

PETER ~~RUHRMANN~~ HOCHSCHULE DER MEDIEN

Compositing 2025

Aufbau und Ziel der Lehrveranstaltung

- Lehrveranstaltung besteht aus theoretischem und praktischem Teil. Im praktischen Teil wird das geübt und nachvollzogen, was im theoretischen Teil behandelt wurde.
- Ziel ist ein Verständnis für die Grundlagen und Methoden des Compositings.
Die Inhalte lassen sich auf die unterschiedlichen Softwarepakete übertragen. Die reine Programmbedienung steht daher weniger im Vordergrund.

Organisatorisches: Prüfungsleistung

- Prüfungsleistung ist eine praktische Arbeit
 - Da alle die gleiche Aufgabe bekommen, stelle ich bei der Abnahme ein paar Verständnisfragen.
 - Bewertungskriterien sind
 - Qualität (wie sieht es aus?)
 - Effektivität (wie schnell rendert es?)
 - Struktur (wie übersichtlich ist das Comp?)
 - Kreativität (welche technischen und gestalterischen Lösungsansätze wurden gewählt?)
- 80% Anwesenheitspflicht

Organisatorisches: Termine

- 8 Doppeltermine
- Danach 2 Wochen Zeit für Abgabe

27.03	10.04.	17.04.	24.04.	08.05.	15.05.	22.05.	05.06	26.06. / 27.06
-------	--------	--------	--------	--------	--------	--------	-------	-------------------

Definition





einfache Bildmanipulationen

einfache Bildmanipulationen: Grundlagen

- Kanäle
 - Bildinformationen sind in Kanälen gespeichert
 - Kanal ist ein Graustufenbild, das eine bestimmte Eigenschaft eines Bildes repräsentiert, z.B. den Grünanteil eines Bildes



Links RGB-Bild, rechts Grünkanal

- übliche Kanäle sind Rot-, Grün-, Blau- und Alphakanal (Transparenzinformation). Weitere Kanäle sind z.B. Z (Tiefe), X-Vektor, Y-Vektor...

einfache Bildmanipulationen: Grundlagen

- Normierung
 - Alle brauchbaren Compositing-Programme arbeiten mit einem normierten System, d.h. die Werte in den Kanälen und die Werte für Positionsdaten sind alle zwischen 0 und 1
 - Vorteil: Auflösungs- und Bittiefenunabhängig.

Beispiel:

Ein Bild mit 8 Bit Farbtiefe hat Werte für R,G,B zwischen 0 und 255. Speichert man ein mittleres Grau, würden die Werte 127,127,127 gespeichert. Würde man später auf 10 Bit Farbtiefe umstellen, müssten die Werte in 511,511,511 umgestellt werden.

In einem normierten System wird dagegen einfach 0.5, 0.5, 0.5 gespeichert, unabhängig von der Bittiefe.

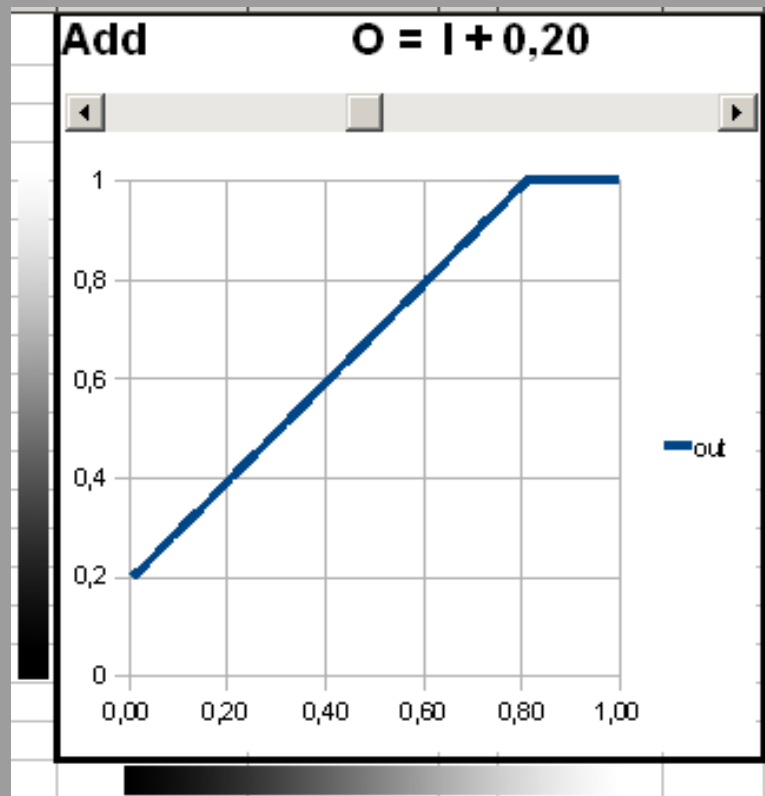
1. Zwischenübung

Erkundet Nuke. Das solltet ihr können:

- Material über Read-Node hinzufügen
- Verschiedene Kanäle im Viewer anzeigen
- Kontextsensitive Shortcuts (r-Taste während Maus über Viewer, oder r-Taste während Maus über Node Graph)
- Einen Node hinzufügen über Toolbar, Quickmenü, Kontextmenü (rechte Maustaste)
- Einen Node beim hinzufügen direkt verbinden
- Einen Node an einer beliebigen Stelle ohne Verbindung hinzufügen
- Navigieren in allen Fenstern (Zoomen, Panning)
- Einen Node ins Properties Panel laden

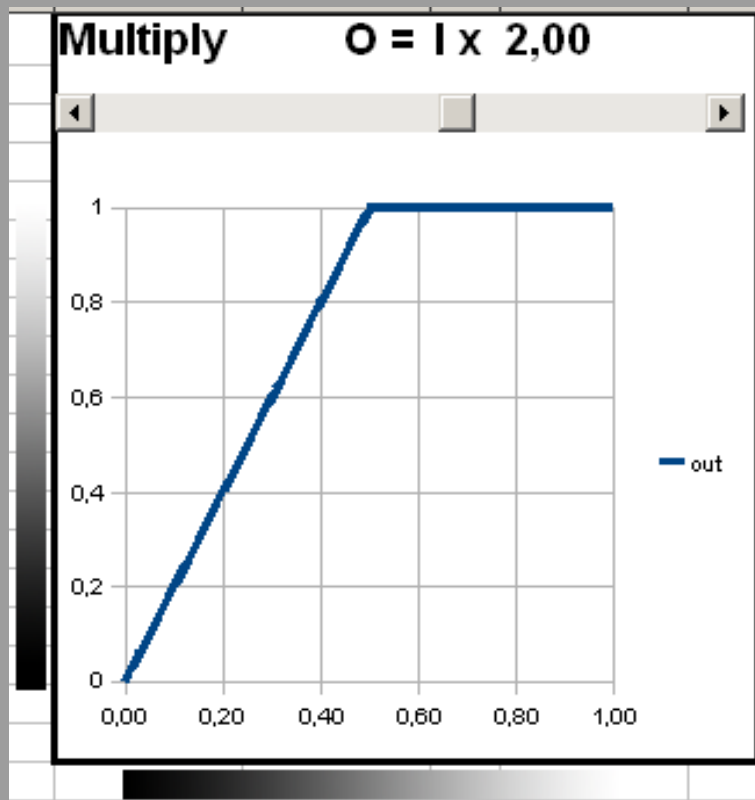
einfache Bildmanipulationen: Operatoren

- Add: $O = I + X$
 - addiert zu jedem Pixel den Wert X



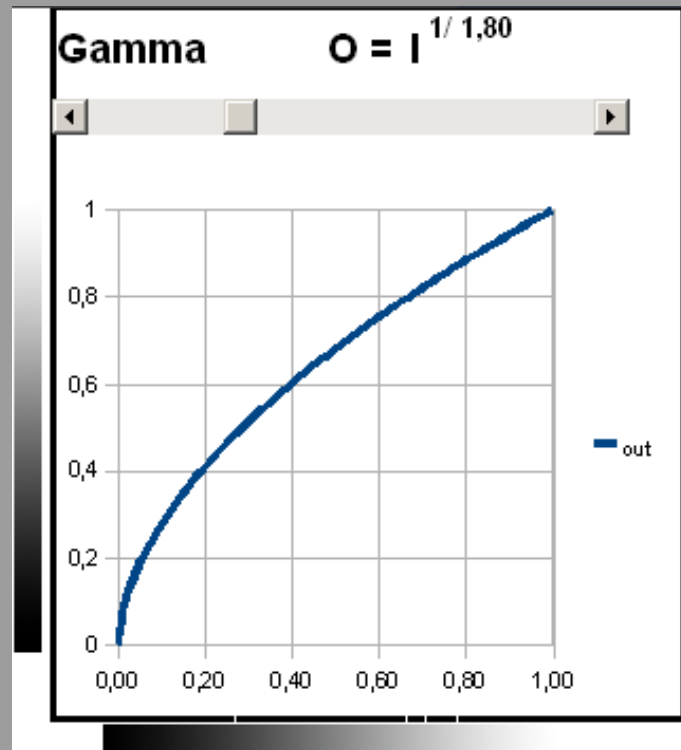
