2018
6
ISSN 1433-2620 > B 43362 >> 22. Jahrgang >>> www.digitalproduction.com
Publiziert von DETAIL Business Information GmbH

Deutschland € 17,90
Österreich € 19,–
Schweiz sfr 23,–

DIGITAL PRODUCTION

DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

NOVEMBER | DEZEMBER 06:2018

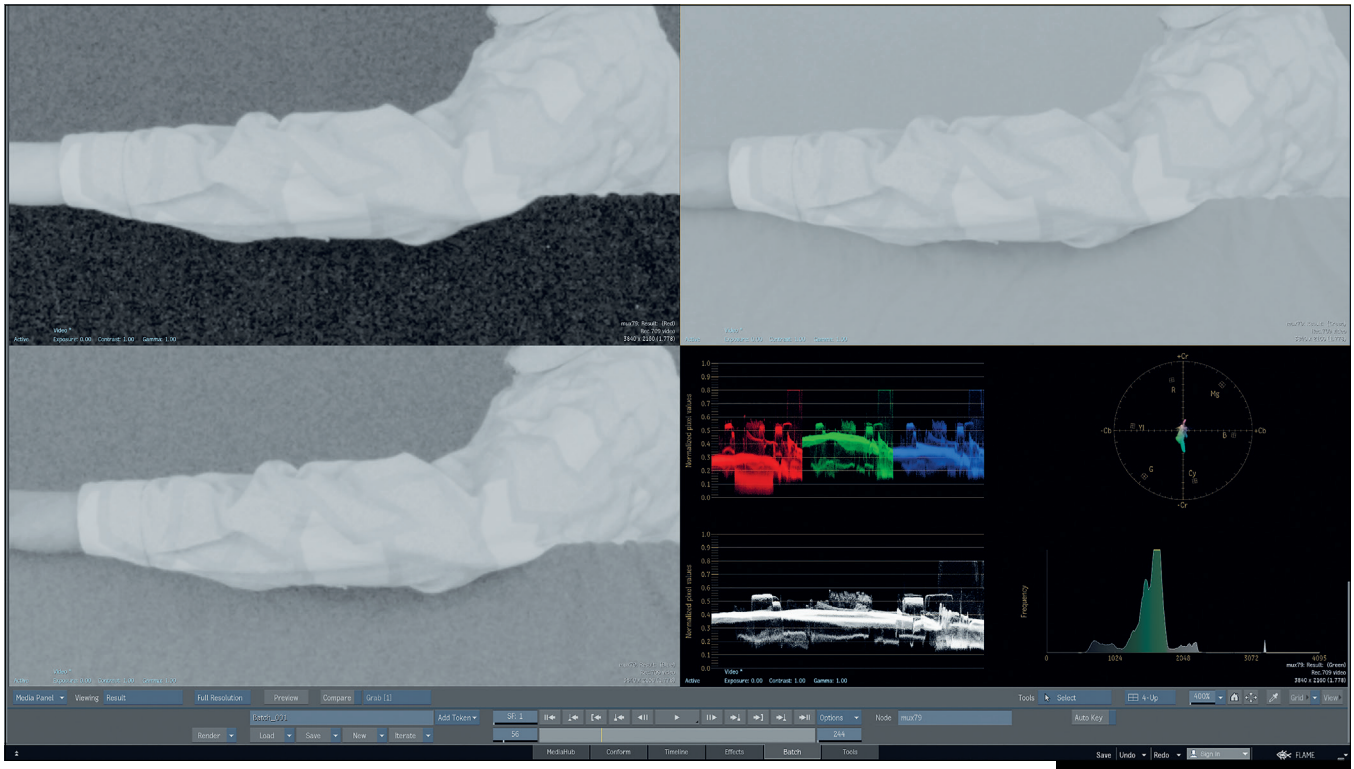
Game Engines
Realtime und Interactive – aber welche taugt für was?

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

animago 2018!
Alle Gewinner des diesjährigen Awards!

4 194336 217907 06

Copyright des Footages diesen Artikels: Finn Jäger / www.finnjaeger.de



In der Verteilung des Noises kommt der grüne Farbkanal eindeutig am besten weg.

Keying in Autodesk Flame

Green- oder Bluescreen-Keying gehören zum täglichen Brot eines jeden Compositing Artists. Wie jedes Compositing-Tool auf dem Markt bietet auch Flame eine Vielzahl an Keying-Tools. In diesem Artikel wollen wir einen Blick in die Keying-Toolbox von Flame werfen, wobei der Fokus auf den Tools und Features liegen soll, in denen sich Flame von den Mitbewerbern unterscheidet. Und da nichts über ein paar praktische Beispiele geht, werden wir uns hier einer Plate aus dem Musikvideo „Felsenfest" von John Known widmen. von Christoph Zapletal

Bevor es ans Keyen geht, braucht es eine gewisse Analyse und Vorbereitung der Plates. Sämtliche Plates wurden auf einer Blackmagic Ursa Mini Pro in UHD gedreht, in Resolve einem technischen Grade unterzogen und in Rec. 709 exportiert. Die verwendete Linse zeigt kaum Vignettierung, was uns einen schönen, gleichmäßigen Greenscreen liefert. Allerdings haben wir mit dem typischen Noise zu kämpfen, welches jeder digitalen Aufzeichnung heutzutage beiwohnt. Da beim in der Ursa verwendeten Bayer Pattern die grünen Pixel doppelt so oft vertreten sind wie rot oder blau, haben wir – passend zu unserem Greenscreen – das geringste Noise im grünen Kanal. Nichtsdestotrotz sollten wir unsere Plate vor dem Keyen einem vernünftigen Denoising unterziehen. Der hauseigene Denoiser in Flame kann diesen Job durchaus machen, allerdings alles andere als verlustfrei. Während er einen guten Job dabei macht, die Kanten nicht aufzuweichen, geht innerhalb der kurz rasierten Haare des Künstlers oder seines fein gemusterten Hemdes leider einiges an Detail verloren. Neat Video hingegen macht hier einen sehr guten Job, lässt uns die Details behalten und erzeugt eine sauber entrauschte Plate. Das einzige Problem: Neat Video läuft nicht nativ in Flame. Allerdings gibt es einen von Lewis Saunders geschriebenen OFX-Wrapper, mit dem man eine OFX-Lizenz von Neat Video in Flame lauffähig bekommt. Das bessere Ergebnis ist den Aufwand auf jeden Fall wert.

