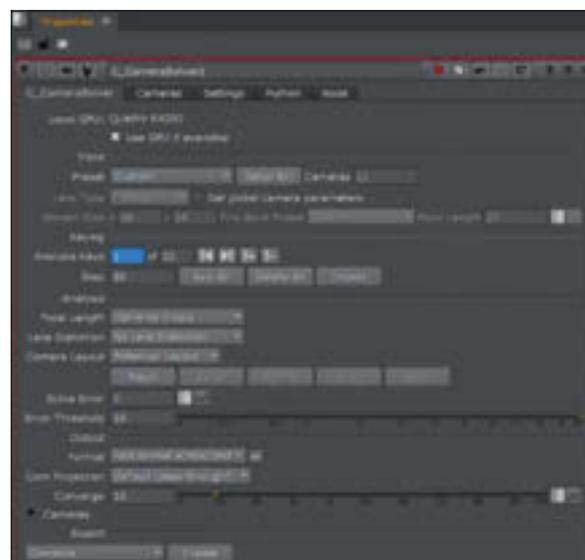
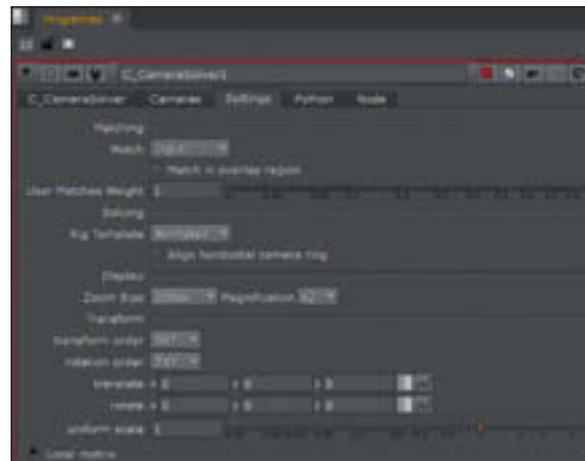
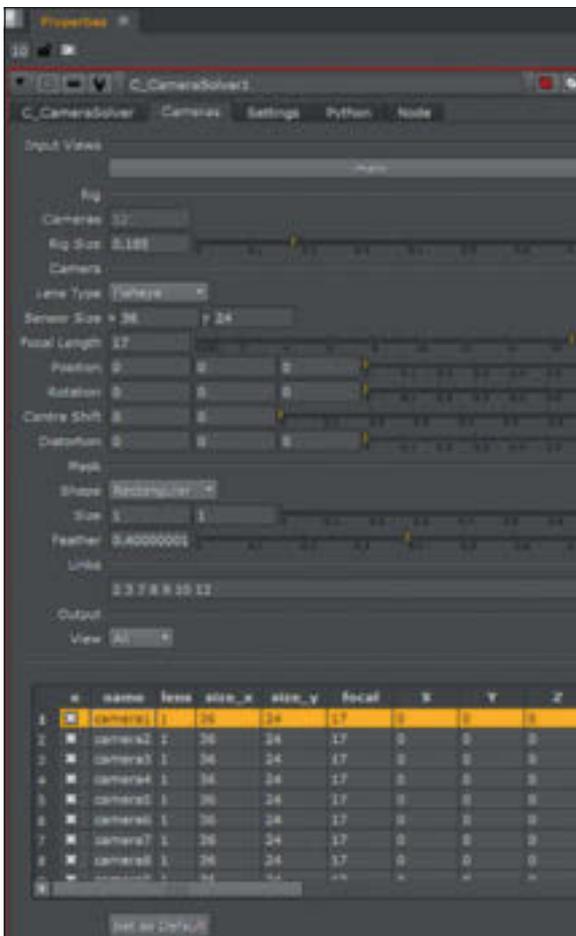
Der C_ColourMatcher verfügt über die gleiche Einstellungsmöglichkeit des Keyframeintervalls wie der C_CameraSolver und kann auf Belichtung und Farben prüfen. Wir können empfehlen, den C_ColourMatcher in jedem Stitching zu nutzen, um kleinere Farbdrifts und Belichtungsunterschiede auszugleichen.

Nach fertigem Prestitch und einer einfachen primären Farbkorrektur wird der

eigentliche Stitch erzeugt. Hierzu wird ein C_Stitcher Node direkt nach dem C_Colour-Matcher erzeugt.

