

IL COMPOSITING IN BLENDER 2.5

Utilizzare i nodi nella fase di post-produzione



FRANCESCO "RedBaron85" MILANESE

BLENDER FOUNDATION CERTIFIED TRAINER

Il Compositing in Blender 2.5

Utilizzare i nodi nella fase di post-produzione

Francesco “RedBaron85” Milanese
www.redbaron85.com

ebook

SOMMARIO

Premessa	3
Introduzione	4
Nodi di Input e Nodi di Output.....	9
INPUT --- Render Layers	9
NOTA --- Il canale Z	11
INPUT --- Image	11
INPUT --- Texture	11
INPUT --- Value	12
INPUT --- RGB	12
INPUT --- Time	12
OUTPUT --- Composite	12
OUTPUT --- Viewer	13
OUTPUT --- SplitViewer	13
OUTPUT --- FileOutput	14
OUTPUT --- Layers	14
Il Nodo Mix del gruppo Color.....	15
Gli altri Nodi del gruppo Color	19
Z COMBINE	19
ALPHA OVER	19
INVERT	20
GAMMA	20
BRIGHT-CONTRAST	20
HSV (Hue, Saturation, Value)	21
RGB Curves	21
TONE MAP	22
HUE CORRECT	23
COLOR BALANCE	23
I Nodi del gruppo Vector	24
VECTOR CURVES	24
NORMAL	24
NORMALIZE	25
MAP TO	25
I Nodi del gruppo Filter.....	27
FILTER	27
BLUR	27
DIRECTIONAL BLUR	28
BILATERAL BLUR	28
VECTOR BLUR	29

DILATE / ERODE	30
DEFOCUS e la profondità di campo	30
GLARE	31
I Nodi del gruppo Convertor	33
RGB to BW	33
COLOR RAMP	33
SET ALPHA	34
ID MASK	34
MATH	35
ALPHA CONVERT	36
I Nodi del tipo Separate e Combine	37
I Nodi del gruppo Distort	38
TRANSLATE	38
ROTATE	38
SCALE	39
FLIP	39
CROP	39
DISPLACE	40
MAP UV	40
LENS DISTORTION	41
I Nodi del gruppo Matte	42
LUMINANCE KEY	42
COLOR SPILL	42
CHANNEL KEY	43
COLOR KEY	43
CHROMA KEY	43
DIFFERENCE KEY	44
DISTANCE KEY	45
Raggruppamenti e dipendenze cicliche.....	46
Esempi pratici di Compositing con i Nodes	48
Sfocare le ombre con Blur	48
Sfocatura animata	49
L'ubriaco (sfocatura e immagini sdoppiate).	49
Atmosfere più fredde o più calde con le curve RGB.	52
Fornire il bagliore GLARE solo a determinati oggetti della scena	53
Bilateral Blur per i rendering AO	54
Watermarking delle immagini	54
Cambiare, colorandolo, lo sfondo trasparente	54
Rendering Stereoscopico (basi)	55
La copertina di questo ebook	56

Premessa

Il sistema del Compositing mediante Nodes di Blender 3D è uno strumento potentissimo, seppur poco conosciuto, per ottenere con facilità trasformazioni ed effetti speciali lavorando sulle immagini prodotte dal programma e non solo.

Ho realizzato questo ebook basandomi sulla versione 2.5 di Blender. La trattazione riguarda esclusivamente i Nodes del Compositing del Rendering finale, non i Nodes per i Materiali e le Textures.

Questo ebook si rivolge a lettori che hanno già una discreta padronanza degli strumenti e delle operazioni fondamentali di Blender.

Nei tutorials e nei videotutorials pubblicati sul mio sito web ho sempre cercato di parlare sia degli strumenti messi a disposizione dai programmi, per ottenere una conoscenza completa di parametri e funzionalità, sia delle applicazioni pratiche e degli effetti che questi consentono di ottenere.

Questo ebook non fa eccezione: nei vari capitoli vengono esaminati tutti i Nodes del Compositing di Blender 2.5 e mostrati alcuni esempi d'utilizzo degli stessi.

Buona lettura... e buon divertimento!