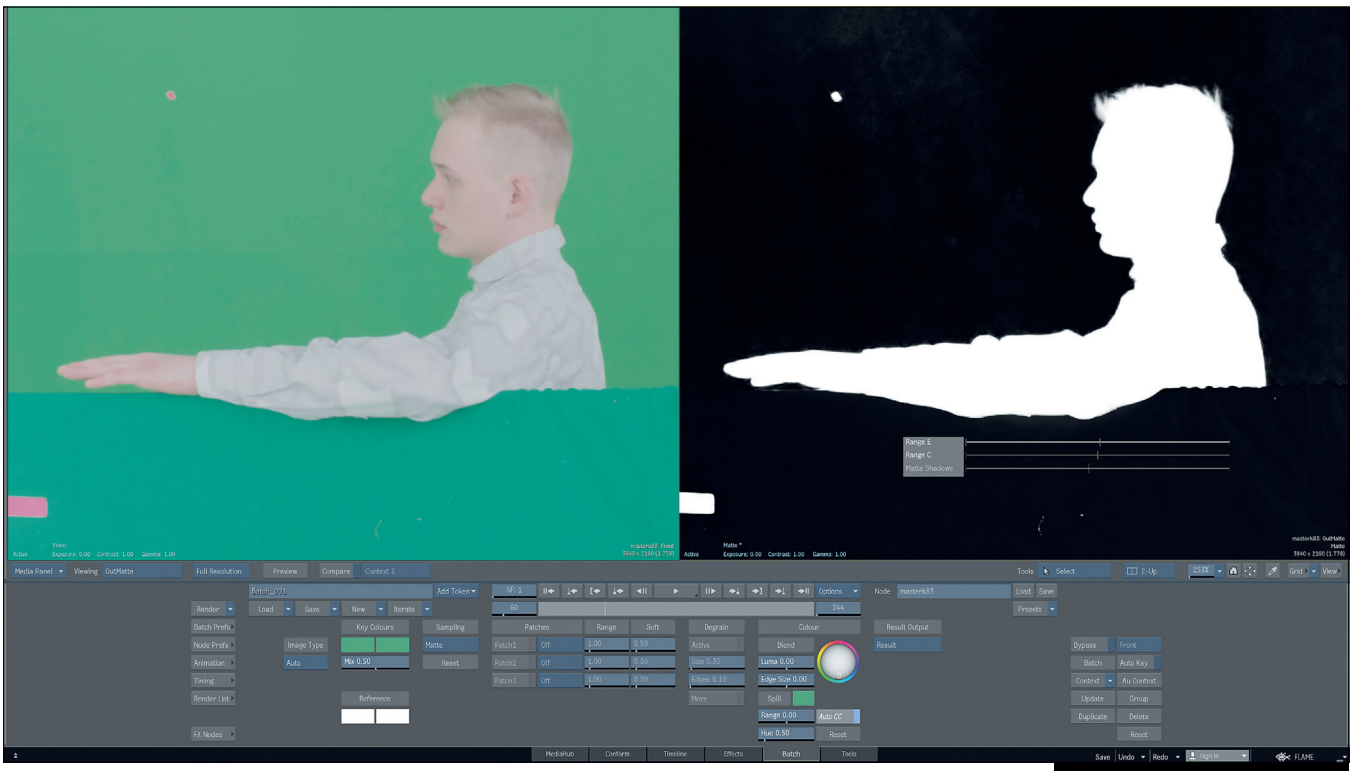
Keyer-Analyse

Egal in welcher App man sich befindet, eine kurze Trial-and-Error-Session mit den gängigen Keyern der Software ist in jedem Fall empfehlenswert, bevor man sich gleich am Anfang in Details verliert, die bei einem anderen Ansatz vielleicht gar nicht nötig wären. Auch kann das kurze Rumspielen mit den unterschiedlichen Algorithmen einen auf neue Ideen bringen, wie man den Key umsetzen möchte. Schon nach wenigen Minuten bekommt man ziemlich schnell ein Gefühl dafür, welcher Ansatz lohnenswert aussieht und auch welche Teilbereiche eines Bildes sich mit welchem Keyer umsetzen lassen. Gerade bei Haaren zum Beispiel ist es ja oft nötig, den letztendlichen Key aus mehreren unterschiedlichen Keyern zusammenzusetzen.

Natürlich verfügt auch Flame über all die üblichen Verdächtigen, die wir aus Nuke, Resolve, After Effects und anderen Programmen kennen. Channel, Luma, Chroma und Difference Key, alle sind sie da und verhalten sich so, wie wir es erwarten. Interessanter sind für diesen Artikel jedoch die Keyer, die sich ausschließlich in Flame finden. Allen voran sind das der Master Keyer, der 3D-Keyer und der GMask Tracer. Mit diesen Dreien wollen wir starten, denn sie werden in vielen Fällen die Go-to-Nodes für Keying in Flame sein.

Master Keyer

Beginnen wir mit dem Master Keyer. Dieser Node ist wirklich als One-Stop-Shop entworfen. So lässt sich auf Wunsch schon



Der Master Keyer mit seiner kontextsensitiven On-Screen UI

direkt im Keyer auch der BG reincompen, die Spill Suppression ist auch schon eingebaut. Das Prozedere, wie der Key erstellt wird, erinnert derweil ein wenig an Primatte. Eine Background-Farbe wird gesampelt und damit wird dann ein Basis-Key erstellt. Dieser kann jetzt über On-Screen-Slider verfeinert werden. Sehr angenehm ist, dass Flame immer nur die Slider anzeigt, die auch tatsächlich einen Einfluss auf die Pixel unter dem Cursor haben. Das gestaltet die Bedienung sehr intuitiv: Man klickt mit gedrückter Alt-Taste auf genau den Bereich der Maske, der noch optimierungsbedürftig ist, und erhält dann die für diesen Bereich relevanten Slider. Flame zeigt dem Artist sogar durch die Helligkeit der Slider an, wie effektiv ein Slider in einem gewissen Bereich ist. Anders als bei Primatte muss beim Keyen keine strenge Reihenfolge eingehalten werden: Ob man sich zuerst Foreground, Background oder Softness widmet, spielt für den Keyer keine Rolle. Das Refinement kann sowohl im Matte- als auch im Result-View erfolgen.

