



# Scratch-Workflow

Seit November sind Scratch und sein kleiner On-Set-Bruder Scratch Lab in der Version 7 für Windows und OSX verfügbar – Zeit, beide Tools und auch die Workflows, die sich aus der Verwendung ergeben, genauer unter die Lupe zu nehmen. In erster Linie ist Assimilate Scratch ein Grading Tool, doch es hat noch mehr auf Lager, viel mehr. Was also unterscheidet es von den Konkurrenten wie Baselight, Lustre, Nucoda und Co.?

von Matthias Aderhold

Zuallererst ist es Flexibilität – die Flexibilität, aus einem Grading Tool eine ganze Pipeline zu machen. Dazu muss man verstehen, wie Scratch funktioniert und das ist, wenn man hinter den Aufbau aus CONstruct und Player gestiegen ist, nicht sehr schwer. Digitale Workflows bestimmen heute den Post-Alltag – rein für diese Art Pipeline ist Assimilate Scratch entwickelt worden und arbeitet dabei komplett auf der Grafikkarte – nicht in Teilen, nicht bestimmte Tools, sondern das ganze Programm ist von Anfang an auf Grafikkarten-Power getrimmt worden.

Damit ist Scratch sowohl auf Windows als auch auf OSX schon jetzt in der Lage, Timelines bis zu einer Größe von 16k zu verarbeiten – aktuelle Quadro-Karten lassen dies problemlos zu – kein Bedarf für eine zweite Grafikkarte. Nichtsdestotrotz gibt es einige Komponenten, wie zum Beispiel Quicktime-Export, oder



HP Z820 mit Quadro 6000, ioDrive2-Duo und zwei Bluefish444 Supernova Karten für 4K Output – noch ist Platz für FibreChannel-Controller oder RED Rocket.

das RED-SDK, die zwingend von CPU-Power profitieren, da nicht GPU-fähig. Gute Erfahrungen haben wir hier mit HP Z Workstations gemacht – aber auch Boxx-Computer und auch ein MacPro sind zuverlässige Systeme. Video-IO kann über eine Nvidia-SDI-Option, über die Kona-Karten von AJA, oder Bluefish 444 Epoch Karten (bis zu 4K RGB 12 bit 4:4:4) gelöst werden. Im Zweifelsfall kann auch der Dual-Head DVI-Ausgang der Grafikkarte herhalten. Das Storage kann der benötigten Bandbreite nach frei gewählt werden, sowohl Thunderbolt, als auch Fiber-Channel oder PCI-Express-Raid-Systeme spielen problemlos mit Scratch zusammen. Getestet haben wir in diesem Fall mit einem Silverstor SSD-Raid von JMR.

Man kann annähernd jedes aktuelle freie Panel mit Scratch betreiben – vom JL Cooper MC Spectrum, über Avid McColor bis zu Tangents Wave, CP-200 und Element Panels. Bei Verwendung der Element-Panels hat man mit Scratch sogar die Freiheit so viele Panels wie benötigt anzuhängen – wer also zu den bestehenden vier Panels noch ein Panel mit zusätzlichen Pan-Pots braucht, kann dieses auch dranhängen. Oder auch fünf. Unterm Strich sind damit die Hardware-Voraussetzungen in den meisten Post-Häusern schon gegeben.

### Die Software

Assimilate Scratch hat einen recht eigenwilligen Aufbau, wie jedes spezialisierte Tool. Um dahinter zu steigen, muss man für einen Moment mal alles über Bord werfen, was man von anderen Systemen kennt und sich auf die Scratch-Denke einlassen. Aber das ist weniger schmerzhaft, als man annehmen möchte. Startet man die Software, beginnt zunächst alles recht harmlos: Man hat in einem übersichtlich gestalteten Menü die Möglichkeit Projekte und User anzulegen oder auszuwählen. Daneben gibt es verschiedene Untermenüs für Project-, User-, und System-Settings.

Jedes Untermenü ist ähnlich übersichtlich strukturiert und man findet sich sofort zurecht. In den Projekt-Settings legt man in erster Linie das sogenannte Media-Directory für jedes Projekt fest. Man kann selbstverständlich von jeder beliebigen Location aus Footage importieren, jedoch macht es Sinn, alles in einen Ordner (dem Media-Directory) zu konsolidieren – zum Beispiel wenn das gesamte Projekt zwischen mehreren Workstations (oder Standorten) wechselt und sich damit der absolute Pfad zu jedem verwendeten File verändert. Öffnet man das Projekt also in einer anderen Grading Suite, gibt man lediglich den neuen Pfad zum altbekannten Media-Directory an und sowohl das Footage, als auch Lookup-Tables sind sofort relinked und online.



Scratch-Grading-System bestehend aus HP Z820, JMR Silverstor 8-bay SSD-Raid, Tangent Element Panel, Eizo CG245w und TV Logics 4K-Display LUM-560W

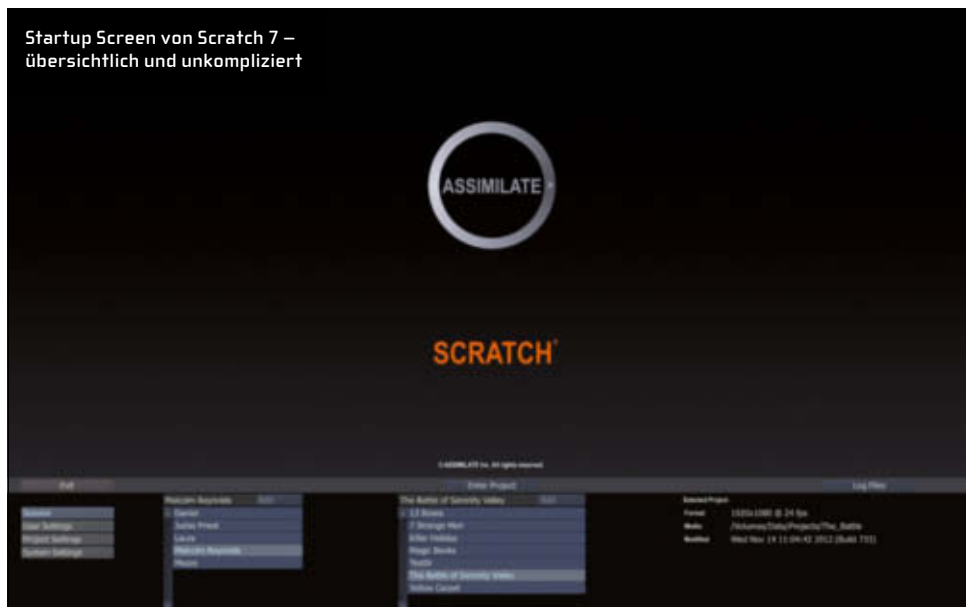
User-Settings beschränken sich hier im Wesentlichen auf Einstellungen wie Stiftempfindlichkeit, und Interface-Optionen. In den System-Settings konfiguriert man sowohl den SDI-Output, als auch die Panels und sogenannte „Custom Commands“. Custom Commands sind selbst erstellte Buttons, die mit einer beliebigen Funktion, wie etwa einem Plug-in, oder einem Script belegt werden können.