einfache Bildmanipulationen: Operatoren

- Multiply: $O = I \times X$
 - multipliziert zu jedem Pixel den Wert X



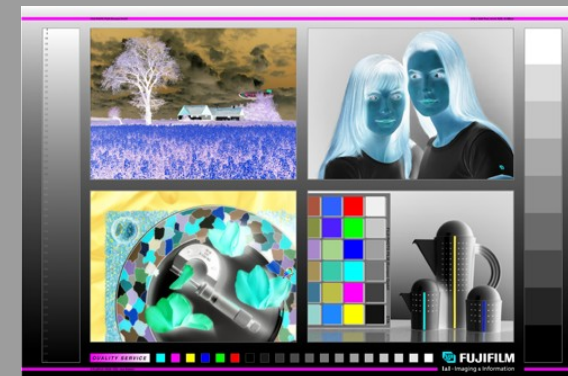
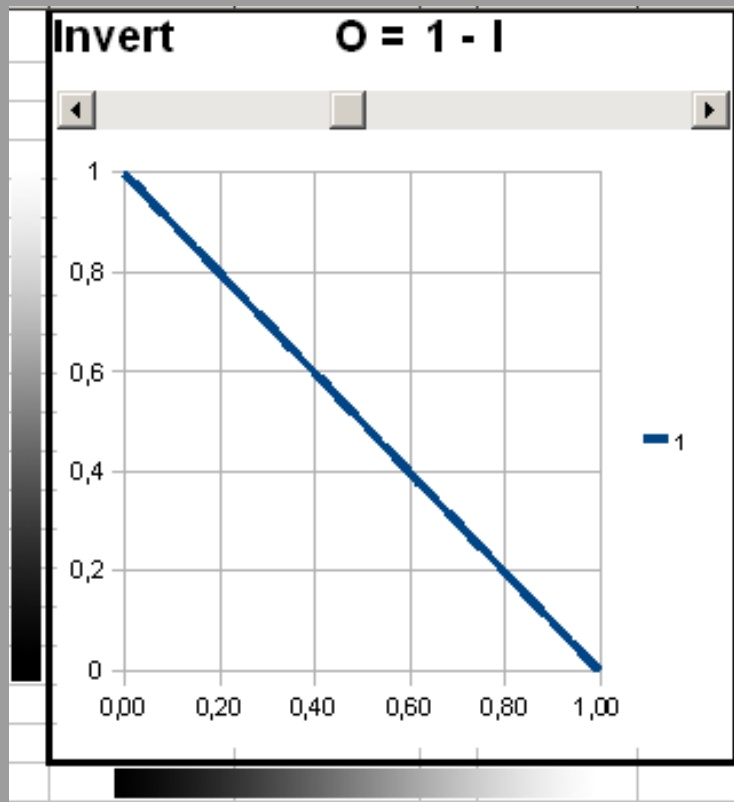
einfache Bildmanipulationen: Operatoren

- Gamma: $O = I^{1/\text{Gamma}}$
 - jeder Pixel wird mit $1/\text{Gamma}$ potenziert
 - Besonderheit: Schwarz bleibt schwarz und weiss bleibt weiss



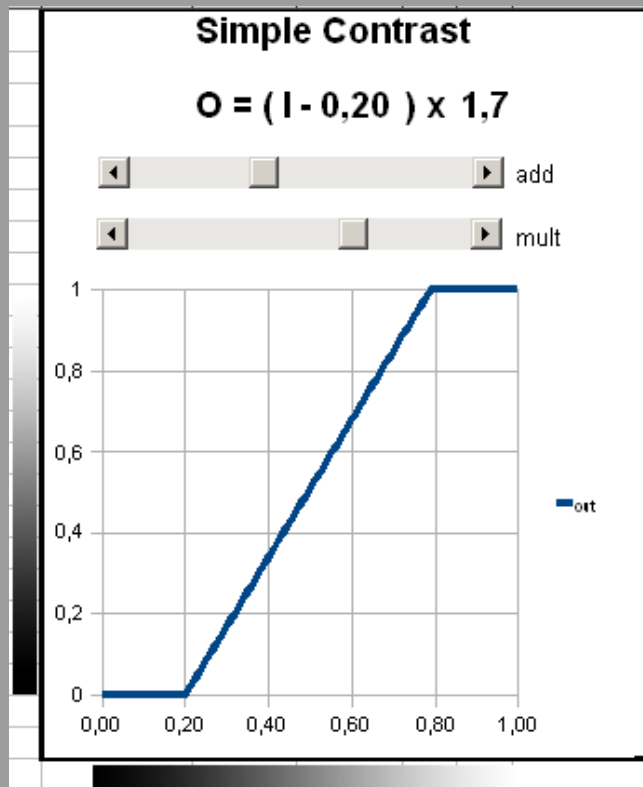
einfache Bildmanipulationen: Operatoren

- Invert: $O = (1 - I)$
 - Keine Variablen



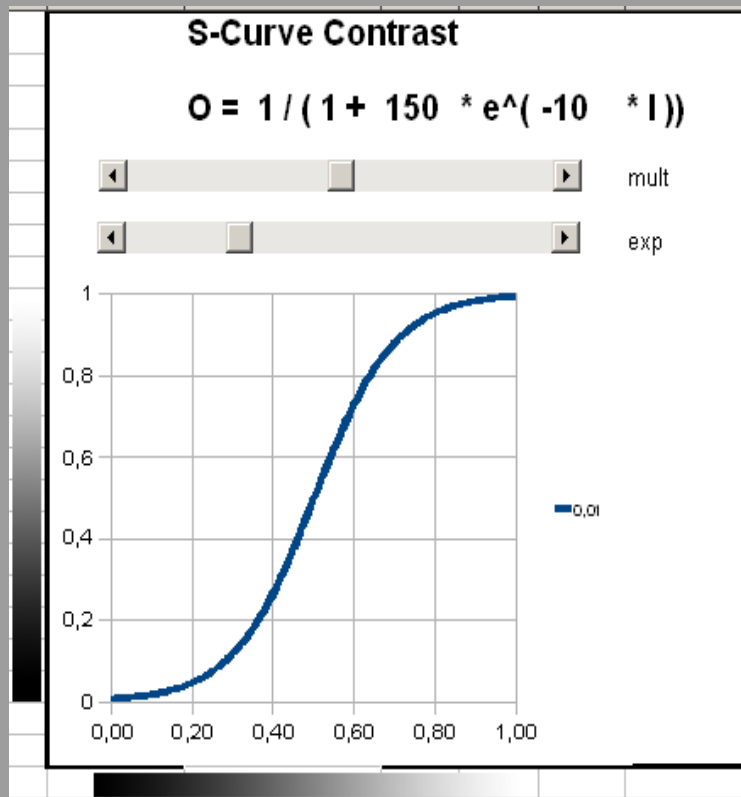
einfache Bildmanipulationen: Operatoren

- Simple Contrast: $O = (I - X) \times Y$
 - Kombination von **Add** und **Multiply**