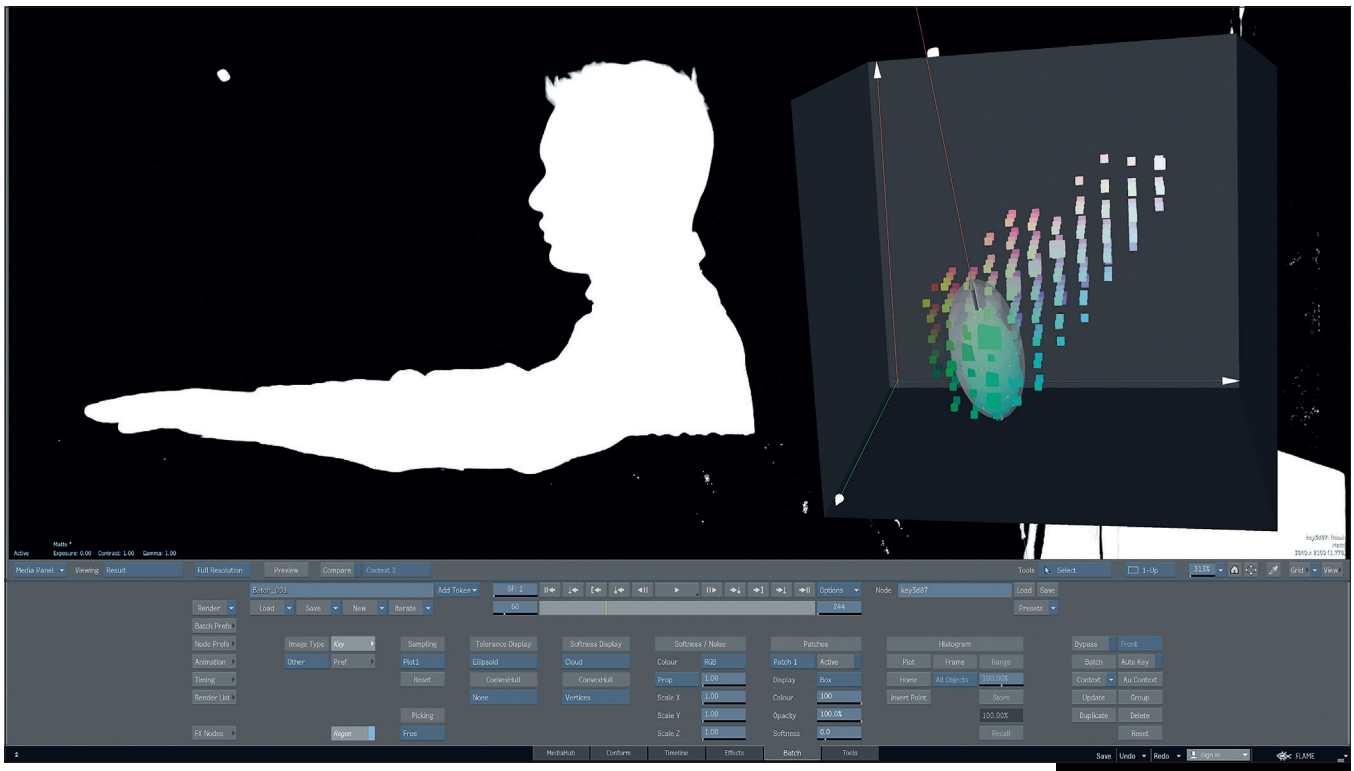
Ein nettes Feature des Master Keyers sei noch erwähnt: Für die Sampling-Farbe verfügt der Master Keyer über zwei Color Pots. So kann man unterschiedliche Grüntöne sampeln – zum Beispiel bei einem ungleichmäßig geleuchteten Greenscreen – und mit dem darunterliegenden Mix-Slider die optimale Mischung finden. Darüber hinaus bietet der Master Keyer ein zugegebenermaßen sehr rudimentäres integriertes Degraining-Tool und drei Patches. Patches sind lokale Keys, mit denen einzelne Fehler in der Maske oder an Edges korrigiert werden können, ohne dass der gesamte Key verändert wird. Allerdings sollte man mit Patches sehr vorsichtig umgehen, denn sehr schnell führen sie zu einer zu harten Maske oder unschönen Kantenartefakten. Sehr schön ist die eingangs erwähnte Spill Suppression des Master Keyers. Sie basiert auf der eingangs gesampelten Key Color und kann mittels derselben On-Screen-Controls sehr fein eingestellt werden. Auch kann anstatt eines Comps einfach nur die so farbkorrigierte Plate ausgegeben werden, weswegen es durchaus hilfreich sein kann, den Master Keyer für die Spill Suppression zu nehmen, selbst wenn man den Key mit einem oder mehreren anderen Keyern erstellt.

Wie schon gesagt, der Master Keyer ist als „One-Stop"-Lösung konzipiert, der zugrunde liegende Algorithmus ist der 3D-Keyer. Diesen gibt es in Flame auch als eigenständigen Node mit einem gänzlich anderen Bedienkonzept. Und obwohl der zugrunde liegende Code gleich ist, lohnt es sich oft, bei schwierigen Keys beide Nodes auszuprobieren, welche einem das bessere Ergebnis liefert, da die Bedienung so unterschiedlich ist.

Schon der Node selbst unterscheidet sich grundsätzlich vom Master Keyer. Hier gibt es nur einen Input und einen Output, mithilfe des 3D-Keyers wird wirklich nur eine Matte erstellt, für alles andere müssen wir hier auf die anderen Nodes in Batch zurückgreifen. Aber was ist überhaupt ein 3D-Keyer? Nun, Herzstück des 3D-Keyers ist ein Histogramm mit X-, Y- und Z-Achse, wobei Blau der X-Achse, Rot der Y-Achse und Grün der Z-Achse zugewiesen wird. In diesem Koordinatensystem wird die Farbverteilung im Bild dargestellt, das heißt, je mehr Pixel mit einem gewissen Farbwert im Bild vorhanden sind, desto größer wird dieser Bereich in der Point Cloud dargestellt. In diesem Histogramm können wir uns dank Shortcuts auch frei bewegen, also auch Rotieren und Skalieren. Sampeln wir nun das Grün unseres Beispielclips, kriegen wir, anders als beim Master Keyer, keinen One-Shot-Key, der schon auf halben Wege zu einem guten Ergebnis ist, sondern wir müssen uns Stück für Stück durch das Bild klicken. Und während wir also mehr und mehr Farbbereiche definieren, die von der Maske ausgeschlossen werden sollen, wächst im Histogramm eine Blase um den Teil der Cloud, der gekeyt wird. Unser Ergebnis jetzt wird sehr hart sein.

Sample the Softness!