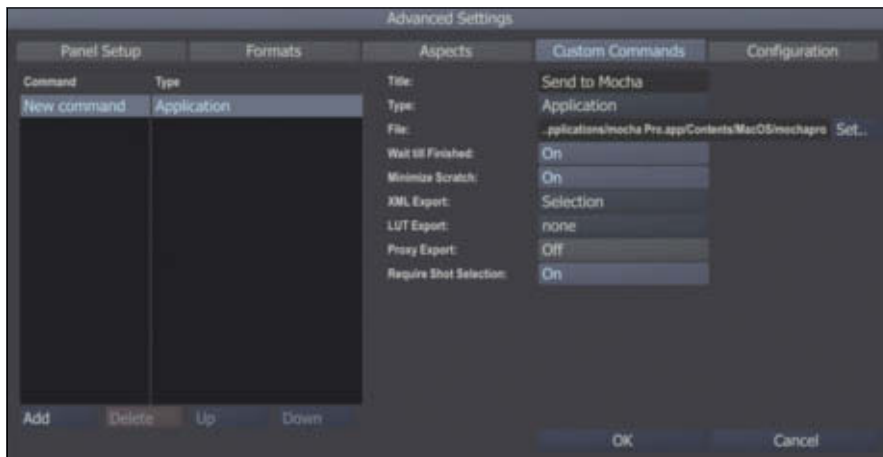
Damit ist es problemlos möglich, einen Shot per Knopfdruck an Nuke, After Effects, Mocha oder das Restaurierungs-Tool Dustbuster+ von HS-ART zu senden und genauso schnell wieder nach Scratch zurückzuholen: Der Shot wird auf Knopfdruck beispielsweise an Mocha weitergegeben, anspruchsvolles Tracking findet dort statt, und hinterher wird lediglich die Tracking-Information via XML nach Scratch zurücktransportiert. Der Verzicht auf Exporte und Importe spart hier doch

ein paar Klicks – und wer Scripting-Skills hat, kann sich mittels Python-Script Custom Commands für jeden Zweck programmieren – die dazu notwendige Dokumentation ist online auf der Support-Seite verfügbar.

### Der CONstruct

Öffnen wir also ein Projekt. Innerhalb eines Projektes splittet sich Scratch im Grunde genommen in zwei Ebenen auf: Den CONstruct und den Player. Zunächst befinden wir uns im CONstruct – hier finden Import / Export und Mediamanagement sowie Conforming statt. Dazu können wir uns mehrere CONstructs anlegen – CONstructs muss man sich vorstellen wie aufgebohrte Timelines – und dort erst einmal Footage importieren. Nativ unterstützt werden alle gängigen Kameraformate, wie ARRI-raw, RED, XDCAM-HD, Sony F65 / SR, Panasonic-P2, AVCHD, Canon MXF, Phan-





Custom Commands für Scratch 7 werden in den System-Settings konfiguriert.



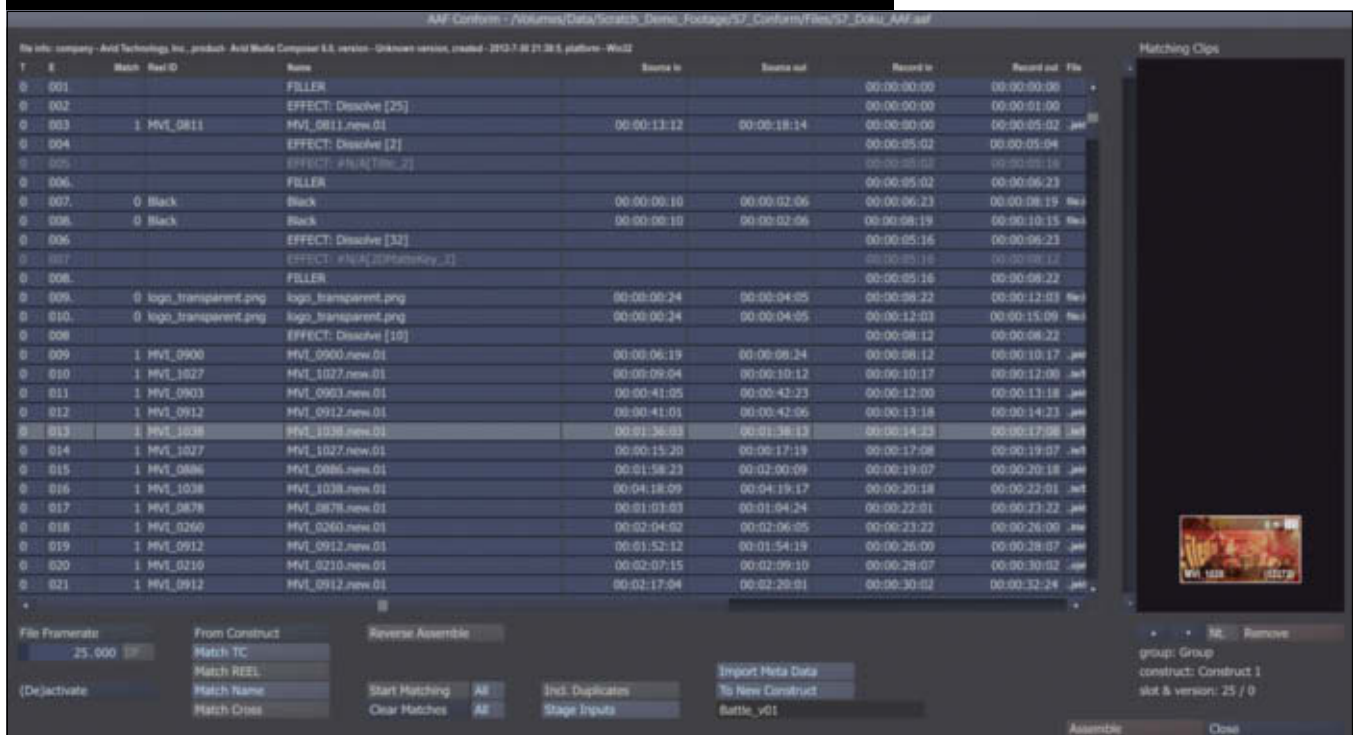
Die Palette an unterstützten Formaten ist von Haus aus breit, kann aber durch Drastics MediaReactor Plug-in noch auf Formate wie PSD, WMV und GXF und viele mehr erweitert werden.

tom Cine sowie diverse Bildsequenzformate und sämtliche auf Betriebssystemebene installierten Quicktime-Codecs (also h.264, ProRes, aber auch AJA, Avid, Blackmagic und Bluefish 444 Codecs).

Die Palette an unterstützten Formaten kann man mit Drastics MediaReactor Plug-in (auch für FCP und Adobe verfügbar) noch beliebig erweitern, wenn nötig.

Über Load shot, bzw. Load Layers (= multiple shots) importiert man nun Footage jeglicher Couleur in die erstellten CONstructs – anschließend hat man die Möglichkeit, die Metadaten der geladenen Clips zu verändern. Zu diesem Zweck existiert der Media Browser. Hier sind alle im Projekt verwendeten Clips samt Metadaten gelistet. Hier hat man die Möglichkeit die File-Interpretation für mehrere Clips auf einmal zu verändern: einerseits Dinge wie Aspect, Framerate und Reel-ID, andererseits hat man Zugriff auf De-bayering-Optionen bei Raw-Files oder kann Audio syncen – kommt dieses mit demselben Timecode wie die korrespondierenden Clips, reichen zwei Mausklicks, um alles Footage mit dem Ton zu verheiraten – hat der Sound Engineer am Set auch Scene- und Take-Information in seine Sound-Dateien eingetragen, werden diese automatisch als Metadaten an den jeweiligen Clip angehängt und auch beim Export berücksichtigt.

Hier wird deutlich, dass Assimilate Scratch unter der Haube wie eine Datenbank arbeitet – das ist einer der Hauptgründe für die Geschwindigkeit des Tools, besonders auf langsameren Rechnern und macht es auch möglich, innerhalb von fünf Sekunden die



Das Conform-Window lässt auch das Matchen einzelner Einträge einer EDL über die individuell ausgewählten Kriterien zu.

Software neu zu starten oder das Projekt zu wechseln.