einfache Bildmanipulationen: Operatoren

- S-Curve Gamma: $O = \frac{1}{(1 + x \cdot e^{-y \cdot I})}$



einfache Bildmanipulationen: Transformationen

- geometrische Transformationen
 - panning: Verschieben in x und y Richtung
 - rotating: Drehen um eine Achse
 - flip, flop: horizontales, vertikales Spiegeln
 - scaling: vergrößern, verkleinern
 - warping: freies Verzerren anhand eines Gitters

- Loggt euch mit eurem HdM-Account ein.
- Macht euch mit der Oberfläche von Nuke vertraut. Ladet in ein neues Projekt das Bild grautreppe.exr
- Erstellt mit Expression-Nodes die Operatoren add, multiply, gamma, invert, contrast. Überprüft mit Hilfe der Grautreppe die Plausibilität der Ergebnisse
- Erstellt mit einem Expression-Node eine einfache Farbkorrektur, die es ermöglicht mit drei Reglern r,g,b in Gamma zu korrigieren.
- Vergleicht den Grade-Node mit den selbst erstellten Tools. Gibt es Regler, die sich wie die selbst erstellten Tools verhalten? Verändert immer nur einen Regler und stellt ihn anschließend in die Ausgangsposition zurück.
- Probiert an einem Bild eurer Wahl mit Hilfe des Transform-Nodes verschiedenen Panning- und Rotationsmöglichkeiten aus, z.B. auch mit verschobenem Pivot-Punkt

einfache Bildmanipulationen: Filter

- Filteralgorithmen bei geometrischen Transformationen
 - Impulse (Nearest-neighbor): verdoppelt bzw. entfernt Pixel. Schnell aber viel Aliasing
 - Cubic: weiche Interpolation, allerdings viele Blöcke beim hochskalieren.
 - Mitchell: Guter Kompromiss zwischen Schärfe und Ringbildung
 - Parzen, Notch: fast keine Artefakte, aber sehr weich, gut für Bilder mit Verläufen
 - Sinc: sehr gut zum runterskalieren, erhält Details, wenig Artefakte.

einfache Bildmanipulationen: Filter

- Filter
 - Median Filter

0,6	0,6	0,9	0,9	0,9
0,6	0,6	0,9	0,9	0,9
0,6	0,6	0,9	0,9	0,9
0,6	0,6	0,6	0,9	0,9
0,6	0,6	0,6	0,6	0,9

Sortierung nach Helligkeit:

0,6

0,6

0,6

0,6

Listenmitte:

0,9

0,9

0,9

0,9

0,9

einfache Bildmanipulationen: Filter

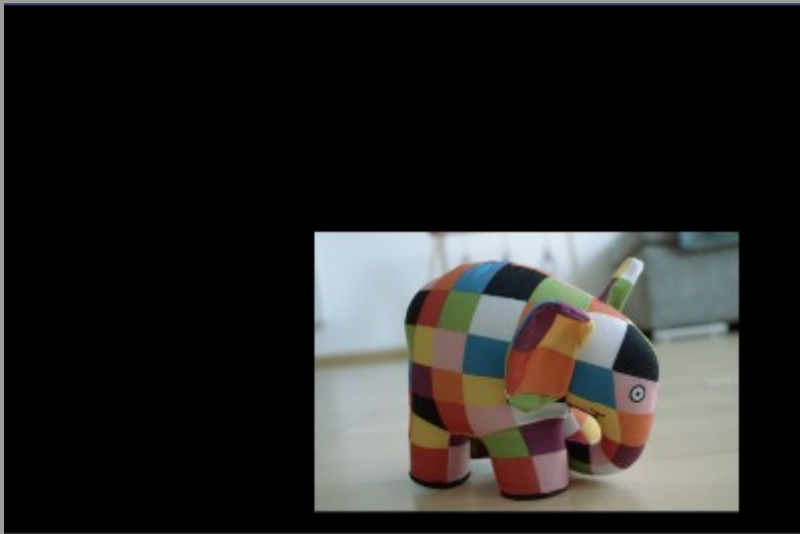
insbesondere Edge-Detection-Filter wichtig zur Erstellung von Edge-Mattes:

- Sobel: erzeugt eine breite Kante
- Laplacian: erzeugt feine Kanten

einfaches Compositing

einfaches Compositing

- Zusammenfügen zweier Bilder mit Add



+

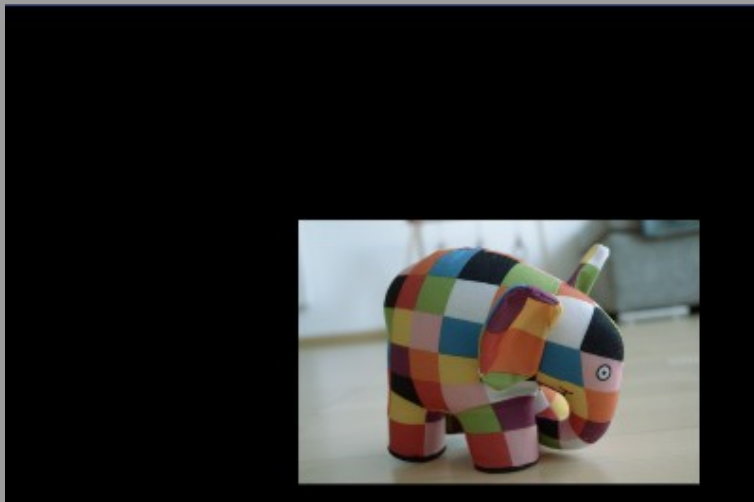


=



einfaches Compositing: Over-Operator

- Steuerung der transparenten und nicht transparenten Bereiche durch Matte-Image (wird in Alpha-Kanal gespeichert)