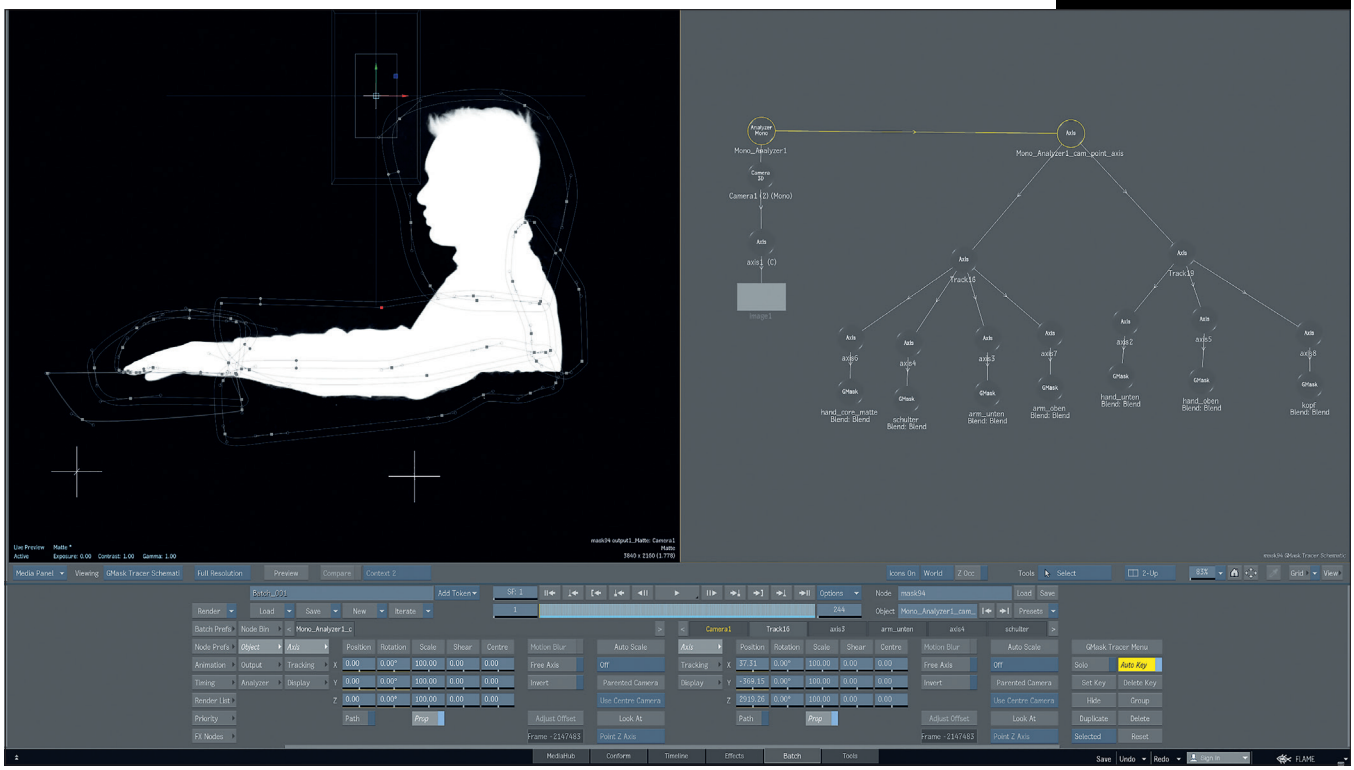
Der nächste Schritt ist das Sampeln der Softness, also all der Bereiche, wo wir feine Übergänge zwischen Background und Foreground haben. Auch dieses Sampling wird in unserem Histogramm dargestellt, wobei wir, ebenso wie beim Sampeln der Toleranz, die Darstellung innerhalb des Koordinatensystems individuell anpassen können. So kann man sich den entsprechenden Bereich als Ellipsoid darstellen lassen oder als Cloud oder auch die einzelnen Vertices. Wenn man so möchte, kann man jetzt anfangen, sich

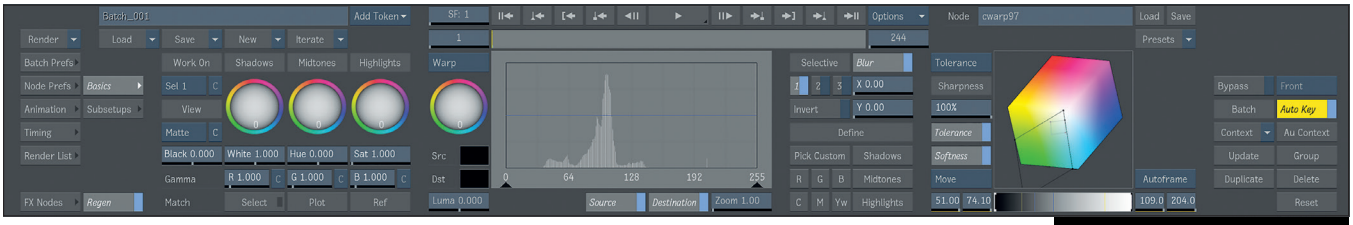


Die Ellipsen im Histogramm visualisieren unseren Key.

seinen Key mit den verschiedenen Parametern zu modeln. Es mag anfangs ein bisschen abstrakt sein, sich vorzustellen, was passiert, wenn man zum Beispiel die Softness auf der Y-Achse (also in Richtung des roten Farbbereichs) skaliert, aber mit ein bisschen Übung kommt man schon dahinter, und die Möglichkeit dieser sehr feinen Kontrolle und der Darstellung im Koordinatensystem können bei schweren Keys wirklich sehr hilfreich sein. Auch der 3D-Keyer verfügt über die drei Patches, die auch der Master Keyer besitzt. Dankeswerterweise werden auch sie in Form von kleinen roten Boxen im Koordinatensystem dargestellt, doch ihre Verwendung ist ähnlich riskant wie beim Master Keyer. Bevor man sich hier einen Key, der schon auf einem guten Weg ist, durch achtlos eingesetzte Patches verhaut, sollte man vielleicht schauen, ob man die Probleme nicht vielleicht mit einer Core oder Garbage Matte im Batch Flowgraph lösen kann.

Localized Keying im GMask Tracer





Der Color Warper lässt sich auch wunderbar zum Keyen nutzen.