Catania, Agosto 2010

Francesco “RedBaron85” Milanese
Blender Foundation Certified Trainer

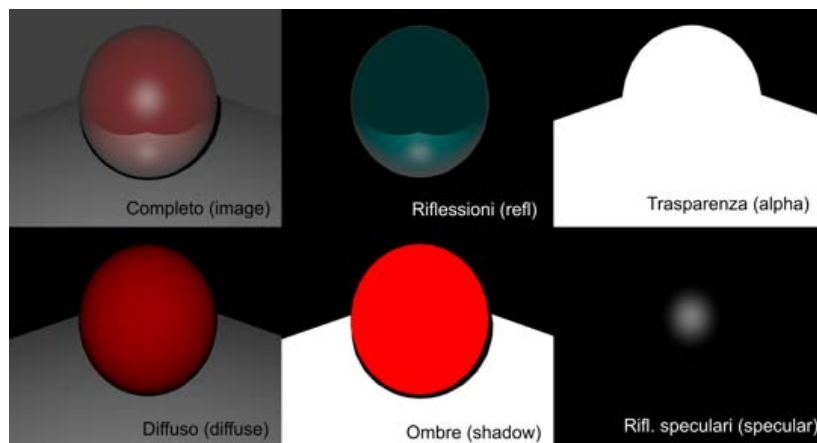
CAPITOLO 1

Introduzione

Quando eseguiamo un **rendering** in **Blender**, l'immagine finale che vediamo a video o che memorizziamo su file è il risultato della **composizione** (*compositing*) di più immagini, ciascuna derivante da un particolare **canale di informazioni**.

Il **rendering** prevede quindi vari **passi** di elaborazione, detti **render passes**, che producono altrettante immagini, una per ciascun canale; sono **render passes**, ad esempio, l'Alpha (la trasparenza), il colore diffuso, i riflessi speculari, le ombre, ecc...

L'immagine seguente mostra il **rendering** di una scena (una sfera rossa con materiale riflettente Mirror a 0.5 e un plane con materiale di base e colore grigio scuro) e le immagini relative ai **canali** diffuso, riflesso, Alpha, ombre e speculare.



Nel **prossimo capitolo** vedremo come impostare i **Nodes** nell'**editor** per poter **salvare su file** le immagini relative ai vari **canali di rendering**.

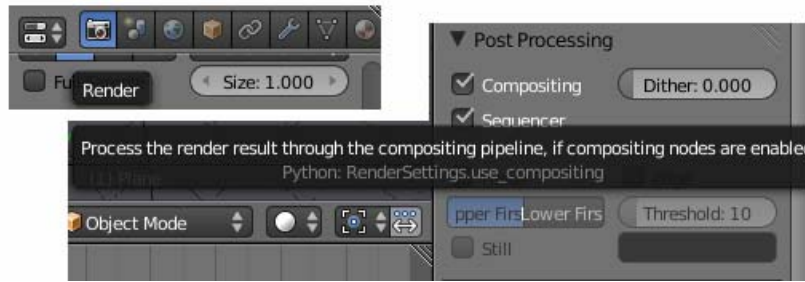
I **Nodes** servono proprio a controllare, modificare e comporre i **render passes** di **Blender** e, quindi, i relativi **canali di rendering**.

Possiamo suddividerli principalmente in **tre tipi**:

- nodi di **input**, sorgenti di dati;
- nodi di **elaborazione** (detti anche filters, **filtri**);
- nodi di **output**, che ricevono dati.

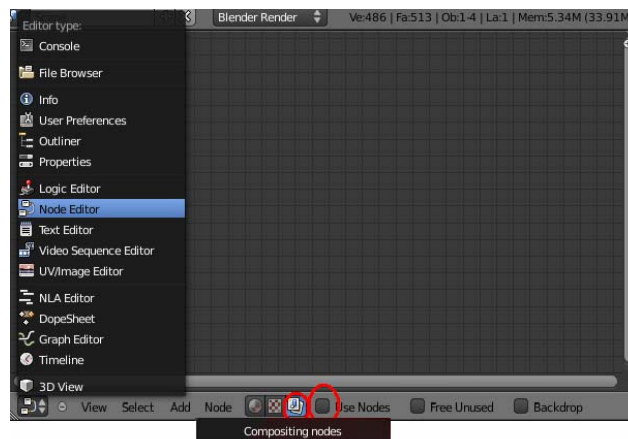
Vediamo quindi come attivare il **compositing** in fase di **rendering** ed accedere all'**editor** dei **Nodes** in **Blender 3D 2.5**.

Per prima cosa, dobbiamo attivare la fase di **Compositing** in **Post-Produzione**; per far ciò, accediamo alla scheda **Render** all'interno della **Properties Window** e, nella sezione **Post-Processing**, selezioniamo **“Compositing”**, come mostrato in figura nella pagina seguente.