Als Nächstes können wir mittels EDL, AAF oder XML eine neue Timeline conformen. Das Conforming-Window ist ebenfalls äußerst übersichtlich und leicht zu handhaben: Zunächst werden die Kriterien gewählt, nach denen gelinked werden soll, anschließend wird gematched – sowohl für alle Einträge in der Schnittliste, als auch für einzelne markierte. Ist dies geschehen, kann man die gefundenen Clips für jeden Eintrag durchsehen und bei mehreren Treffern für einen Eintrag die ungewollten löschen. Mit einem Klick auf Assemble hat man seine neue Timeline und kann anfangen zu graden.

### Der Player

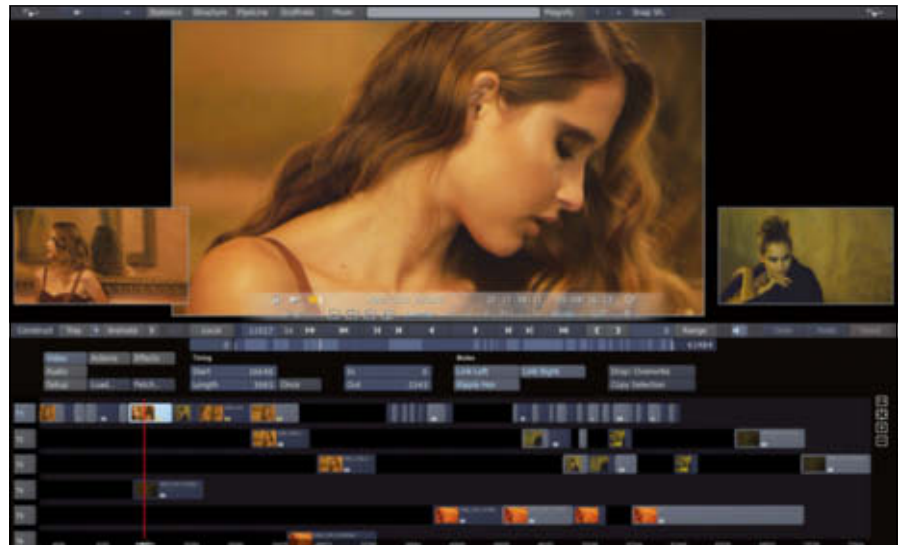
Durch den Playbutton in der Mitte betritt man nun den Player – quasi die bessere Hälfte von Scratch. Der Player unterteilt sich in vier verschiedene Untersektionen: Player, Edit, Matrix und Process. Alle Sektionen des Players treffen für den Clip auf dem man sich mit dem Playhead befindet parallel zu.

In der Tat ist die erste Untersektion, der Player, nicht mehr als ein Player – er wird vorrangig zum Playout auf eine MAZ genutzt. Im Process-Menü wird eingestellt, wie Scratch den vorhandenen Clip interpretieren soll – im Falle von RED oder ARRI-RAW hat man hier also Zugriff auf Debayering Settings, Farbraum, Farbtemperatur, etc. – dasselbe wie im Media-Browser, jedoch auf einer Shot-by-Shot-Basis.

Der Edit Mode ist, wie der Name schon vermuten lässt, zum Schneiden da. Während man in Version 6 hier nicht unbedingt einen 90-Minüter schneiden wollte, wurde das Editing Toolset in der Version 7 stark ausgebaut und man kann gegebenenfalls auch auf ein dediziertes Editing Tool wie Avid, Premiere, oder Final Cut verzichten – ein wenig Eingewöhnung vorausgesetzt. Des Weiteren kann man im Edit Mode Scene Detections durchführen und Ton auf Shot-by-Shot-Basis anlegen, wenn kein gemeinsamer Timecode vorhanden ist. Dies geht hier dank der Auto-Sync-Funktion um ein Vielfaches schneller als in den gängigen Schnittprogrammen. Auch ein fertiger Audio-Mix für die gesamte conformte Timeline lässt sich hier anlegen. Zuletzt bleibt die Matrix – im Grunde der umfangreichste Teil von Scratch, denn neben Grading ist auch Compositing hier bis zu einem bestimmten Grad möglich.

### Die Matrix

In der Matrix findet also der kreative Teil statt – hier finden sich sämtliche Tools, die ein Mensch zum Grading, zum Compositing und zum Finishing braucht: Input Transforms, Lift



Im Edit Mode können wir Multi-Layer Timelines bearbeiten.



Matrix mit der neuen 3D-Kamera – alle Grading Layer sind im 3D-Raum angeordnet.

/ Gamma / Gain, Look-up Tables und Curves gehören zu den Primarys und sind auch in Scratch Lab enthalten.

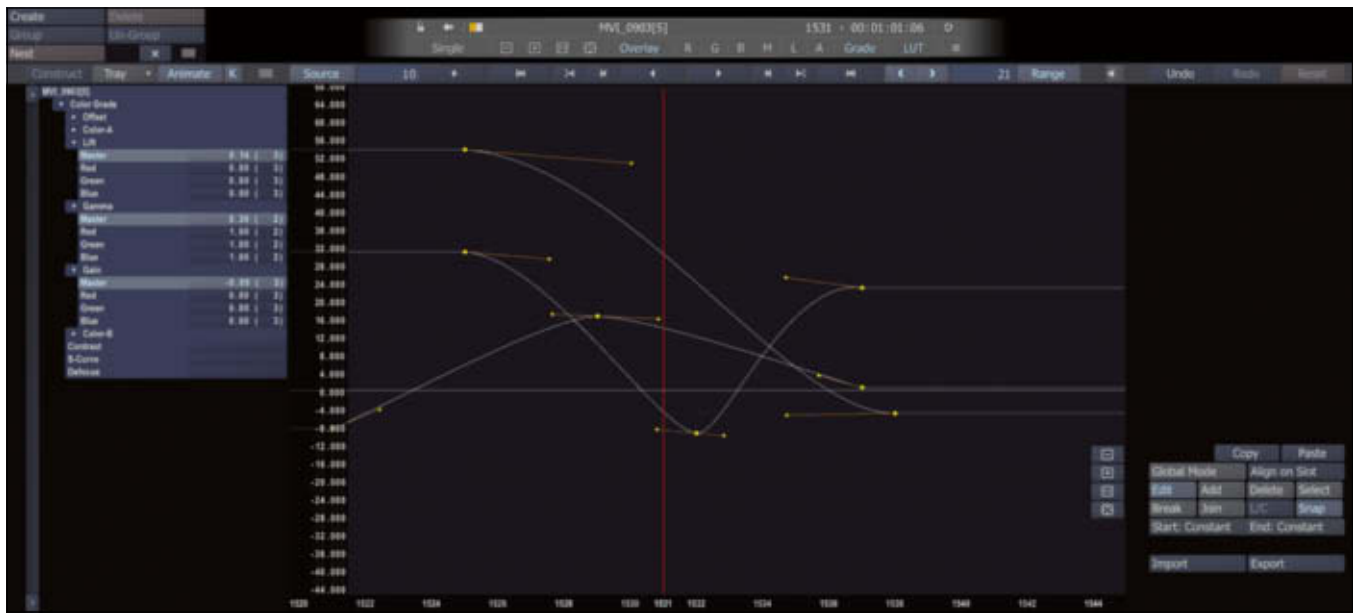
Die Secondarys wie Masken, Vektoren und Keyer sowie Grading-Layer (die sogenannten Scaffolds) und Textures sind dem großen Bruder vorbehalten.