GMask Tracer

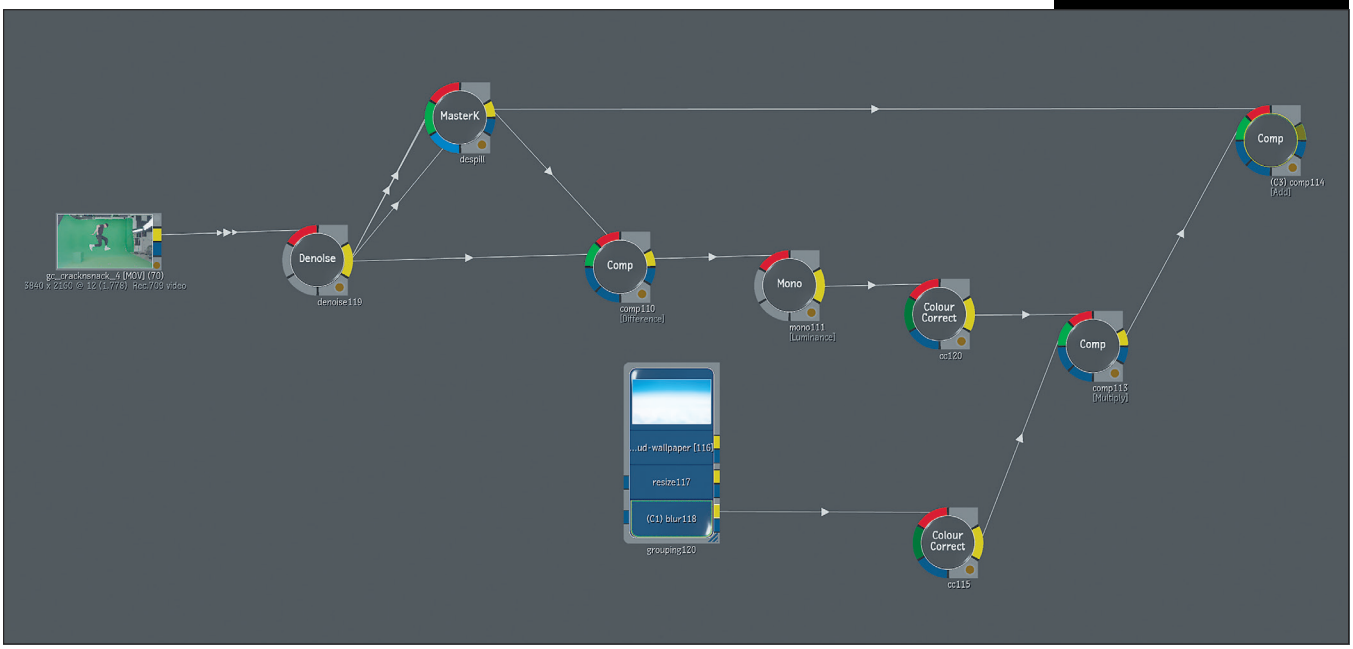
Der dritte große Keyer, den wir uns genauer anschauen wollen, ist der GMask Tracer beziehungsweise die Tracer-Funktion. Diesen Keyer als gut versteckt zu bezeichnen, ist wahrscheinlich noch eine Untertreibung. Der GMask Tracer ist – neben der ursprünglichen, aber nicht mehr aktiv weiterentwickelten GMask – das Rotomasking Tool in Flame. Um nun auf die Keyer-Funktion in diesem Tool sinnvoll eingehen zu können, müssen wir einen kleinen Exkurs in das Roto-Tool selbst unternehmen. Der GMask Tracer basiert auf den GMasks des Action-3D-Compositors, ist allerdings in seiner Funktionalität auf Roto und eben – wenn auch ein bisschen versteckt – auf Keying zugeschnitten.

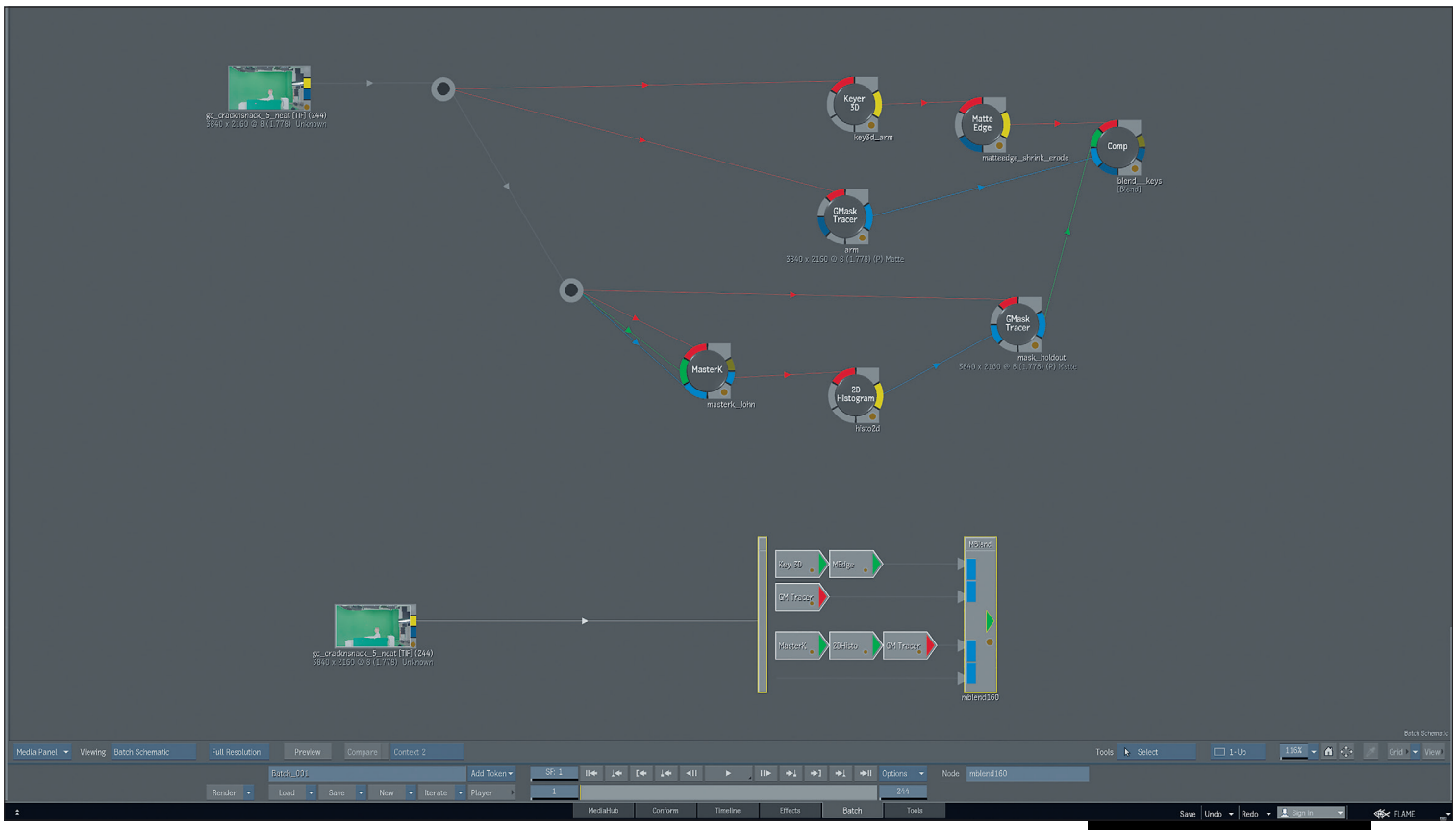
Mit der Verwandschaft zu Action geht auch einher, dass der GMask Tracer über eine eigene schematische Darstellung verfügt und Trackingdaten von Point- und Planar-Trackern, aber auch zum Beispiel Kamera-Tracks leicht mit beliebig vielen Roto-Shapes verbunden werden können. Diese Daten können natürlich auch von externer Software kommen, zum Beispiel von Mocha, Syntheyes oder PFTrack. Die Roto-Shapes verfügen über all die Funktionen, die man erwartet und auch aus anderen Tools kennt. Interessant ist hier allerdings Folgendes: Jede einzelne Roto-Shape verfügt über einen individuell einstellbaren Keyer, den eingangs erwähnten Tracer.