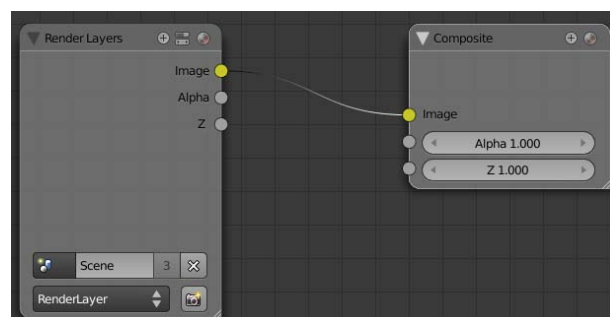


In **Blender 2.53** tale opzione è selezionata di default.

A questo punto dobbiamo cambiare una delle **finestre** in **Node Editor**, selezionare la casella “**Compositing Nodes**” ed attivare l’opzione “**Use Nodes**”, come mostrato in figura.

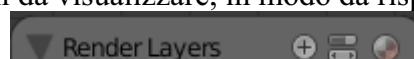


All’interno della finestra **Node Editor** appariranno due **oggetti** collegati da una **curva**: si tratta dei **Nodi Render Layers** e **Composite**, mentre la **curva** che li collega è una **connessione** a tutti gli effetti ed indica che il **canale Image** in uscita da **Render Layers** manda le sue informazioni al **canale Image** di input di **Composite**.



Intuiamo quindi subito che un **Nodo** (sia esso una sorgente di dati, una destinazione o un filtro) è rappresentato da una “casella” nel **Node Editor**, mentre il **flusso** delle informazioni è rappresentato da una **curva**, che identifica una **connessione** tra **porte di input** (quelle sulla sinistra del Nodo) e **porte di output** (quelle sulla destra): l’immagine prodotta da **Render Layers**, nel nostro caso, arriva in input al Nodo **Composite**, che la mostra a video.

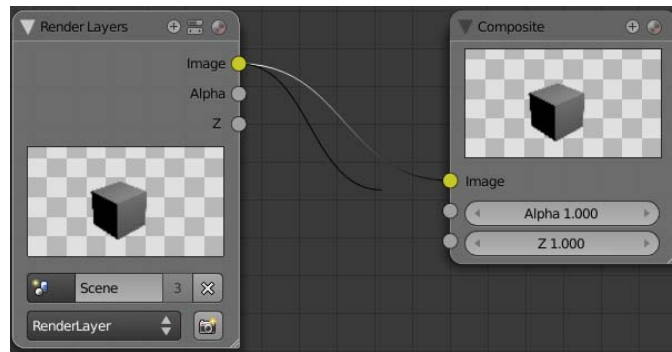
Nella parte superiore di ciascun **Nodo** sono presenti il nome ed altri simboli, che consentono di ridurre la quantità di informazioni da visualizzare, in modo da risparmiare spazio.



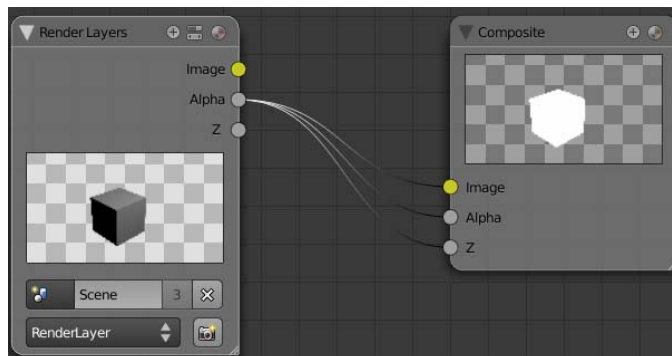
In particolare:

- cliccando sulla **freccia diretta verso il basso ▼** ridurremo il Nodo alla sola **intestazione**;
- cliccando sul **simbolo +** nasconderemo le **porte** di input e output non utilizzate e i relativi **controlli** associati (ad esempio, lo slider Alpha nel nodo Composite);
- cliccando sul **simbolo dei pulsanti e degli slider**, nasconderemo i pulsanti per le **impostazioni** proprie del **Nodo** (ad esempio, le due file di pulsanti in basso nel **Nodo Render Layers**);
- cliccando infine sulla **sfera** nasconderemo la **preview** del risultato (non visibile nell'immagine mostrata precedentemente in quanto non abbiamo ancora effettuato un **rendering** né abbiamo creato o cancellato delle **connessioni**).