Neu in der Version 7 ist das Kamera-Menü, in welchem wir eine Kamera animieren können, denn alle unsere Grading Layer sind in einem 3D-Raum angelegt. Damit wird auch rudimentäres Compositing möglich gemacht, denn man kann problemlos Green-screen Layer keyen und übereinanderlegen oder CGI-Elemente mit integriertem Alpha auf Objekte tracken oder sogar Objekte aus dem Bild retuschieren. In allen Menüs des Players bleibt die obere Hälfte des Schirms, der Viewport, immer gleich. Die Elemente und Menüs desselben werden jedoch besonders in der Matrix genutzt. Hierzu „stößt“ man mit dem Cursor links, rechts, oder oben gegen den Bildschirmrand und ein Menü erscheint:

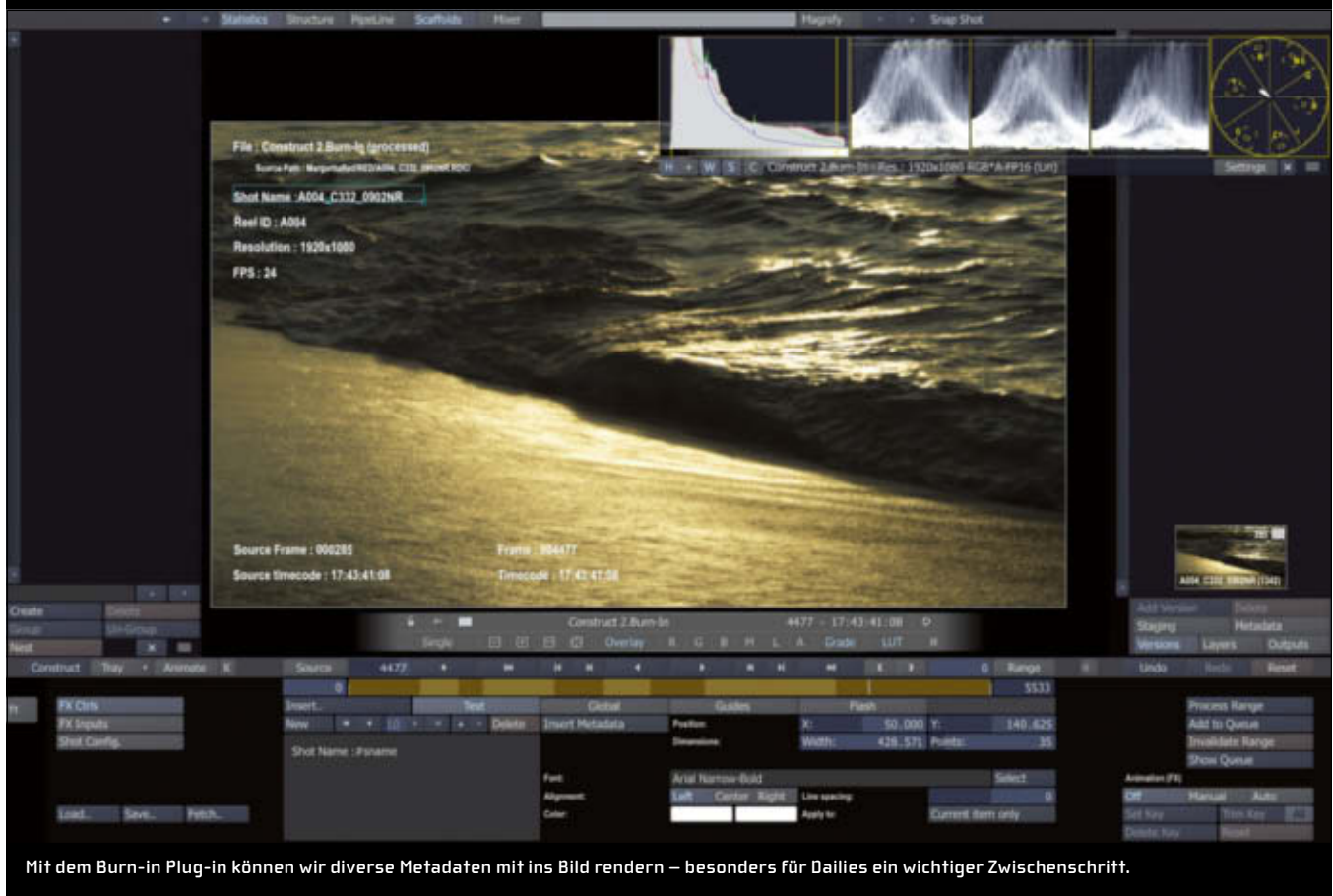
Swiped man nach rechts, erscheint das Versionsmenü.

Dessen Aufgabe ist recht simpel – hat man einen Grade auf einem Clip erstellt, möchte jedoch noch eine Variante ausprobieren, so erstellt man sich durch einen Klick auf „Add Version“ eine neue Version desselben Clips und kann entweder direkt weitermachen oder den Grade zurücksetzen und von vorne anfangen. Wichtig hierbei ist: Die Version, die am Ende gerendert werden soll, muss zuunterst liegen. Hat man also einige Versionen eines Gradings erstellt, kann man sie (während der Clip abspielt) dem DoP oder Regisseur präsentieren und am Ende den gewünschten Grade einfach zuunterst ablegen. Von oben nach unten verlaufen auch die Scaffolds – mit einem Swipe nach links erscheint das dazugehörige Menü – hier können wir neue Layer erstellen, anordnen, gruppieren und deren Blend Modes wechseln.

Der Vorteil hier ist, dass wir einem Grading Layer auch eine Textur zuweisen können – in



Mit dem Animation Editor kann man bequem Animationen erstellen oder verändern.



Mit dem Burn-in Plug-in können wir diverse Metadaten mit ins Bild rendern – besonders für Dailies ein wichtiger Zwischenschritt.

Form eines Greenscreen Clips, der keyed werden muss, oder in Form einer Bauchbinde oder eines anderen beliebigen Objektes, welches im 3D-Raum angeordnet werden kann. Wahlweise kann man einem Scaffold auch ein Plug-in zuweisen – Scratch ist vollständig OFX-kompatibel und so kann man unter anderem GenArts Sapphire, Monsters GT, Neat Video oder Revision FX Plug-ins einsetzen respektive die Lizenzen, die man für andere OFX-kompatible Plattformen wie

Nuke, Fusion oder Sony Vegas bereits besitzt, weiter verwenden. Das bringt viele Freiheiten mit sich, und man muss nicht für jede Compositing-Kleinigkeit Exporte rendern, die Suite verlassen, Files über das Netzwerk schieben und hinterher alles zurückholen.

Mit einem Swipe nach oben erscheint das Hauptmenü – hier können die eben angesprochenen Menüs per Klick angezeigt werden, aber auch ein Audio-Mixer, mit dem wir Audio-Channels pegeln und routen können.

Dazu gibt es noch sogenannte Statistics, wie Histogram, Waveform und Vectorscope – alle laufen in Echtzeit mit.

Mittels der Mini-Timeline in der Mitte navigiert man bequem von einem Shot zum nächsten – Grades kann man sich sowohl auf den acht Memory-Buttons zur rechten oder im Tray, einer Art Gallery, speichern. Es dürfen beliebig viele Trays erstellt werden. Grundsätzlich ist jeder Parameter in Scratch vollständig animierbar – dazu gibt es zur Rechten

die Animation Buttons. Diese funktionieren ähnlich wie in jedem anderen Tool, so dass man manuell Keyframes setzen, aber auch die Auto-Animate-Funktion nutzen kann.

Alle Animationen können im Animation Editor als Splines dargestellt und verändert werden. Dazu zieht man den Animation Button einfach auf den Parameter, dessen Animation man verändern oder ansehen möchte. Alternativ kann man auch einfach in den Animation Editor wechseln und sich die Parameter anzeigen lassen, die man braucht.