Das Interface ist denkbar simpel: Im Tracer-Menü kann man die „Analysis" dazuschalten, woraufhin in der Object View der jeweiligen Roto-Shape zwei Sampling-Boxen erscheinen, eine grüne und eine rote. Die grüne definiert den Background, die rote den Vordergrund. Das Sampling findet auf dem jeweils aktiven Frame statt. Bewegen wir uns auf der Timeline, verschwinden auch die Sampling-Boxen. Somit läuft man nicht Gefahr, dass bewegte Objekte den Key verhauen. Man kann aber auf einem beliebigen anderen Frame weitere Sampling-Boxen setzen, auch mehrere Sampling-Boxen auf demselben Frame sind kein Problem. Und jeder Keyframe ist über die Buttons „Previous" und „Next" schnell ansteuerbar. Ebenso sind die Position und Größe frei anpassbar. In der Object View sehen wir die Auswirkungen dieses indivuellen Samplings auf den ganzen Frame angewendet, schalten wir aber zurück in die Result View, ist der Tracer Key durch die ihm zugrunde liegende Roto-Shape begrenzt.

So können wir jetzt Stück für Stück mit relativ groben Roto-Shapes unseren Key aufbauen. Gerade in unserem Beispiel, wo der grüne Stoff, mit dem der Künstler abgehängt wurde, einen anderen Farbton hat als die grüne Wand hinter ihm, ist diese Arbeitsweise echt praktisch. Denn anders als beim klassischen Keying, wo man ja als Erstes versucht, so viel vom Bild wie möglich mit einem Key abzudecken, bricht man sich das Bild durch die einzelnen Roto-Shapes von vornherein runter. Autodesk nennt diesen Workflow Localized Keying. Insbesondere bei Footage, welches eigentlich nicht zum Keyen gedacht war, ist der GMask Tracer eine echte Bereicherung, vor allem weil man wie auch in unserem Beispiel Trackings, die ohnehin schon bestehen, wie hier für den Background, ohne größere Probleme mit verwenden kann. Jede Roto-Shape verfügt darüber hinaus über individuelle Post-Processing-Optionen wie Blur, Shrink und Dilate und die unterschiedlichen Keys in den jeweiligen Shapes können auch über Blend Modes wie Add, Multiply, Difference, Exclusion und einige mehr miteinander verrechnet werden.

All diese Operationen fasst Crok Additive Key übersichtlich in einem Shader zusammen.





Der gleiche Key – einmal klassisch mit Batch Nodes zusammengesetzt und einmal im Matte Blend Tool

Nachdem wir uns in unserer kleinen Keyer-Analyse nun eingehend mit den drei großen Keyern beschäftigt haben, wollen wir unser Augenmerk noch auf drei weitere, vielleicht nicht ganz so offensichtliche Keyer in Flame lenken. Zuerst wäre da der im Color Warper integrierte Diamond Keyer. Eigentlich ist dieser Keyer für das Definieren der drei selektiven Farbkorrekturen im Color Warper gedacht. Da es aber möglich ist, die Matte dieses Keyers anstatt des farbkorrigierten Bildes auf den Output zu legen, kann man so diesen Node auch ziemlich schnell zum Keyen zweckentfremden. Die Vorgehensweise ähnelt hier dem 3D-Keyer, basiert allerdings auf einem anderen Algorithmus und ist auch etwas simpler im Umgang. Anstatt mit einem 3D-Histogramm wie im 3D-Keyer arbeiten wir hier mit einem zweidimensional dargestellten Farbraum. Nachdem wir den Hintergrund gesampelt haben, erscheinen zwei Trapeze, mit denen Toleranz und Softness definiert werden können. Da man aus dem Color Warper allerdings nur Matte oder Foreground ausgeben kann und dieses Tool eigentlich nicht für das Keyen ausgelegt ist, müssen sämtliche Anpassungen der Maske extern im Batch Flowgraph erfolgen. Nichtsdestotrotz ist der Color Warper eine Option, die es wert ist, getestet zu werden.

Und dann gibt es da noch zwei andere Keyer, die man eigentlich nicht in Flame erwartet, sondern die ihre Anleihen tatsächlich aus anderen Compositing-Anwendungen erhalten haben. Dass sie ihren Weg in Flame gefunden haben, ist der Matchbox-Schnittstelle zu verdanken, über die Entwickler und auch User GLSL-Shader skripten können. Hier gibt es mittlerweile eine Vielzahl an kostenlosen und auch kostenpflichtigen Shadern. Unter **https://logik-matchbook.org** findet man eine wunderbar aufbereitete Sammlung der kostenfreien Shader. Hier finden wir auch die beiden Keyer, die es noch zu betrachten gilt. Das erste wäre Crok_IBK, der, wie der Name schon vermuten lässt, seine Anleihen bei Nukes Image Based Keyer nimmt.

Das Prozedere entspricht in etwa auch der Arbeit in Nuke. Man kann entweder eine gedrehte Clean Plate als zusätzlichen Input in den Shader pipen oder aber Crok_IBK erstellt intern die Clean Plate. Diverse Werte wie zum Beispiel die Gewichtung der einzelnen Farbkanäle, eine Sampling Color und Ähnliches können eingestellt werden. Leider bietet die Matchbox-Schnittstelle einige Limitierungen, die die Arbeit mit dem Keyer etwas müßig machen. So gibt es leider keine On-Screen-Controls und die Arbeit geht insgesamt nicht so flüssig von der Hand wie bei den nativen Keyern in Flame oder dem entsprechenden Vorbild in Nuke.

Crok Additive Keyer

Der zweite Matchbox Shader, den wir uns anschauen wollen, ist der Crok Additive Keyer, wobei dieser kein Keyer im eigentlichen Sinne ist, sondern eine Aneinanderreihung von Blending-Operationen. Mit ihm kann man Dinge wie feines Haar-Detail oder Motion Blur besonders gut vor einen Background setzen, eben die Dinge, die sonst schnell bei einem Key in Mitleidenschaft gezogen werden. Compositing-Veteranen werden dieses Feature noch aus Quantels Henry und Editbox kennen.