Die Checkboxes auf dem ersten Reiter lassen wir für den Moment außen vor und passen erneut das Keyframeintervall an, das standardmäßig auf 30 Frames eingestellt ist. Als Richtwert kann hier erwähnt werden, dass wir die besten Ergebnisse mit einem Abstand von einer halben Sekunde ausgehend von der Bildrate des aufgenommenen Materials erzielen. Während die beiden vorherigen Nodes einmal das Intervall bei der Prüfung durchlaufen, ist der C_Stitcher gewissermaßen dynamisch und prüft bei jeder Wiedergabe, vor allem aber nach Veränderung entweder im C_Stitcher selbst oder wenn in den Quellen Veränderungen vorgenommen wurden.

Die zuvor angesprochenen Checkboxes dienen folgendem Zweck: Während „Warps“ für die perspektivisch korrekte Verzerrung auf unserer Kugel/Zylinder sorgt, dient „Override Cameras“ dem nachträglichen Verändern der Kameraverbindungen. Um die Kameraverbindungen zu überschreiben, wird wieder der C_CameraSolver aktiviert. Auf dem Reiter „Cameras“ ist eine Tabelle





mit den verwendeten Kameras sowie deren Positionierung auf dem Rig. Sobald eine Kamera in der Tabelle ausgewählt ist, sieht man über dieser ihre Verbindung mit anderen Kameras. Da nicht jede Kamera mit einer anderen überlappt, können diese Verbindungen angepasst werden. Das ist zwingend notwendig, wenn ein stereoskopischer Stitch erzeugt wird. Für monoskopische Stitches ist es nicht notwendig. Womit wir bei der letzten der drei Checkboxes wären: „Enable Stereo Stich“. Sobald diese aktiviert wurde, werden weitere Views im Viewer hinzugefügt. Bei zwölf Input-Views und Main-View werden 26 weitere hinzugefügt; pro Kame-

ra jeweils eine für links und rechts sowie zwei weitere korrespondierende Main-Views für das linke und rechte Auge. Im Anschluss an die Aktivierung des Stereo-Stiches werden die Views für links und rechts auf dem Reiter „Stereo“ festgelegt. Die Distanz zwischen den Augen wird mit 6,5 cm auf dem Standardwert belassen. Der „Falloff Type“ kann auf Cosinus belassen werden. Er legt fest, wie sich die Verzerrung zum Nord- und Südpol unserer Sphäre verhält.

Der Reiter „Vector“ lässt die Steuerung des Warpens zu. Hier sorgen höhere Werte für längere Rechenzeiten, die Standardeinstellungen liefern in der Regel ein sehr gutes Ergebnis.

Anzeige

www.3darchitettura.com

3D ARCHITETTURA

DIGITAL ARCHITECTURE MAGAZINE



STANDALONE UTILITY FOR
IMAGE-BASED LIGHTING
custom 32-bit .exr files on
www.real-hdr.com



STANDALONE UTILITY FOR
PHOTOMETRIC LIGHT CREATION
custom type-C .ies files on
www.real-ies.com

Working with CG artists since many years and being CG artists ourselves, we have developed our own tools that can facilitate 3D visualization process. They are software-agnostic and can be easily inserted in your workflow to create custom lighting scenarios for photorealistic 3D renders and animations, both offline and real-time.

CREATED BY CG ARTISTS FOR CG ARTISTS



Stitch

Ein Parameter, der bisher noch nicht erwähnt wurde, ist der Konvergenzparameter. Dieser legt fest, wo Objekte oder Personen ineinander überführt werden. Hierzu ein Beispiel: Es steht ein Schauspieler vor der Kamera und hat einen Abstand von drei Metern zum Kamerasystem. So ist der Konvergenzparameter mit drei anzugeben, in diesem Umkreis nähern sich also alle gefilmten Objekte an und wir reduzieren so Ghosting in den Überlappungsbereichen. Ghosting ist daran zu erkennen, dass um Objekte oder Darsteller eine semitransparente Hülle zu sehen ist, gewissermaßen geisterhaft. Wenn das im Stitch auftritt, muss viel rotoskopiert werden, doch das muss an der richtigen Stelle erfolgen. Hierzu später mehr. In einem Test kann man es begrenzen, indem festgelegt wird, dass beispielsweise keine Aktionen in den Überlappungsbereichen durchgeführt werden. In der Praxis wird es jedoch etwas schwieriger. Um den Konvergenzparameter nicht jedes Mal anzupassen, kann er per Expression Link verbunden werden (1.0v4), ab CaraVR 2.0 erfolgt es automatisch.