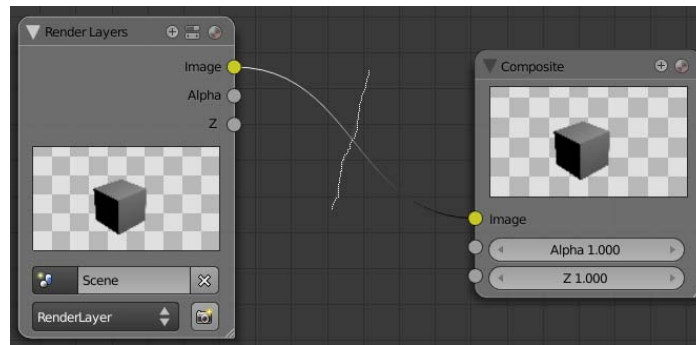
Per **creare una connessione** tra due **porte** è sufficiente fare click col tasto sinistro del mouse su una porta di **Output** (ad esempio, Image in Render Layers), trascinare e rilasciare su una porta di **Input** (ad esempio, Alpha in Composite).



E' possibile associare una **porta di Output** a più **porte di Input**, inviando quindi la stessa informazione a più **porte**, anche di **Nodi** differenti, ma non è possibile collegare più **porte di Output** ad una stessa **porta di Input**.



Per **cancellare una connessione** possiamo fare click col tasto sinistro del mouse su una porta di input, oppure premere il tasto **CTRL** e il **tasto sinistro del mouse** e, tenendoli premuti, **trascinare** il mouse verso il basso “tagliando” una connessione (il cursore del mouse assumerà la forma di una matita e, muovendolo, disegneremo una linea), per poi rilasciare sia CTRL che il tasto sinistro del mouse.



Per **spostare** un **Nodo** sul **Node Editor** è sufficiente fare **click** col tasto sinistro del mouse sul suo **header** e **trascinare** per poi **rilasciare** nella posizione desiderata, oppure **selezionarlo** (con un singolo click del mouse) e premere **G** per spostarlo muovendo il mouse, un po' come avviene con **oggetti** e **vertici** in **modo oggetto** e in **modo edit** in **Blender**.

Per **cancellare** un **Nodo**, è sufficiente **selezionarlo** e premere **X** o **CANC**, mentre per **duplicarlo** (creando così un secondo **Nodo**, identico al primo) basta **selezionarlo** e premere **Shift-D**.

Nodi e **collegamenti** possono essere selezionati anche con le **maschere rettangolari** (shortcut: **B**) o **selezionando e deselegionando tutto** con il tasto **A**.

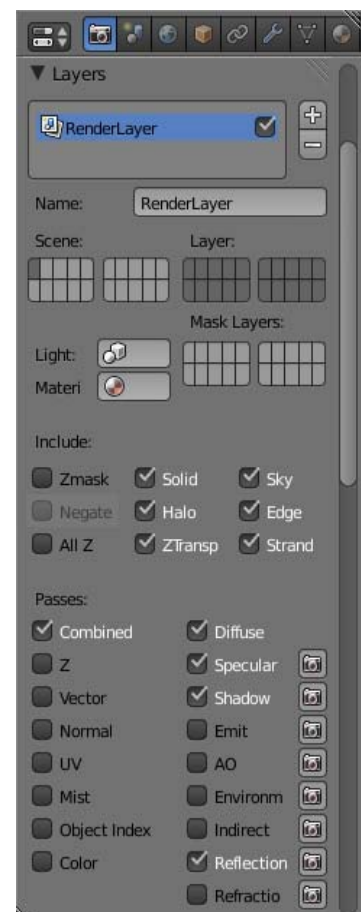
Queste ed altre operazioni di **selezione** dei **Nodi** (anche in base ai **collegamenti**, con **Select Linked** nelle sue varianti) sono disponibili, comunque, nei **menù Select** e **Node** presenti nell'**header** della finestra **Editor dei Nodi**.

Prima di procedere con la descrizione dei **Nodi** e le loro applicazioni pratiche, è bene precisare alcune cose sul **processo di Compositing**.