Da es in Flame kein entsprechendes Feature gab, haben sich Artists dieselben mathematischen Operationen mit mehreren Nodes nachgebaut. Ivar Beer aus Berlin hat diesen Additive Keyer nun als Matchbox Shader geskriptet und dadurch deutlich alltagstauglicher gemacht. Der Additive Keyer eignet sich vor allem beim Bearbeiten von Clips mit viel Motion Blur, feinem Detail wie zum Beispiel Haaren oder Fokus-Pulls vor einem Greenscreen.

Durch das Differenzieren von Originalmaterial und dem gedespillten Footage und anschließendem Addieren auf den Background kann man fast alle Details beibehalten, erhält allerdings fast immer einen leichten Film oder Schleier über dem Comp. Diesem muss man nun mit gezielter Farbkorrektur von Vorder- und Hintergrund entgegenwirken.

Wie das als Aufbau mit den handelsüblichen Flame-Nodes aussieht, kann man auf dem folgenden Screenshot sehen. Der Matchbox Shader fasst all diese Funktionen in einem Node zusammen.

Anwendung

Auch wenn es verführerisch ist, den bereits erstellten 3D-Track und den GMask Tracer zu benutzen, wäre es am Ende ein absoluter Overkill. Beim Testen liefert der Master Keyer schon ein wirklich anständiges Ergebnis und erhält uns selbst die feinen Details der Haare. Einzig am Arm kriegen wir leichte Edge Issues, weswegen wir den Arm mit einem separaten 3D-Keyer freistellen und dann hinterher mit unserem Main Key kombinieren.

Kleine Helferlein

Wie schon eingangs erwähnt, macht ein Keyer noch keinen Key. Oft braucht es mehrere Keys und einiges an Post-Processing-Operationen, um den Key wirklich zum Ziel zu bringen. Neben den üblichen Nodes wie Blur, 2D-Histogramm, Comp etc. gibt es aber noch den einen oder anderen Node, der einem dabei helfen kann.

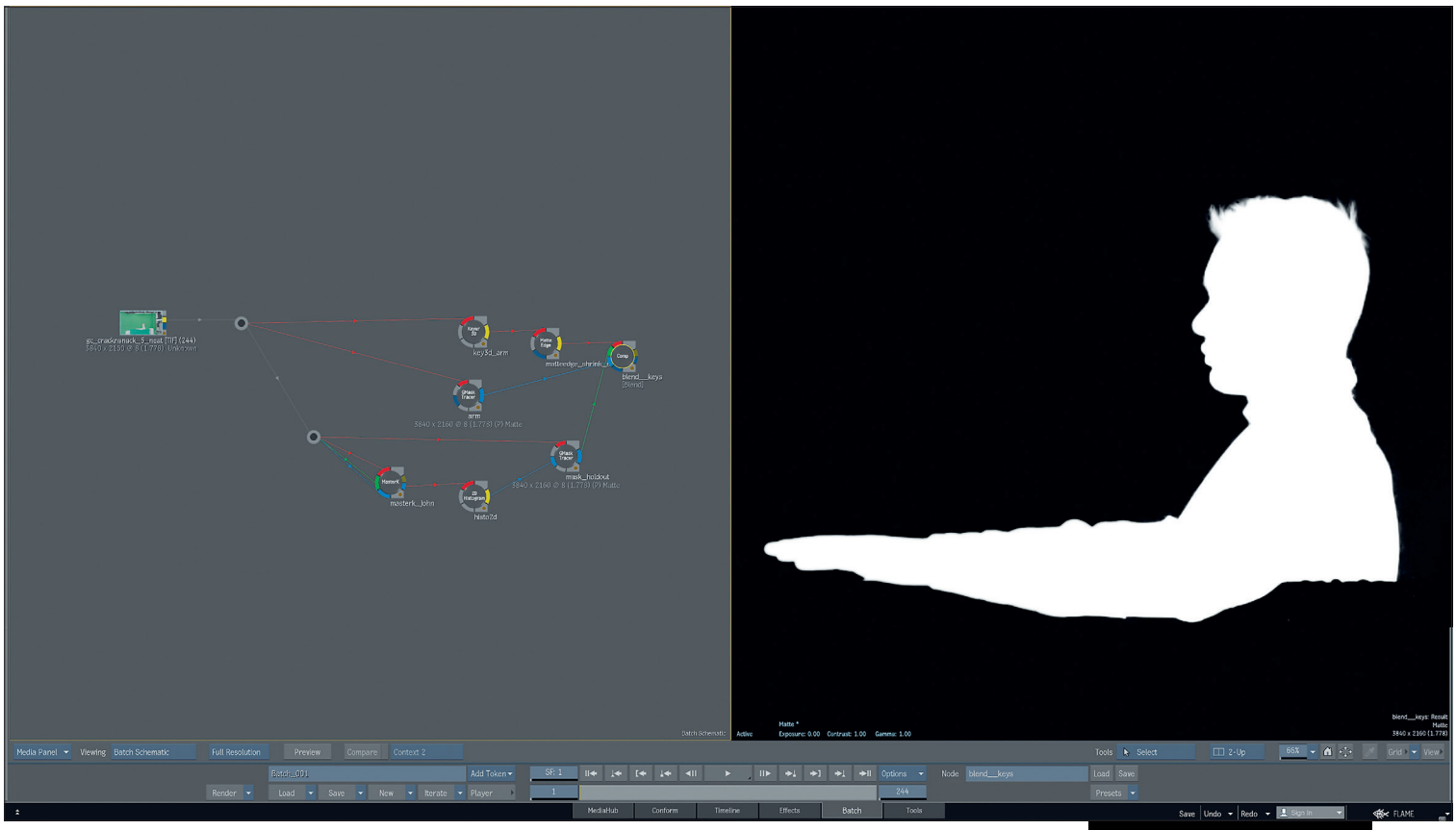
Matte Blend Tool

Das Matte Blend Tool hat seinen Ursprung im Modular Keyer von Flame. Der Modular Keyer ist eine Art Mini-Batch innerhalb des gleichnamigen Nodes oder eines Action Layers. Allerdings hat dieses Modul – bedingt durch seine Integration in Action – einige Stolpersteine, weswegen mehr und mehr Artists ihre Keys lieber in ihrem Batch Flowgraph erzeugen, als dafür einen dedizierten Modular Keyer zu verwenden. Allerdings war das Matte Blend Tool lange nur exklusiv im Modular Keyer zu finden. Autodesk hat aber dem veränderten User-Verhalten Rechnung getragen und das Matte Blend Tool auch im regulären Batch Toolset – ebenso wie das ihm verwandte Color Blend Tool – verfügbar gemacht.