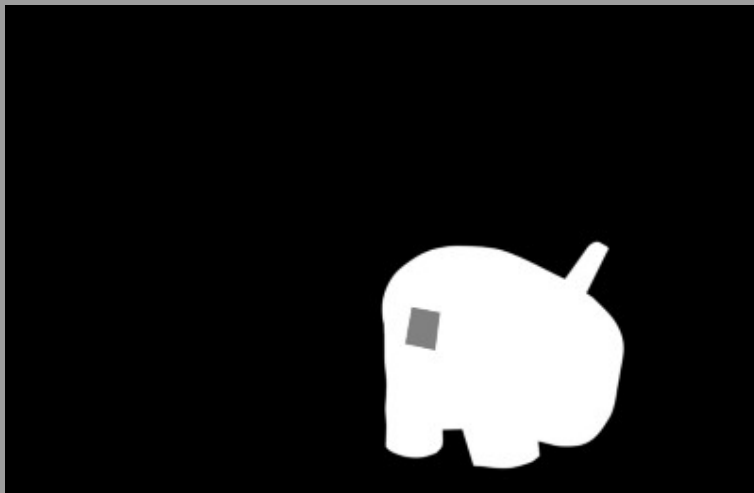
+



=

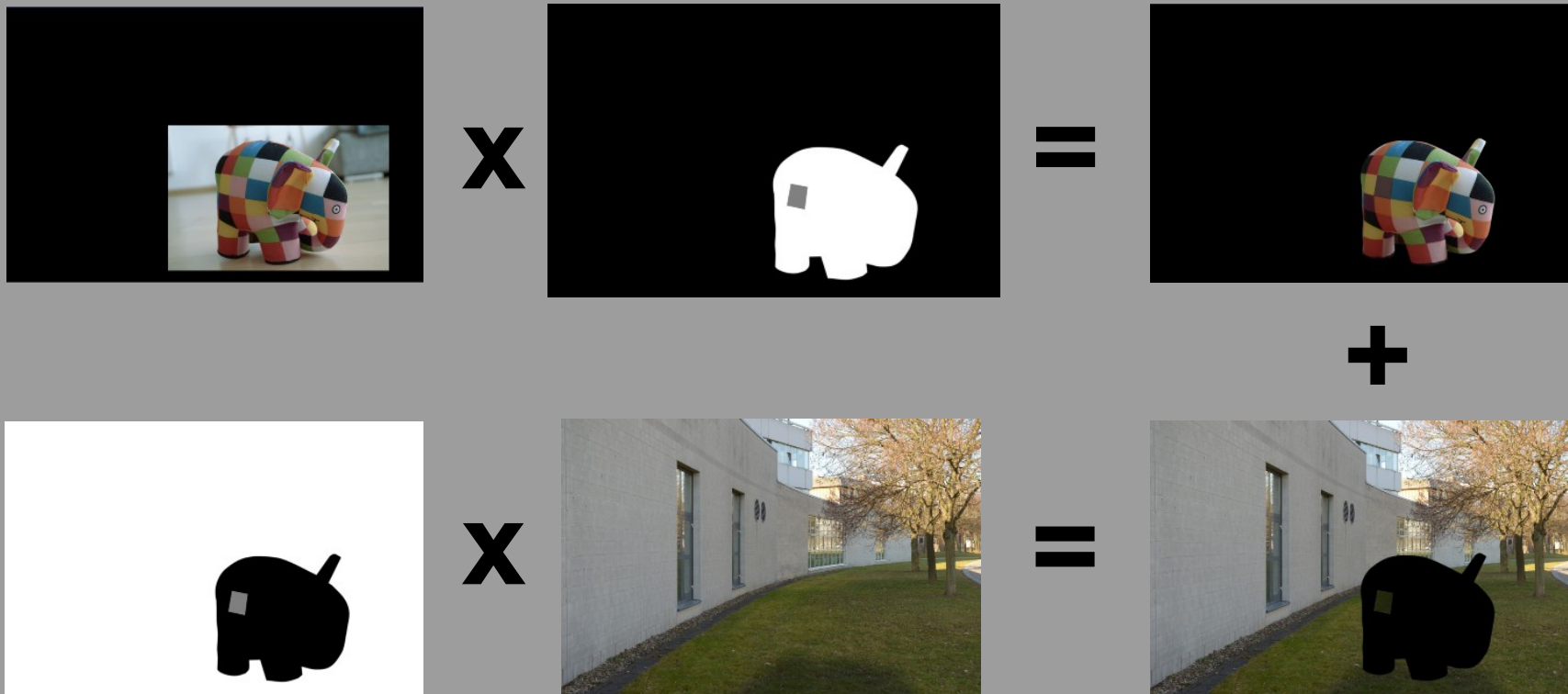


mit



einfaches Compositing: Over-Operator

- Over-Operator: Legt zwei Bilder übereinander, wobei die transparenten Bereiche durch ein drittes Matte-Image kontrolliert werden
- $O = (A \times M) + ((1 - M) \times B)$



einfaches Compositing: Premultiplied Images

- premultiplied Images:
 - erste Multiplikation der Over Operation schon ausgeführt. Viele 3D-Programme geben solche Bilder aus.
 - Konsequenzen: Vorsicht bei Bildern mit halbtransparenten Matten (z.B. softedge). Doppelte Multiplikation wirkt sich dort aus.



keine Multiplikation



Doppelte ausgeführte Multiplikation

einfaches Compositing: Mix-Operator

- Mix-Operator: Blendet zwei Bilder ineinander
- $O = (MV \times A) + ((1 - MV) \times B)$



einfaches Compositing: weitere Operatoren

- Subtract-Operator: Subtrahiert zwei Bilder voneinander
 - $O = A - B$
 - ist nicht symmetrisch , d.h. $A-B \neq B-A$
- In-Operator: Zeigt Bereiche von Bild A, die mit der Matte von Bild B überlappen: $A \text{ in } B$
 - $O = A \times B_\alpha$
- Out-Operator: Zeigt Bereiche von Bild A, die nicht mit der Matte von Bild B überlappen:
 - $O = A \times (1 - B_\alpha)$

einfaches Compositing: weitere Operatoren

- Kombinierte Operatoren z.B. Atop: Beschneidet Bild A entsprechend der Matte von Bild B und setzt es anschließend über Bild B
 - $O = (A \text{ in } B) \text{ over } B$
 - $O = (A \times B_\alpha) + (1 - A_\alpha) \times B$

- Skaliert das Bild 5weiss_1schwarz_4weiss.tif mit dem Reformat-Node auf ein Zehntel seiner Größe. Probiert unterschiedliche Filtermethoden aus. Dazu muss unbedingt der Viewer auf 100% stehen.
- Skaliert Bild katze_klein.jpg um den Faktor 5 und probiert unterschiedliche Filtermethoden: Welche ist am besten geeignet?
- Skaliert Bild katze.jpg um den Faktor 0.25 und probiert unterschiedliche Filtermethoden: Welche ist am besten geeignet?
- Versucht die Störungen in buffel.tga mit einem Filter zu reduzieren, ohne zuviel Schärfe zu verlieren.
- Bildet nur aus merge-Nodes mit den Bildern elmar.jpg, elmar_bg.jpg und elmar_matte.png eine Over-Operation nach.
- Bildet mit einer MergeExpression mit den Bildern kreis.tif und viereck.tif die Operatoren in, out und atop nach. Vergleicht die Resultate, wenn ihr kreis.tif und viereck.tif mit den entsprechenden in Nuke vorgefertigten Nodes verbindet.
- Verbindet hund_mit_alpha.tga und katze_mauer.tga mit einem merge-Node. Warum erscheint nicht das gewünschte Resultat?
- Baut mit einem MergeExpression-Node einen Mix-Operator. Vergleicht das Resultat mit dem Dissolve-Node in Nuke

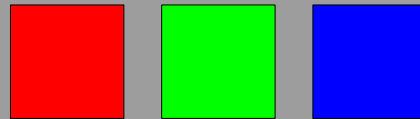
Matte-Images erzeugen

Matte-Images erzeugen

- Manuell erzeugen (Paint)
- Luma Key
- Difference Key
- Chroma Key / Color Difference Key

Matte-Images erzeugen: Luma Key

- Aus RGB-Bild wird monochromes Bild (Luminanzbild) erzeugt. Das menschliche Auge empfindet verschiedene Farben unterschiedlich hell:



einfache Mittelwertbildung aus RGB ($O = (R+G+B)/3$) würde im Bild zu einer für den Menschen falschen Helligkeitsverteilung führen.