## Export

Nach dem Grading wechseln wir zurück in den CONstruct und gehen in den Output-Bereich. Hier sehen wir unser CONstruct als Node, den Main Output. Diesem Node können wir Auflösung, Framerate, Aspect und einen Render Type, zum Beispiel DPX oder TIFF zuweisen. Über Add Output können wir einen neuen Output-Node hinten dranhängen und diesem eine andere Auflösung und auch einen anderen Render Type (JPEG oder Targa) geben. Wir können auch mit dem Node in den Player wechseln und nun unsere gesamte Timeline wie einen einzigen Clip behandeln – zum Beispiel um über den gesamten Film noch eine Vignette zu legen, oder Bauchbinden und Titel als Scaffolds und diese mittels Transparenz ein- und ausblenden.

Genau wie bei einem Clip haben wir auch die Möglichkeit, ein Plug-in auf unser CONstruct anzuwenden – für Dailies bietet sich hier das Burn-in Plug-in an, welches mit Scratch ausgeliefert wird. Hier können wir sowohl sämtliche Metadaten, wie Source-Dateiname, Framerate, Reel-ID, Scene- und Take-Name einblenden, als auch Action und Title-Safes. Einmal konfiguriert lässt sich diese Anordnung als Preset abspeichern und bei Bedarf wieder aufrufen.

Wechseln wir zurück in den Output, ist nun ein neuer Node hinter dem Main Output entstanden – das Burn-in. Statt des Burn-ins können wir auch einen Avid-MXF oder Quicktime Exporter anfügen.

Auch existieren Plug-ins, um direkt nach Interplay zu rendern oder ein fertiges DCP zu erstellen (dazu später im Workflow-Teil mehr). Mittels des MediaReactor Plug-ins von Drastic ist man auch in der Lage MPEGs oder andere branchenübliche Codecs herauszurechnen. Da jeder einzelne Node seine eigenen Parameter hat, können hier komplexe Bäume entworfen werden – beispielsweise vom Main Output (1920x1080) ausgehend in ein Burn-in Node, und von dort einerseits in ein Avid MXF Export Node und in ein Quicktime Export Node – dieses jedoch mit SD-Auflösung, für Preview-Zwecke, während die anderen Nodes ihre HD-Auflösung behalten und so fort. Einen solchen „Render-Baum“



Ein typischer Output Tree mit diversen Outputs für Preview, Editorial, VFX in verschiedenen Auflösungen und Containerformaten

Mithilfe der Metadaten eines Clips können wir automatisierte Output- Filenames generieren.

können wir als Output Preset speichern und bei Bedarf aus einem Dropdown-Menü wieder aufrufen.

Abschließend noch die Namenskonvention. Den Output Filename für jeden Output Node legt man in der Regel mittels diverser Parameter fest, die man hintereinander hängt. Das heißt konkret, dass man sein CONstruct als einen zusammenhängenden Film herausrendern kann, wenn man als Parameter lediglich „#construct“ wählt – so wird ein File mit dem Namen des CONstructs gerendert.

Wählen wir hingegen „#construct\_#sfile“, wird der Source Filename in die Namensgebung mit einbezogen – da sich dieser bei jedem Shot ändert, ist Scratch beim Export nun gezwungen, bei jedem neuen Shot in der Timeline ein neues File zu beginnen, wodurch wir in der Lage sind, auch einzelne Clips zu exportieren. Fügen wir noch die „#frame“-Komponente ein, wird für jeden Frame ein separates File gerendert, was bei Bildsequenzen absolut Sinn macht. Im Übrigen können Parameter nicht nur durch Unterstriche separiert werden, sondern auch durch Slashes – dann werden separate Ordner mit dem jeweiligen Parameter als Namen erstellt.

Haben wir einmal eine Namenskonvention entworfen, können wir auch diese als Tem-

plate abspeichern. Jedes abgespeicherte Preset, jedes Projekt und jeder User kann via USB-Stick auf jede beliebige Scratch-Workstation mitgenommen und dort eingefügt und genutzt werden. Das funktioniert innerhalb einer Release-Version (Scratch v6.1.714 auf Scratch v6.1.729) problemlos – bei einem Sprung von v6.1 auf v7.0 kann es von Fall zu Fall sein, dass nicht alle Parameter gleich übernommen werden – hier ist Vorsicht und vor allem Kontrolle geboten. Grundlegende Dinge wie User oder Projekt können immer in einem neueren Release geöffnet werden – werden aber im selben Moment inkompatibel zur vorherigen Version, deswegen hier auf jeden Fall Sicherheitskopien machen. Zu guter Letzt fügt man die Outputs, die nun tatsächlich gerendert werden sollen, in die sogenannte Render Queue ein, startet diese und kann während sie durchläuft schon in anderen CONstructs weiterarbeiten.

## Der Workflow

Wie sähe nun ein Workflow aus, der am Set beginnt und bei einem fertigen Deliverable endet? Da hier sowohl verschiedenartige Hardware, als auch verschiedene Software-Pakete gefragt sind, gehen wir an dieser Stel-

Name	Format	Range	Status	Free space	Path
Group: Battle_002					
Avid DNxHD 36 HQF	Res.: 1920x1080 RGBA-8 (Lin)	0 - 5533	00:06 / 20:30 - 4.48 fps	0.9%	VolumeData\Scratch_Demo_Footage\Render\
ProRes 422 HQ	Res.: 1920x1080 RGB-16 (Lin)	0 - 5533	00:04 / 17:48 - 5.17 fps	0.4%	VolumeData\Scratch_Demo_Footage\Render\
H.264 PAL Hrd	Res.: 1024x576 RGB-16 (Lin)	0 - 5533	00:03 / 05:38 - 18.32 fps	0.3%	VolumeData\Scratch_Demo_Footage\Render\
DPX for VFX-Dept.	Res.: 1920x1080 RGB-10 (Lin)	0 - 5533	00:03 / 09:44 - 9.44 fps	0.3%	VolumeData\Scratch_Demo_Footage\Render\DPX for VFX-Dept\
Avid DV PAL	PAL: 720x576 RGBA-8 (Lin)	0 - 5533	00:03 / 11:48 - 7.79 fps	0.4%	VolumeData\Scratch_Demo_Footage\Render\

Die Process Queue zeigt alle zum Export anstehenden Outputs, nach CONstruct sortiert, an.



Mit dem Audio-Mixer und der Auto-Sync-Funktion können wir Ton auch ohne Timecode anlegen, pegeln und routen.

le nicht nur auf Scratch, sondern auch auf andere hilfreiche Tools ein.

Beginnend am Set wird eine D.I.T. Station benutzt – zum Beispiel die Wrangler Pro Workstation von 1beyond, oder eine On-Set Cart von JMR, beides zuverlässige und vor allem robuste On-Set-Systeme. Zunächst wird das Footage mit dem sehr effizienten Backup Tool „Silverstack“ der Firma Pomfort auf ein Raid-System, mehrere externe Festplatten und LTO gebackupt und die Vollständigkeit der Daten mittels Checksummen abgeglichen. Eine der Backup-Locations (eine der externen Festplatten) wird unser Media-Directory, in dem sich alles Footage findet. Oft sieht man sich heutzutage gezwungen, mit mehreren Kameraformaten umgehen zu müssen. Gehen wir davon aus, dass wir hier sowohl RED-Material, als auch Canon 5D/7D

h.264-movs sowie extern aufgenommenen Ton mit Timecode, passend zu dem der RED-Clips, verarbeiten müssen. In Scratch Lab erstellen wir nun also ein Projekt und dazu die passende CONstruct-Hierarchie – beispielsweise zwei Groups (Red und Canon), die die jeweiligen CONstructs (Canon\_Reel\_01, ... Reel\_02, usw.) beinhalten.

In diese CONstructs wird nun das Footage geladen – über den Mediabrowser können wir bei den RED-Clips die Debayering Settings für alle importierten Clips einstellen und die angelieferten Audiofiles synchronisieren. Für alle h.264-movs der Canon-Kameras generieren wir sowohl eine Reel-ID, als auch neuen Timecode um die Clips eindeutig identifizieren zu können.