Jetzt wo der Stitch erzeugt ist, geht es ans Exportieren unserer Bilder. The Foundry selbst empfiehlt das Multi-View OpenEXR-Format (.sxi), wir haben uns dem angeschlossen. Es wird ein Write Node erzeugt und festgestellt, dass dieser über einen CaraVR-Reiter verfügt. Dieser wird aufgerufen und die zu schreibenden Views können festgelegt werden. In Abhängigkeit dessen, was nach dem Export editierbar sein soll, nehmen wir sicherheitshalber alle Views. Nicht unerwähnt bleiben soll, dass eine

Edit

Was muss nun getan werden, um Änderungen an unseren Bildern vorzunehmen? Hierzu gibt es verschiedene Möglichkeiten. Das Eingangsmaterial kann entweder vor oder nach dem Stitching editiert werden, oder man editiert das exportierte Material.

CaraVR ist nicht nur in der Lage, einen Stitch zu erzeugen, mit den enthaltenen Werkzeugen können auch bereits erzeugte 360-Grad-Videos bearbeitet werden.

Die Bearbeitung des Eingangsmaterials kann mit den Standard-Nuke-Werkzeugen erfolgen. So kann bereits während des Prestitches eine Komposition erzeugt werden. Wenn wir an den Nuke-Nodegraph denken und die Reihenfolge C_CameraSolver > C_ColourMatcher > C_Stitcher haben, befinden wir uns vor dem C_Stitcher.

Sofern die Bearbeitung nach dem Stitching erfolgt, werden die Nodes hinter C_Stitcher eingefügt.

Nuke liefert zu diesem Zweck „Split and Join Views“ mit, damit werden die einzelnen Ansichten wieder aufgetrennt und wir sind so in der Lage, einzelne Ansichten zu editieren. „Split and Join“ kann sowohl im Prestitch als auch nach dem eigentlichen Stitching durchgeführt werden. Bei Änderungen im Prestitch übernimmt C_Stitcher die Aufgabe des Zusammenfügens, Änderungen nach C_Stitcher müssen mit C_Blander wieder zusammengefügt werden.

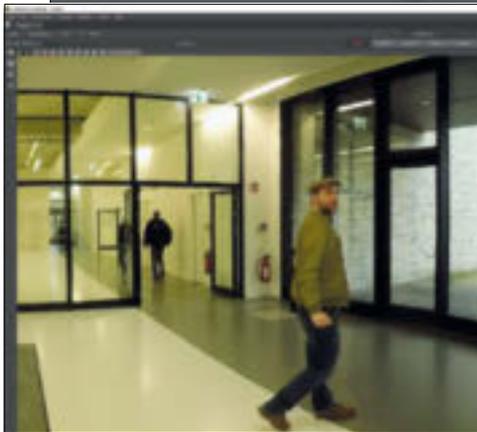
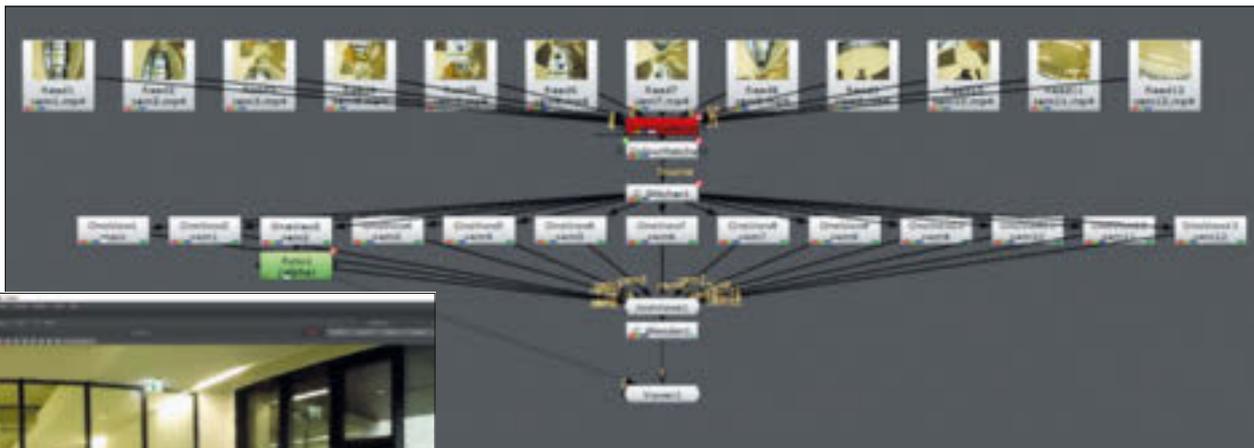
Hierzu nun ein einfaches Beispiel, bei dem Ghosting entstanden ist und korrigiert werden soll. Am Kopf der Person in der Bildmitte tritt es auf. Der C_CameraSolver wird aktiviert und es wird geprüft, welche Ka-