Daher unterschiedliche Faktoren für R, G und B:

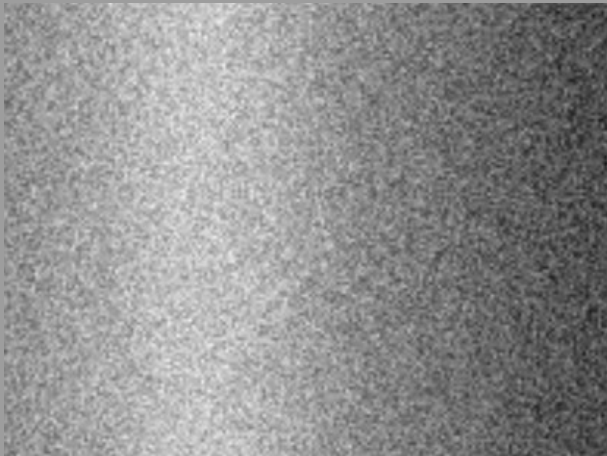
$$O = 0,30 * R + 0,59 * G + 0,11 * B$$

Matte-Images erzeugen: Luma Key

- Für Luma Key Luminanzbild mit eigener „Graumischung“ erzeugen um möglichst großen Helligkeitskontrast zu erreichen -> Raw Matte
- Mit einer Colorcurve wird der Kontrast aufgestellt, bis eine dichte Matte erreicht wird
- Je weniger die Raw Matte aufgestellt wird, desto besser werden die Matte-Kanten

Matte-Images erzeugen: **Bump Mattes**

Bump Mattes werden benutzt, um die Lichter (highlights) aus einem Bild zu isolieren (z.B. für eine Farbkorrektur)



ungleichmäßiges Ausgangsbild



bump matte

Matte-Images erzeugen: Difference Mattes

Bei Difference Mattes benötigt man zusätzlich zum normalen Bild (target plate) ein Bild nur mit dem Hintergrund ohne den Vordergrund (clean plate).



target plate



clean plate

Diese Bilder werden voneinander abgezogen:

$$M = (\text{target} - \text{clean}) + (\text{clean} - \text{target})$$

oder mit Absolutwert:

$$M = \text{abs}(\text{target} - \text{clean})$$

Matte-Images erzeugen: Color Difference Mattes

- Color Difference Mattes:
 - Funktioniert mit einer der drei Grundfarben R,G oder B als Hintergrund.
 - Kommt auch mit weichen, halbtransparenten Kanten zurecht.
 - Ultimatte und Keylight arbeiten nach diesem Prinzip

$$\text{raw matte} = G - \max(R,B)$$

- Erstellt mit dem Expression-Node einen Monochrom-Operator, der aus den drei Farbkanälen den Mittelwert bildet. Baut dann mit einem weiteren Expression Node einen Monochrom-Operator, der das Helligkeitsempfinden des menschlichen Auges berücksichtigt ($0,30 * R + 0,59 * G + 0,11 * B$) und wendet diesen auf das Bild farbtest.psd an. Vergleicht die Ergebnisse mit dem Saturation in Nuke
- Erzeugt ein Luma-Key mit dem Bild cornfield.tif und tauscht den Himmel durch das Bild blauer-himmel-weiße-wolken.jpg. Probiert als Grundlage für den Key unterschiedliche Kanäle und Mischungen der Kanäle aus.
- Isoliert mit Hilfe eines Luma-Keys das Wort „Hallo“ in luma-key-target_grey.tif. Verwendet dazu eine Colorcurve. Versucht dasselbe mit dem Nuke-Lumakeyer.
- Erzeugt eine Bumpmatte mit „grassy slope.tif“. Mit Hilfe der Bumpmatte und eines Grade-Node soll das Grass mit „Frost“ überzogen werden.
- Baut mit Hilfe des Expression-Nodes einen Colordifference-Keyer und wendet diesen auf „greenscreen sequence“ an. Erweitert den Keyer um zwei User Knobs die die Aufsteilung der Raw-Matte erledigen (Stichwort: „Simple Contrast“)

Preprocessing für Color Difference Mattes

- Channel Clamping: Verringern des Grünpegels bzw. Erhöhen des Rot- oder Blaupegels für bestimmte Helligkeitswerte
- Channel Shifting: Anheben oder Senken eines kompletten Kanals

Despill: Entfernen von durch Streulicht hervorgerufenen Farbsäumen.

Um zu einem korrekten Despill-Ergebnis zu kommen müsste man viele Informationen zu jedem Objekt in der Szene und jeder Lichtquelle verarbeiten.

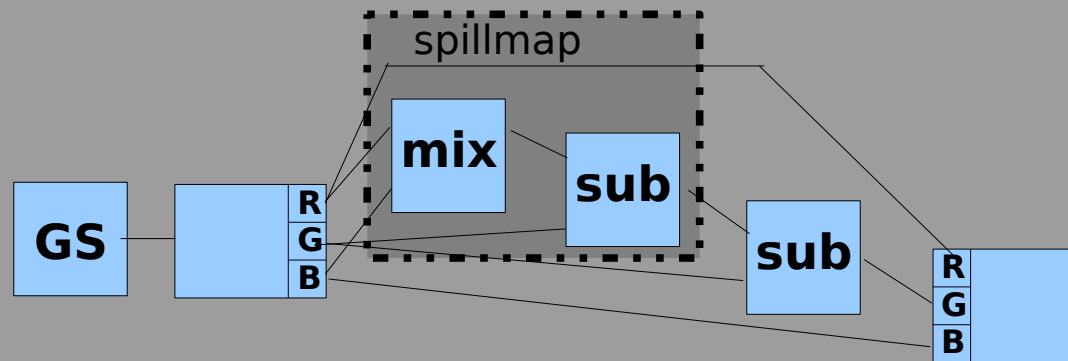
Verwendet werden vereinfachte Modelle, die immer auch zu Artefakten führen: Helligkeitseinbrüche und Farbverschiebungen.