Wir wählen das erste CONstruct und gehen in den Player – auf den ersten Clip legen

wir eine LUT, die vorher in Pomforts Live-grade, oder in Scratch selbst erstellt wurde (sowohl 1D-, als auch 3D-LUTs sind möglich). Über die Copy&Paste-Forward-Funktion wird das Grading nun auf alle nachfolgenden Clips im CONstruct kopiert und wir können die Clips bei Bedarf durchsehen. Falls nötig, syncen wir die Audiofiles auf die h.264-movs im Edit Mode von Hand, indem wir uns den Audio-Mixer zunutze machen: Wir scrubben auf den Frame, auf dem die Klappe schlägt, laden das passende Audiofile (in der Regel korrespondiert der File-Name mit den Informationen auf der Klappe) und verwenden die Auto-Sync-Funktion von Scratch, um den Ton passend zu slippen. Nächster Shot, Load Audio, Auto-Sync, und so fort. Bei Bedarf können wir den Ton auch millisekundenweise slippen. So verfahren wir mit jedem neuen CONstruct.

Als Nächstes geht es an den Output: Wir fügen ein Burn-in Plug-in ein und laden ein von uns vorkonfiguriertes Preset, welches uns immer für den jeweiligen Clip auf dem wir stehen, die passenden Metadaten anzeigt. Hinter das Burn-in Plug-in hängen wir einen Avid-MXF (DNxHD 36) Output und einen ProRes Output in HD-Auflösung sowie einen h.264 Output in SD-Auflösung für Preview-Zwecke.

Den Baum, den wir gebaut haben speichern wir uns als Template ab, damit wir ihn bei allen nachfolgenden CONstructs nur noch aufrufen und nicht neu bauen müssen. Als Nächstes bekommt jeder Output seinen Output Filename – die DNxHD-MXF Files sollen clipweise in einen Unterordner namens „Avid-Editorial“, während die ProRes Clips nach „FCP-Editorial“ und die SD-h.264-movs nach „html“ exportiert werden sollen. Der Output Filename für die Avid-MXFs sähe dann etwa so aus:

`/#project/Avid-Editorial/#construct_#reelid[0]/#sfile.`

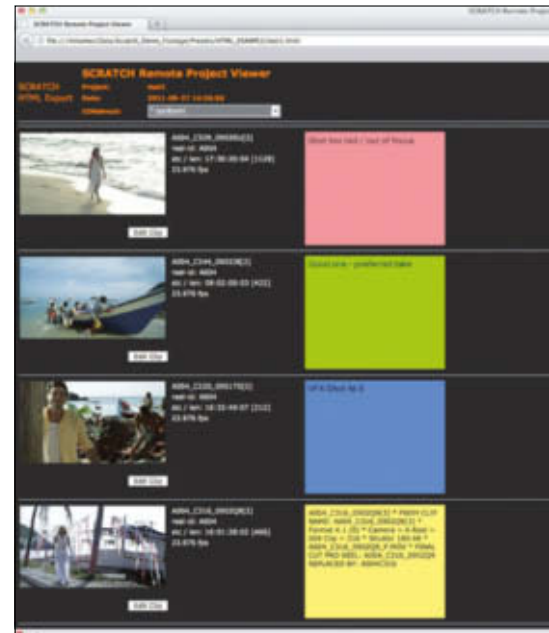
Das heißt, es wird eine Ordnerstruktur aus dem Projektnamen, manuell eingetipptem „Avid-Editorial“ und Construct-Name und Reel-ID erzeugt und dorthinein jeder Clip mit seinem original Filenamen gerendert. Auch dieses Template speichern wir uns ab, damit wir es in Zukunft nur noch aus dem Dropdown-Menü aufrufen müssen.

Danach fügen wir die Exports, die wir benötigen an die Process Queue an und feuern

dieselbe ab. Während die Queue durchläuft, können wir bereits neue CONstructs anlegen, mit Footage füllen und weiterarbeiten. Als Nächstes exportieren wir das Scratch-Projekt als html-Version auf einen lokal installierten html-Server (zum Beispiel xampp) und fortan kann jeder am Dreh Beteiligte via Browser die angelegten CONstructs und die darin befindlichen Shots anschauen und mit farbigen Kommentaren versehen – etwa welche Shots erste Wahl sind (grüne Notes), welche Shots ans VFX-Department müssen (blaue Notes) und so fort.

Durch einen Klick auf „Refresh Project“ kann sich der Operator alle Kommentare als Notes auf jedem Clip in Scratch anzeigen lassen. Nun folgt der Schnitt auf Avid, respektive Final Cut – die ProRes QuickTimes lassen sich ohnehin problemlos in FCP importieren. Die gerenderten DNxHD-MXF Files kommen mit dazugehörigem AAF. Dies bedeutet konkret, dass wir die MXF-Files lediglich in den AvidMediaFile-Ordner legen müssen, und die dazugehörigen AAF-Files in einem Rutsch (oder einzeln / reelweise) importieren – voilà, das Projekt ist online. Bei einer bestehenden Interplay-Anbindung haben wir über ein Plug-in der Firma DiaQuest auch die Möglichkeit, direkt auf den Interplay-Server zu ingesten. Das ist besonders für größere Posthäuser und Fernsehsender interessant.

Während der Schnitt auf Hochtouren läuft, bereiten wir das Finishing vor. Dazu kopieren



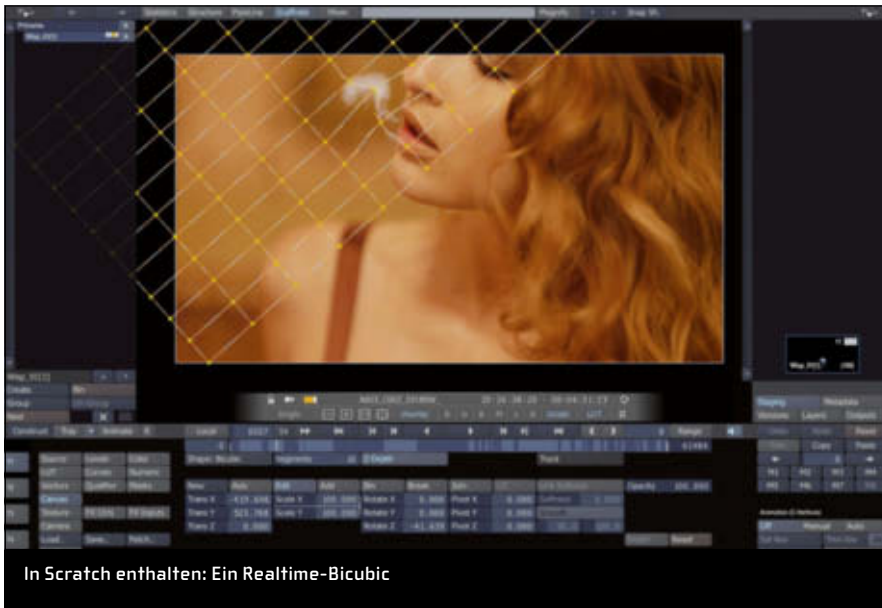
Mittels Browser kann jeder Kunde von jedem Ort aus die Clips im Scratch-Projekt anschauen und kommentieren.

wir das Scratch-Lab-Projekt vom Wrangler herunter und auf die Scratch Finishing Workstation in der Grading Suite – das hat den Vorteil, dass wir das Projekt und alle CONstructs nicht erneut anlegen müssen, sondern das bestehende Projekt öffnen und dort weitermachen können, wo wir am Set aufgehört haben. Ebenfalls kopieren können wir ggf. User,



Im Dual View Mode können wir unsere conformte Timeline mit einem Referenz-File abgleichen.