meras diesen Fehler verursachen. Nachdem festgestellt wurde, dass sich die Person in Kamera zwei und drei befindet, wird Kamera drei bearbeitet. Hier kann die Entscheidung recht einfach gefällt werden, in beiden Kameras ist der Abstand zum Überlappungsbereich fast gleich. Es wird eine einfache Maske um die Person gezeichnet und das Ergebnis betrachtet. Dafür geben wir den C_Stitcher aus. Nun stellen wir fest, dass die Änderung an dieser Stelle keinen Sinn macht, und wir können unsere „Split and Join“-Nodes wieder löschen.

In diesem Fall ist es sinnvoll, die Änderung nach dem C_Stitcher Node durchzuführen. Dafür werden die Ansichten nach dem C_Stitcher getrennt und wieder zusammengeführt (Split and Join). Nun wird wieder ein Roto Node an die entsprechende Kamera angefügt und eine einfache Maske gezeichnet. Im Anschluss daran wird der Maskenverlauf noch etwas angepasst, damit sich die Korrektur ins Gesamtbild einfügt. In unseren Tests zeigte sich, dass die Maske entsprechend der Bewegungsrichtung des Objekts über die einzelnen Views transportiert werden muss.

Wie wir am Ghosting erkennen können, ist es wichtig zu entscheiden, wo die Korrektur vorgenommen wird. Sollte an der falschen Stelle, in diesem Fall im Prestitch, eine Korrektur durchgeführt werden, kann der Stitch darunter leiden. So kann Ghosting zwar reduziert werden, doch flacht dadurch das Bild ab und es entstehen Verzerrungen an Kanten. Mit der Korrektur des gewarpten Materials tritt dieser Fehler nicht auf. Jegliches Ghosting sollte bereits während des Stitchings entfernt werden.

The Foundry hat in CaraVR 2.0 einen weiteren Node namens C_GlobalWarp eingeführt, der Ghosting reduziert und beim Ausgleichen von Verzerrungen hilft. Während unserer Tests stellte sich heraus, dass Ghosting entfernt und Compositings nachträglich integriert werden sollten. An dieser Stelle sollte der Export des Stitches erfolgen. Der Write Node befindet sich nun



dene Toolsets mit, unter anderem „Latlong Comp“ bestehend aus zwei miteinander verlinkten C_SphericalTransform und zwei Merge Nodes. C_SphericalTransform ist dabei das Allzweckwerkzeug, um Compositings in 360-Grad-Video zu erzeugen. Der Node wird genutzt, um die Projektionsart des Eingangsmaterials zu ändern.

Die beiden verlinkten Nodes des Toolsets sorgen dafür, dass beispielsweise ein Cornerpin oder eine Texteinblendung an der korrekten Stelle erscheint. Hierzu ändert

der erste Node die Projektionsart von Latlong nach Rectilinear, mergt den Cornerpin oder Text ins Bild und führt die Rückprojektion durch. Ein weiteres Toolset, das ähnlich funktioniert, ist „Latlong Rotopaint.“ Durch die Projektion von Latlong nach Rectilinear kann das Bild ohne Nutzung einer VR-Brille betrachtet werden – ein weiterer Vorteil, den C_SphericalTransform bietet.