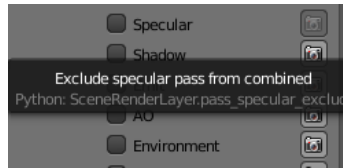
Come sicuramente avrete notato, la **sorgente** di base delle informazioni è il **Nodo Render Layers**: esso contiene le **porte** associate ai vari **render passes** e, quindi, alle varie **informazioni** che contribuiscono a formare l'**immagine finale renderizzata**; in particolare, **Image** contiene **tutte** le informazioni di **tutti** i **canali** abilitati per il **rendering** finale.

Le **porte** presenti di default per **Render Layers**, comunque, non sono tutte le porte disponibili per tale **Nodo**: per poter inserire le altre **porte** – e, quindi, **accedere agli altri canali** – dobbiamo **selezionare** le relative voci nella sezione “**Layers - Passes**” della scheda **Render**, all'interno della **Properties Window**, come mostrato nella **figura** a destra.

Adesso però dobbiamo fare una **distinzione** importante: anche se, di default, queste voci non sono abilitate, ciò non significa che non saranno presenti nel **rendering “Combined”**, in **uscita** dalla porta **Image** di **Render Layers**, infatti ogni volta che lanciamo un **rendering** effetti come i riflessi, le ombre, le trasparenze ecc... sono sempre presenti, senza aver bisogno di selezionare tali opzioni in “**Layers Passes**”; selezionando o deselegionando queste caselle, le inseriremo o le toglieremo **solo dall'insieme delle porte di Render Layers**.



Per poter **escludere** uno di questi **canali** sempre presenti, come i riflessi speculari (Specular), dal **rendering** calcolato per **Image** di **Render Layers**, invece, dobbiamo cliccare sul simbolo della **macchina fotografica** posto accanto al **canale** da **escludere**, sempre nella scheda **Layers Passes**, come mostrato in figura sotto.



Fatte queste premesse, possiamo passare al **secondo capitolo** ed analizzare in dettaglio i **Nodi** di **input** e di **output**.

* * *

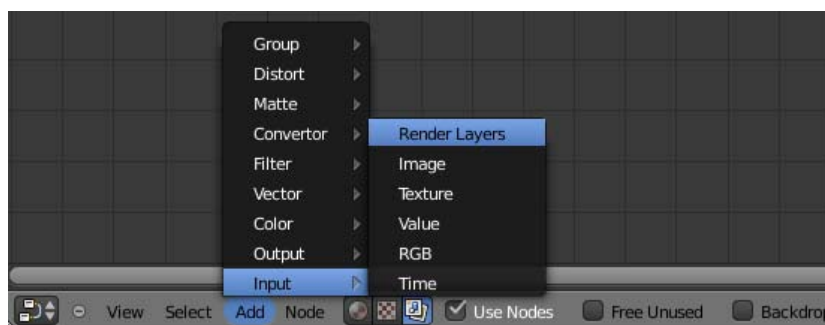
CAPITOLO 2

Nodi di Input e Nodi di Output

Blender raggruppa i **Nodi** in “gruppi”, a seconda della loro **funzionalità**.

In questo capitolo prenderemo in esame i **Nodi** di **Input**, ossia quelli che **forniscono informazioni**, e i **Nodi** di **Output**, che le **ricevono**, tralasciando per il momento i **Nodi Filtri**, che **elaborano** le informazioni.

Per aggiungere un **Nodo** al **Node Editor**, clicchiamo su **Add** nell’header della finestra e scegliamo un **Nodo** all’interno di un **gruppo**, come mostrato in figura.



Iniziamo con l’esaminare i **Nodi** del gruppo **Input**, a partire da **Render Layers**, già visto nel **capitolo precedente**.

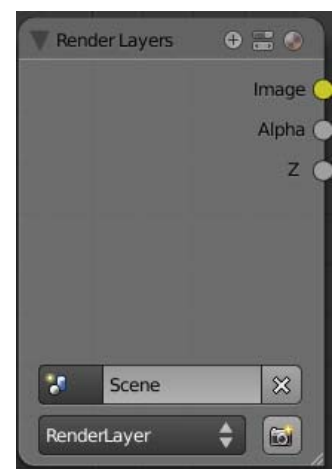
INPUT --- Render Layers

Come anticipato, il nodo **Render Layers** contiene le informazioni riguardanti una **scena** e un **Render Layer** del file **.blend** aperto.

E’ possibile scegliere quale **scena** utilizzare e, all’interno della stessa, quale **Render Layer** prendere in considerazione attraverso i due **selettori** posti in basso nel **Nodo Render Layer**.