Das Matte Blend Tool ist – wie der Name schon sagt – darauf ausgelegt, verschiedene Keys und Masken zu einer Matte zu verblenden. Hierzu stellt das Tool bis zu vier Pipelines zur Verfügung, in denen Nodes hintereinandergelegt werden können. Jede Pipeline verfügt zusätzlich über einen Kanal für Holdout-Masken, um einen Key zusätzlich zu begrenzen. Am Ende hat man noch – ähnlich wie bei Nukes Keymix-Node – Blending Curves, um die unterschiedlichen Matte Layer optimal zu verblenden. So kann man seinen Key sehr ordentlich und kompakt zusammenfassen.

Allerdings birgt das Ganze auch einige Nachteile, die nicht unerwähnt bleiben sollen: So ist man, anders als beim regulären Zusammensetzen des Keys im Node Tree, auf vier Layer beschränkt. Individuell vergebene Namen der Nodes werden, wenn sie in die Pipeline geworfen sind, nicht mehr angezeigt, und man kann nicht mehr das Ergebnis eines Keys, zum Beispiel bevor er geshrinkt oder von einer GMask begrenzt wird, aus der Pipeline zur weiteren Verwen-

Anzeige



Unsere finale Matte mit minimalem Post-Processing

dung im Comp herauspipen. Darüber hinaus muss man sagen, dass all die Funktionen, die das Matte Blend Tool bietet, auch mit anderen Nodes erreicht werden können. Wer zum Beispiel die Key-Mix-Funkionalität aus Nuke vermisst, kann auf den Matte Curves Node in Flame zurückgreifen. Letzten Endes ist das Matte Blend Tool ein Workflow-Tool, und jeder Artist muss für sich selber entscheiden, ob er diesen Workflow nutzen möchte.

Funktionen wie Edges, Erode, Shrink und Ähnliches, die in anderen Apps wie zum Beispiel Nuke in einzelnen Nodes verteilt sind, hat Autodesk im Matte Edge Node zusammengefasst. Die Tools verhalten sich, wie man es erwartet und aus anderen Applikationen kennt. Das Zusammenfassen der Tools macht aber durchaus Sinn, da man schnell die Auswirkungen der verschiedenen Operationen ausprobieren kann, ohne ständig im Flowgraph hin- und herspringen zu müssen.

Pixel Spread

Zu guter Letzt sei dann noch der Pixel Spread erwähnt. Pixel Spread benötigt als Input den gegebenenfalls gedespillten Vordergrund und den Key und führt dann intern mehrere Blur- und Divide-Operationen aus und erreicht somit einen Edge Extend. Das heißt, der Farbwert der Pixel innerhalb der Maske wird auf die benachbarten Pixel erweitert. Somit können effektiv Halos und Kanten bei Keys bearbeitet werden. Allerdings ist Pixel Spread mit Vorsicht zu genießen. Denn sobald die Pixel zu stark extendet werden, entstehen Warping-Effekte und Artefakte, die spätestens wenn der Darsteller zum Beispiel seinen Kopf dreht, schnell auffallen können. Pixel Spread kann bei schwierigen Keys echt weiterhelfen, aber solange man die Maske noch verfeinern kann, sollte man sich eher darauf konzentrieren, als den Vordergrund über die Maßen zu manipulieren.

Natürlich gibt es noch zahlreiche weitere Nodes und Matchbox Shader, die einem für das Erreichen eines guten Keys zur Verfügung stehen, und gerade durch die Matchbox Community kommen ständig neue dazu. Die Auswahl hier sollte lediglich die hervorheben, die entweder exklusiv in Flame sind oder sich grundsätzlich von anderen Applikationen unterscheiden.

Für unseren Shot kommen wir zum Glück mit sehr wenig Post-Processing aus. Der Main Key wird minimal mit dem 2D-Histogramm enger gezogen und unser 3D-Keyer für den Arm wird mit homöopathisch eingesetztem Shrink und Erode aus dem Matte Edge Node verfeinert.

Fazit

Flame bietet mit dem Master Keyer, dem 3D-Keyer und dem GMask Tracer drei sehr mächtige hauseigene Keying Tools, die das Set an Standard-Keyern sinnvoll ergänzen. Gerade der Master Keyer ist mit seinen On-Screen-Controls auf Geschwindigkeit ausgelegt und passt somit wunderbar in die ganze Bedienungsphilosophie von Flame. Allerdings sucht man gewisse Industriestandards wie zum Beispiel Keylight, Ultimatte oder Primatte vergeblich. Weder über die Sparks-Schnittstelle noch durch Matchbox-Shader sind diese Tools in der Vergangenheit in Flame verfügbar gemacht worden. Vielleicht wird dies aber über die konsequente Weiterentwicklung, die Matchbox erfahren hat, in Zukunft möglich. Die ersten Ansätze, die wir uns mit Crok_IBK ja in diesem Artikel anschauen konnten, sehen auf jeden Fall vielversprechend aus. Und selbst mit dieser Lücke deckt das Toolset in Flame alles ab, um selbst mit anspruchsvollen Keys fertig zu werden. **› ei**

Christoph Zapletal ist seit mehreren Jahren als freiberuflicher Compositing Artist in Hamburg tätig. Sowohl mit Autodesk Flame als auch The Foundrys Nuke arbeitete er bereits an diversen Werbe- und Spielfilmprojekten mit. Darüber hinaus ist er als Dozent für fxphd und das iSFF (Institut für Schauspiel, Film- und Fernsehberufe) in Berlin tätig.
www.christophzapletal.de

CUT ABOVE THE REST

There's a Media Composer for everyone

Whether you're just starting out, editing to a tight budget, or part of a team building the next Hollywood blockbuster, the **Media Composer**® product family has a solution to fit your needs.

Get started with **Media Composer | First** and use the tools the pros use for FREE. Step up to **Media Composer** for just $19.99 per month* for more power and versatility. And for real-time collaboration and game-changing features like ScriptSync® and PhraseFind™, advance to **Media Composer | Ultimate**.

Find out more at: **avid.com/media-composer**

YOUR PLATFORM
TO **CREATE**

© 2018 Avid Technology, Inc. All rights reserved. Avid, the Avid logo, Media Composer, PhraseFind, and ScriptSync are either registered trademarks or trademarks of Avid Technology, Inc. in the United States and/or other countries.
*All prices are USMSRP for the U.S. only and are subject to change without notice. Contact your local Avid office or reseller for prices outside the U.S.