Für Trackingaufgaben bietet CaraVR den C_Tracker Node. Er beruht auf dem Standard-Nuke-Tracker-Node, ist aber weiterhin

hinter dem C_Blender Node. Mit dem von Ghosting befreiten Stitch kann das Compositing beginnen. CaraVR liefert dafür verschie-

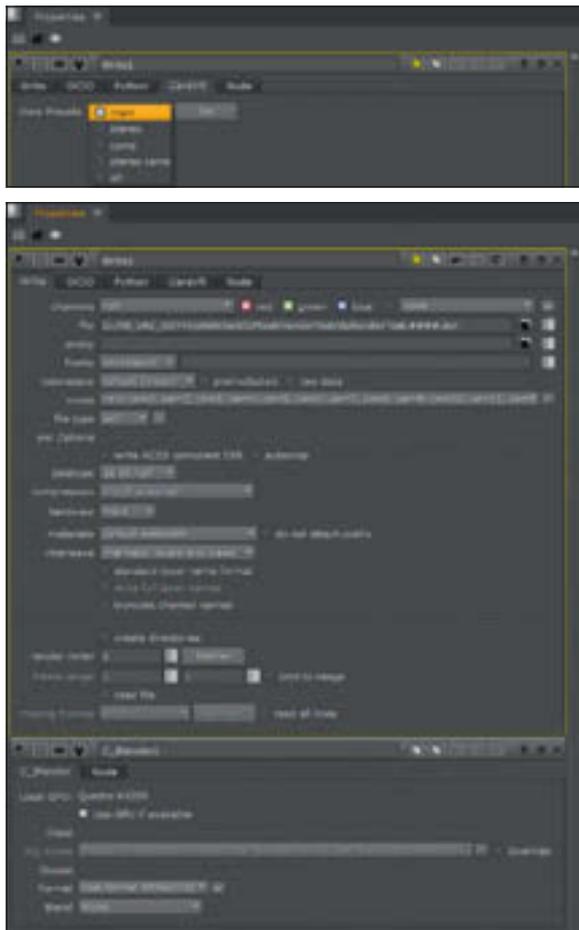
Anzeige



IN DER LEISTUNG EINMALIG. SECHSMAL IN DEUTSCHLAND.

Wenn Ideen zu Erfolgen werden, setzen große Namen oder inspirierte Talente in Sachen Postproduktion, Koproduktion und internationalem Vertrieb auf die Services von Deutschlands großem Filmdienstleister: ARRI Media.

Wir freuen uns auf alles, was Sie vorhaben. Innovation, Qualität und Partnerschaft. An sechs Standorten in Deutschland.



rektionales 360-Grad-Video zu erstellen. Dank der verwendeten Hero 5 sind wir auch in der Lage, ein sehr kompaktes Rig zu haben. Die Sichtbarkeit der Statusanzeige sowie die Möglichkeiten zur Kabelführung und externen Stromversorgung ermöglichen auch längere Drehs, sofern die Speicherkarten über ausreichend Kapazität verfügen. Die meisten unserer Tests führten wir mit 1440p50 durch, laut Displayanzeige hätten wir so 2:04 Stunden aufzeichnen können. Eine Empfehlung können wir dahingehend geben, dass 360-Grad-Videos nicht mit weniger als 50 fps gemacht werden sollten. Mehr ist in diesem Fall besser, bedeutet aber auch mehr Berechnungszeit im Stitching. Wichtig ist es auch, wie an anderer Stelle erwähnt, eine leistungsstarke Workstation zu nutzen. Diese sollte über eine schnelle CPU und GPU verfügen und entweder mehrere SSDs, ein sehr schnelles

in der Lage, Latlong-Material zu verarbeiten. Zudem können Preset Nodes für Matchmoving und Stabilisierung exportiert werden.

Neben den erwähnten Features von CaraVR bietet es seit Release die Möglichkeit, den kompletten Stitching-Prozess wie auch die Nachbearbeitung per VR-Brille zu überprüfen. Uns wurde die HTC Vive zur Verfügung gestellt. Um die Vorschau für VR-Brillen zu aktivieren, muss sich der Mauszeiger über dem Viewer befinden und „S“ gedrückt werden. Unter „Monitor Output Device“ können das jeweilige Headset und die Auflösung gewählt werden. Wurde ein stereoskopischer Stitch erzeugt, kann auch eine Stereo-Vorschau erfolgen.

Alles in allem liefert CaraVR alles für die Produktion von 360-Grad-Video, angefangen beim Stitching sowie der Nachbearbeitung. Hierfür liefert The Foundry bereits alles Nötige in Form des C_SphericalTransfer Nodes und des Latlong Toolsets mit. So können alle in Nuke verfügbaren Tools genutzt werden.