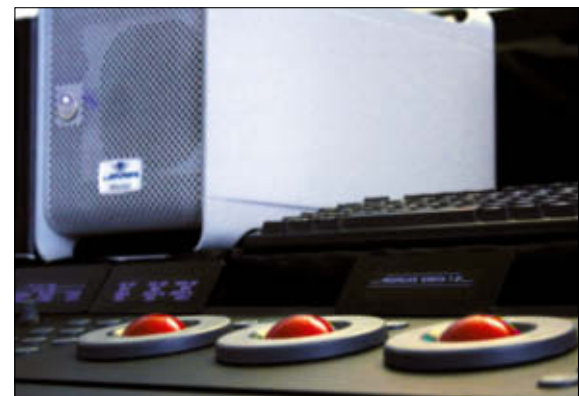


auswählen. Mittels Lightspace CMS von Light Illusion können wir Scratch außerdem problemlos in eine bestehende Color-Management-Pipeline integrieren.

Einige Shots erfordern sicherlich intensiveres Compositing – für das VFX-Department benötigen wir in der Regel DPX- oder Open-EXR-Bildsequenzen. Besagte Shots wurden am Set schon mit blauen Notes gekennzeichnet – wir müssen sie jetzt nur noch finden und rendern. Dazu gibt es eine Suchfunktion – mit dieser können wir in einem bestimmten CONstruct, einer Group oder dem ganzen Projekt suchen. Dabei können wir unser Projekt nach bestimmten Metadaten durchforsten, oder auch ganz ohne Suchworte einfach nach allen Shots mit blauen Notes (VFX-Shots) drauf suchen und die gefundenen Shots in einem neuen CONstruct konsolidieren und clipweise als DPX-Sequenz herausrendern. Während des Gradings machen wir uns neben den üblichen Grading Tools wie Masken, Keyern, Curves und Vektoren sowohl die Custom Commands, als auch OFX-Plug-ins, oder die bereits in Scratch integrierten Plug-ins wie Vector Paint (sehr nützlich für einfache Retuschearbeiten) oder ReTimer zunutze. Auch bringt Scratch von Hause aus einen Realtime-fähigen Bicubic mit, mit dessen Hilfe wir das Bild auch während des Playbacks nach Belieben verzerren können.

Viele Dinge wie Object Tracking, Green-screen-Compositing oder Retouching kann man direkt in Scratch erledigen und bewahrt so den ein oder anderen Regisseur vor dem Freak-out kurz vor der Deadline. Aber es spart vor allem Zeit und Geld. Über den neuen Source Mode können wir alle Grading Layer in einer Timeline-Ansicht betrachten und damit so trimmen und timen, wie es für den jeweiligen Shot nötig ist. Nach einer Grading Session exportieren wir erneut kleine h.264 QuickTimes und das Projekt als html-Version, damit Regie oder Agentur jeden Clip via Browser begutachten können und Änderungswünsche in Form von Kommentaren abspeichern.

Zur nächsten Session hat der Colorist nach einem Klick auf „Refresh Project“ alle abgegebenen Kommentare in Form von Notes auf



Output-Templates und Burn-in Presets. Die Scratch Workstation ist eine aktuelle HP Z820 mit zwei 8-Core Intel Xeons, 32 GB Arbeitsspeicher und Nvidia Quadro 6000.

Quadro-Karten sind von Assimilate offiziell getestet, die neuesten AMD-Karten werden jedoch ebenfalls unterstützt. Als Grading-Panel kommt ein Set Elements von Tangent zu Einsatz – diese sind anständig verarbeitet, bieten eine tolle Haptik und das bei moderatem Preis. Daneben gesellt sich ein Wacom-Grafik-Tablet – unerlässlich für das Zeichnen von Masken. Hinzu kommt ein JMR Silverstor PCIe Raid, was uns theoretisch bis zu 2,1 GB/s liefert. In der Praxis variiert der Wert jedoch je nach File-Größe der Frames.

Wer höhere Datenraten braucht, kann kompakten SSD-Hochleistungsspeicher von Fusion IO, oder Fibre Channel Raids zum Beispiel von DotHill nutzen – diese schaffen zwischen 2,5 und 3,0 GB/s. Für größere Facilities, gerade mit mehreren Schnittplätzen, Grading Suiten und VFX-Arbeitsplätzen bietet sich unter Umständen auch gleich ein zertifiziertes SAN als zentrales Storage an. Video IO wird über zwei Bluefish444 Epoch Supernova Karten gelöst, mit denen wir Auflösungen bis zu 4096x2160 Pixeln in 12 bit RGB 4:4:4 über HD-SDI ausgeben können. Endgerät ist ein TV Logic XVM-245W Class-1 Monitor und ein Christie HD7K-J 3-chip DLP Projektor.

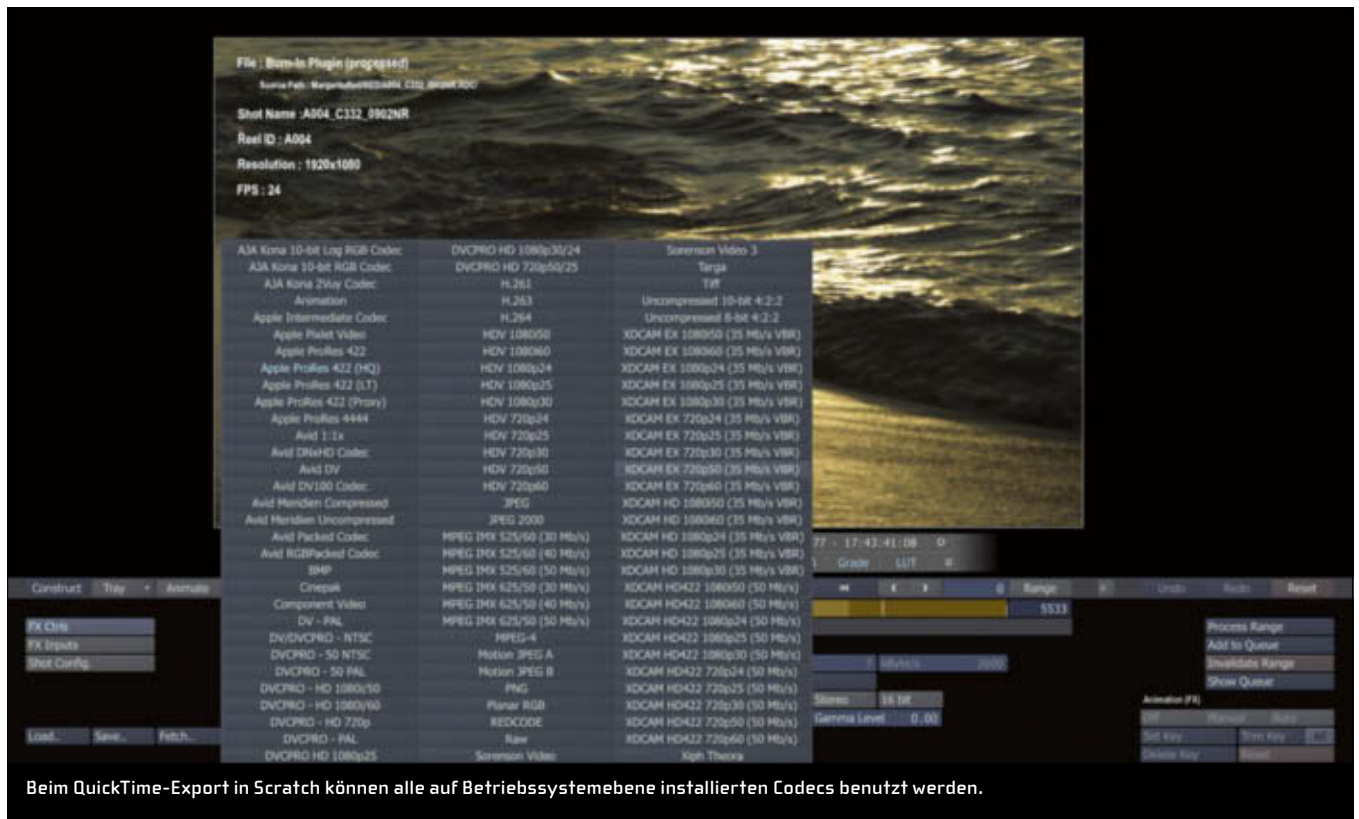
Wir hängen die externe Festplatte mit dem gesammelten Footage an die Workstation, aktualisieren die Location des Media-Directorys und öffnen das Projekt – alle Mediafiles sind sofort online und wir können mit dem Conform beginnen. Dazu laden wir die EDL/XML oder das AAF aus dem Avid und können die Kriterien zum Matchen auswählen. Dank des generierten Timecodes für die Canon h.264-movs können wir problemlos auf unsere Original-Files relinken. Nachdem wir die neue