- > viele verschiedene Algorithmen führen je nach Bildmaterial zu unterschiedlich guten Resultaten

- Grün limitiert durch Rot:
 - Grünwerte sollen nie höher als rot sein, wenn sie höher sind sollen sie auf R-Pegel gebracht werden:
 - **despilled green = if $G > R$ then R else G**
 -
- Grün limitiert durch Blau:
 - Grünwerte sollen nie höher als blau sein, wenn sie höher sind sollen sie auf B-Pegel gebracht werden:
 - **despilled green = if $G > B$ then B else G**

- Grün limitiert durch Durchschnitt von Rot + Blau:
 - **despilled green = $g > (r+b)/2 ? (r+b)/2 : g$**
- Beliebige Kombinationen von Rot + Blau
 - z.B. wenn G größer 20% B + 80 % R dann G = 20% B + 80 % R sonst G behalten.
 - Mixvalue Despill:
 - **$g > r*MV + b*(1-MV) ? r*MV + b*(1-MV) : g$**

Die Despill-Operation lässt sich statt mit der if-Anweisung auch über mehrere einfache Math-Operatoren darstellen:



Von Grün wird die Mischung aus Rot und Blau abgezogen. Dabei entsteht die Spillmap. In ihr sind alle Grünanteile des Bildes dargestellt, die über der Mischung aus Rot und Blau liegen. Diese Spillmap wird dann vom Original Greenscreen (GS) nochmals abgezogen.

Despill



Greenscreen (GS)

$G\text{-mix}(R,B)$



Spillmap

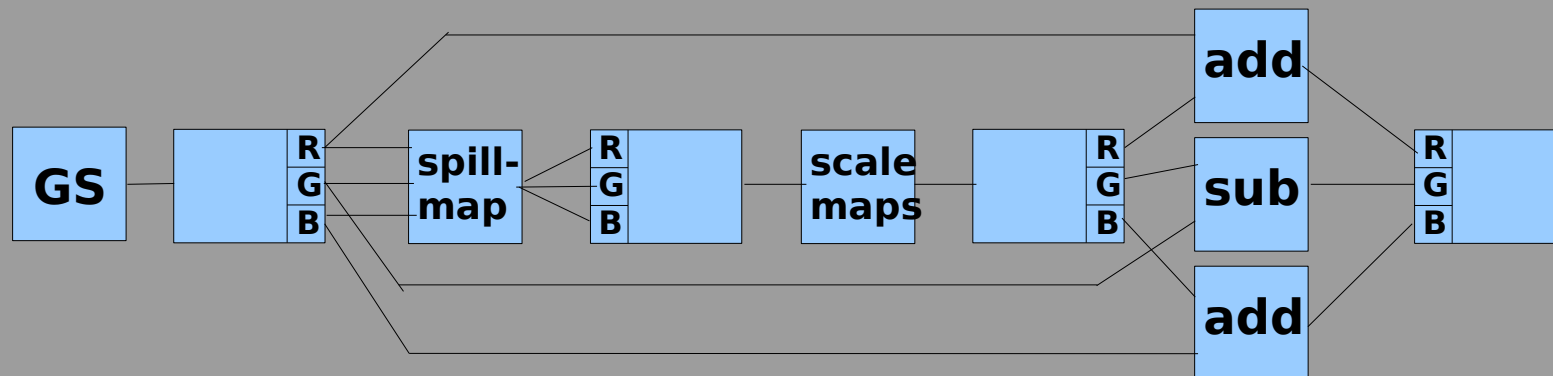
Greenscreen - Spillmap



despilled Image

- „Unspill“

- Beim Unspill wird nicht nur wie beim Despill grün herausgefiltert, sondern auch durch Zufügen von Blau und Rot gegengesteuert. Damit lässt sich die Farbe und Helligkeit des Spills steuern.
- Dazu wird die Despill-Map unterschiedlich skaliert und von G subtrahiert und zu R und B addiert:



Das Ganze lässt sich in Nuke auch in einem einzigen Expression-Node umsetzen:

```

spillmap:      max(g-(r*MV+b*(1-MV)),0)
red channel:   r+r_mult*spillmap
green channel: g-g_mult*spillmap
blue channel:  b+b_mult*spillmap
    
```

Übung 4

- Verbessert den Key aus Übung 3 durch Preprocessing: Channel Clamping / Shifting
- Nehmt die in Übung 3 erzeugte Matte und erstellt damit ein Composite der Frau über einem mittelgrauen Hintergrund (1920x1080 Pixel). Entfernt den Spill in den Haaren mit den Methoden „Grün limitiert durch Rot“, „Grün limitiert durch Blau“ und „Grün limitiert durch eine Mischung von Rot und Blau“. Verwendet dazu jeweils ein Expression Node. Achtet auf Farbveränderungen im Shirt.
- Vergleicht die Resultate mit den Möglichkeiten des HueCorrect-Nodes bei Verwendung der green_sup-Kurve
- Bei den verwendeten Methoden entsteht teilweise ein dunkler Rand. Versucht mit der „Unspill“-Methode den Rand zu verhindern.