Fazit

Im Verlauf unserer Projektarbeiten konnten wir feststellen, dass unser Omniburger wichtige Parameter zur Produktion von 360-Grad-Videos erfüllt. Er ist in der Lage, ein vollständiges stereoskopisches omnidi-

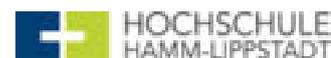
Festplatten-RAID oder SSD-RAID nutzen. Ein Testframe, der im Rahmen des Artikels erzeugt wurde, hat eine Größe von 97 Mbyte. Also wird auch viel Speicher benötigt. Bei der Wiedergabe des Stiches dürfte dies im Zweifel der Flaschenhals sein, der eine Echtzeitwiedergabe verhindert. Während bei monoskopischem Stitching der Aufwand bereits hoch ist, verdoppelt er sich im stereoskopischem Stitching.

Was uns zum wichtigsten Punkt bringt: Die Produktion muss gründlich geplant werden. So sollten jegliche Übergänge und Aktionen samt Radius vorher berücksichtigt sein. Für die Person, die das Stitching bzw. das Rotoskopieren des Ghostings übernimmt, wird so der Aufwand erheblich reduziert. Dies kann bei Studiodrehs einfacher realisiert werden als bei Außendrehs mit vielen Menschen, die sich unkontrolliert bewegen.

Anders als bei klassischen Drehs kann die Crew nicht vor Ort sein oder ist permanent zu sehen, so muss auch dies bei der Produktionsplanung berücksichtigt werden.

Wir konnten uns in unseren Projektarbeiten ausgiebig mit dem Thema 360-Grad-Video auseinandersetzen und wissen nun, was für die Produktion wichtig ist. Unserer Meinung nach müssen 360-Grad-Videos stereoskopisch sein und eine hohe Framerate haben. So können sie das Gefühl der Anwesenheit erzeugen und ein neues Videoerlebnis er-

HSHL



Die Hochschule Hamm-Lippstadt ist eine junge staatliche Hochschule mit Doppelcampus in Hamm und Lippstadt. Geboten werden den Studierenden moderne Lernbedingungen und eine Laborumgebung auf aktuellem Stand der Technik. Das interdisziplinäre Studienangebot legt den Fokus auf die Bereiche Ingenieurwissenschaften, Naturwissenschaften, Informatik und Wirtschaft. In der Forschung prägen Zukunftsthemen wie autonome Systeme, Computervisualistik und Design, Informationstechnologie und Elektromobilität die Hochschule.

Der Studiengang Computervisualistik und Design behandelt im Schwerpunkt Themen der Informatik und des Designs, lehrt u.a. Computergrafik, Bildverarbeitung, Animation, CAD und UX-Design und bietet in Wahlmodulen Vertiefungsmöglichkeiten in Visualisierung, Interaktionstechnologien und User Experience an. 2018 stellte er sich mit einem Messestand erstmals auf der FMX vor. Infos zu den Studiengängen und der Hochschule unter www.hshl.de.



Bild: HSHL/Barbara Wendling

möglichen. Ob Spielfilme oder Serien in 360 Grad funktionieren, lässt sich bisher schlecht abschätzen. Andererseits hat Alejandro G. Iñárritu mit „Carne y Arena“ gezeigt, wie 360-Grad-Video bzw. eine VR-Experience aussehen muss. Somit bleibt 360-Grad-Video spannend und die nächsten innovativen Produktionen in dem Bereich werden kommen. Für den Moment bleibt festzuhalten, dass wir uns auf einem Spielplatz der kreativen Möglichkeiten befinden. > ei



Olaf Erler studiert an der HSHL Visual Computing and Design mit Schwerpunkt Visualisierungen. Hierbei richtet er seinen Fokus auf 360-Grad-Video, gerne in Stereo und mit 120 fps. Zuvor hat er von IT-Mensch bis Marktbeschicker alles mögliche gemacht, und seit 2009 lassen ihn Filme nicht mehr los.



Kevin Otte ist Absolvent des Studiengangs Visual Computing and Design mit Schwerpunkt auf User Experience, Innovation und Produktentstehungsprozess. Während seiner Studienzeit war er Mitarbeiter im Labor für additive Fertigung und hat Tätigkeitsbereiche wie die Konstruktion und Fertigung von 3D-visualisierten Modellen betreut.