Timeline assembled haben, laden wir uns ein Offline-Referenz-File, um den Schnitt zu überprüfen – dieses wandert geradewegs in einen Reference Tray. Aus den View-Options können wir sowohl Over als auch Dual Mode auswählen, um nun unsere conformte Timeline und das Offline File neben- oder übereinander im Split-Screen zu sehen. Schnitte und Überblendungen können wir so im Edit-Mode problemlos anpassen.

Nachdem wir mit unserem Conform zufrieden sind, exportieren wir das gesamte Projekt als Scratch-Projekt, um ein Backup zu haben. Denn als Nächstes löschen wir alles aus dem Projekt, außer der frisch conformten Timeline. Wir verlassen das Projekt und gehen in die Project-Settings – hier versteckt sich eine Consolidating-Funktion, von der wir jetzt Gebrauch machen.

Diese Funktion macht nichts anderes, als alles im Projekt verwendete Material vollständig und in Sequence an eine neue Location zu befördern. Da unser Projekt nur noch ein CONstruct mit dem Footage enthält, was tatsächlich verschnitten worden ist, wird auch nur das darin enthaltene Footage von der externen Festplatte auf das JMR-Raid kopiert und spart so Zeit und Speicherplatz.

Beim Consolidate werden sowohl die originalen Ordnerstrukturen kopiert, als auch das Footage in Scratch auf die neue Location (das Raid) gelinkt. Ist dieser Vorgang abgeschlossen, können wir die externe Festplatte abhängen. Bei Bedarf oder Schnittänderungen können wir die neuen Shots jederzeit von dort importieren. Als Nächstes fügen wir sämtliche benötigten Grafikelemente, Lookup-Tables, etc. zu unserem Media-Directory hinzu und können anfangen den Film zu graden. Über die Player-Settings können wir uns sowohl Action und Title Safes als Overlay anzeigen lassen, als auch Display-LUTs für den Interface-Monitor oder den SDI Output



Beim QuickTime-Export in Scratch können alle auf Betriebssystemebene installierten Codecs benutzt werden.

den Clips in Scratch und kann nach Anweisung weiter korrigieren.

Bestimmte Situationen erfordern hin und wieder einen Re-Conform, wenn der Schnitt aus rechtlichen Gründen oder Ähnlichem doch noch einmal geändert worden ist. In Scratch haben wir nun zwei Möglichkeiten in einem solchen Falle zu verfahren:

Wir können unser existierendes CONstruct nehmen und darauf den neuen Conform fahren – sollten neue Clips hinzugekommen sein, die nicht in diesem CONstruct (oder dem gesamten Projekt) enthalten sind, können wir die externe Festplatte wieder anhängen, die Einträge der fehlenden Clips in der Conform-List markieren und die externe Festplatte als Location der Source Files angeben.

Damit haben wir auch neue Clips einfach dazugepackt – das Grading (inklusive aller Versions) auf den bereits existierenden Shots bleibt dabei vollständig erhalten. Es empfiehlt sich, hiernach nochmals einen Consolidate zu machen, damit die neuen Shots, die noch auf der externen Festplatte lagern, auch ins Media-Directory auf das Raid-System kopiert werden (Scratch berücksichtigt schon vorhandene Mediainfiles und kopiert nur das Footage, das dort noch nicht existiert).

Die zweite Möglichkeit wäre, einen komplett neuen Conform von Anfang an zu starten – mit dem Ergebnis einer vollkommen frischen Timeline ohne jedes Grading. Um jedoch nicht alle bisher geleistete Arbeit zu verlieren, öffnen wir noch einmal das alte CONstruct, welches das bisherige Grading enthält, markieren alle enthaltenen Clips

und klicken auf Copy. Im neu conformten CONstruct angekommen, kopiert ein Klick auf den Paste Grade Button die Grading eines jeden Shots auf den korrespondierenden Shot im neuen CONstruct – unabhängig von der veränderten Location in der neuen Timeline.

Dies kann man sich auch zunutze machen, wenn man ein Pre-Grading auf Offline Files erstellt hat und nun alle Grades auf ein CONstruct mit den originalen Kamera-Files kopieren will. Zuletzt wird der fertige Mix aus dem Tonstudio unter die Timeline gelegt, die VFX-Shots eingefügt und der Film entweder auf HDCAM SR ausgespielt, respektive in einem File-Format herausgegeben. Grundsätzlich ist es möglich, auf beiden Plattformen, Windows und OSX, sowohl professionelle Bildsequenzformate wie DPX, TGA, oder TIFF, als auch Avid MXF, oder QuickTime-files in allen branchenüblichen Codecs wie DNX-HD, h.264, ProRes, oder XDCAM zu rendern.

Dabei kann man sowohl über Drastics MediaReactor, als auch über ein dediziertes DCP-Plug-in der Firma 4K Inc. ein fertiges DCP herausgeben – inklusive KDM und Subtitles. Mittels Easy DCP Creator und Player vom Fraunhofer IIS ist weiteres DCP-Authoring und auch Wiedergabe und Kontrolle des erstellten DCPs möglich.

### Fazit

Der hier geschilderte Workflow ist einer von vielen möglichen. Assimilate Scratch ist definitiv nicht nur ein Grading Tool, sondern

universell einsetzbar – ob nun am Set für Review, Dailies, für Conforming, Grading oder Finishing. Die Möglichkeit, eigene Scripts und Plug-ins zu implementieren erweitert die Flexibilität dieses Tools nochmals um ein Vielfaches. Ein weiterer Vorteil ist, dass man nicht auf bestimmte (im schlimmsten Falle proprietäre) Hardware angewiesen ist, sondern sich sein System, dem Bedarf entsprechend selbst zusammenstellen kann.

Von der 4K-Grading Solution mit der besten und schnellsten Hardware, bis hinunter zur MacBook-Pro-Konfiguration für flexiblen On-Set-Einsatz ist alles möglich. Gerade in Zeiten von immer knapperen Budgets und schier unmöglichen Produktions-Deadlines ist ein schneller, funktionierender Workflow das Rückgrat einer jeden Produktion. All das macht Scratch zu einer ernst zu nehmenden Alternative zu den etablierten Systemen unserer Branche, denn Scratch ist nicht nur ein Tool – es ist eine Pipeline. > ei



Matthias Aderhold ist Application Engineer und Workflow-Spezialist bei Dreamwalks – Solutions for Moving Pictures in Taufkirchen, südlich von München. Schwerpunktmäßig arbeitet er auf den Gebieten Compositing und Color Grading. Durch seine Ausbildung und die Zusammenarbeit mit den größten Posthäusern in Deutschland, Österreich und der Schweiz hat er weitreichende Erfahrungen im Umgang mit den verschiedensten Kameraformaten, Applikationen und den Workflows, die sich aus den unterschiedlichsten Komponenten zusammensetzen.