La scena viene **renderizzata** dal punto di vista della **telecamera attiva** presente in essa e le informazioni relative ai vari **canali** (colore diffuso, riflessi speculari, ecc...) possono essere recuperate e manipolate grazie alle **porte di output**, come discusso nel **capitolo precedente**.

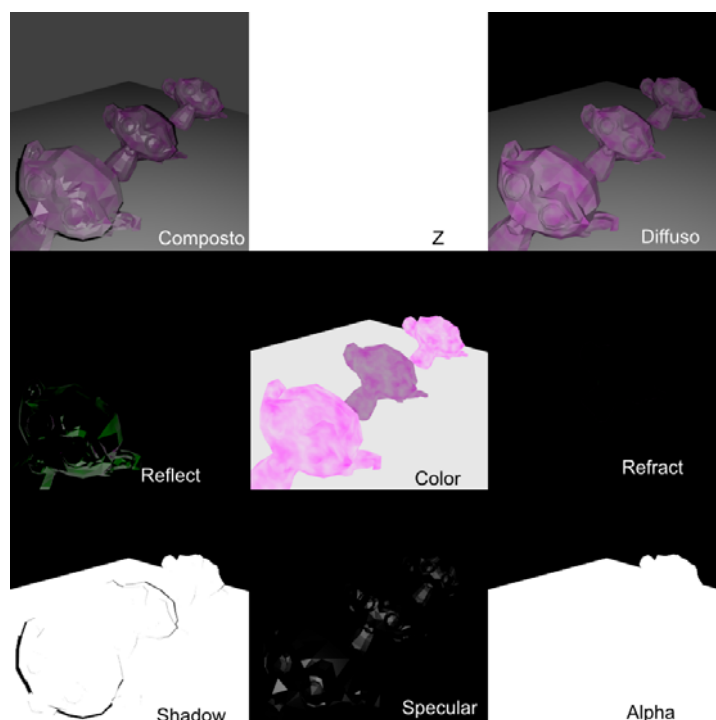
E’ possibile quindi combinare le **immagini** derivanti da due o più **Render Layers**, anche di scene differenti, utilizzando **filtri** (ad esempio **Mix**), o realizzare altri effetti che prevedano l’utilizzo di uno o più **Render Layers** contemporaneamente.



Prendiamo quindi in esame i vari **canali di output** presenti:

- **Image** --- L'immagine ottenuta considerando tutti i Render Passes non esclusi dal processo di rendering nella sezione "Layers – Passes" nella scheda Render della Properties Window.
- **Alpha** --- Il valore di trasparenza di un pixel.
- **Z** --- La distanza di un pixel dalla telecamera (vedi nota in seguito).
- **Normal** --- Definisce come il colore di un pixel viene influenzato dalla luce in arrivo.
- **Color** --- Il colore "vero e proprio" di un oggetto, indipendentemente dall'illuminazione della scena (privo di ombreggiatura, piatto).
- **Diffuse** --- Il colore, l'aspetto dell'oggetto considerando l'illuminazione.
- **Specular** --- I riflessi speculari.
- **Shadow** --- Le ombre proiettate dagli oggetti su altri oggetti (escludendolo dal rendering, non ci saranno oggetti in ombra).
- **AO** --- Il "peso" dell'effetto Ambient Occlusion (se presente) sul rendering finale.
- **Reflect** --- Le riflessioni.
- **Refract** --- La rifrazione dei raggi luminosi che passano attraverso oggetti (semi)trasparenti.
- **Index OB** --- Un indice numerico (un intero) associato a ciascun oggetto della scena in base al punto di vista della telecamera attiva.
- **Emit** --- Per i Materials con Emit maggiore di 0, indica l'emissione del materiale.
- **Environment** --- Canale associato alla luce ambientale.
- **Indirect** --- Canale associato all'illuminazione indiretta.
- **UV** --- Canale associato alle textures mappate mediante tecnica di UV Mapping.
- **Mist** --- L'effetto Nebbia / Foschia proprio di Blender.
- **Vector** --- Noto anche come Speed (fino a Blender 2.49), indica la velocità e la direzione con le quali si sta muovendo un oggetto.

L'immagine seguente mostra il **rendering** di una **scena** (tre oggetti **texturizzati** con **Clouds** su un piano, con l'oggetto nel mezzo semitrasparente **Raytrace** e l'oggetto più vicino con **Mirror Reflectivity** a 0.5) e le immagini relative ai **canali** Z, Alpha, Diffuse, Color, Reflect, Refract, Shadow e Specular.

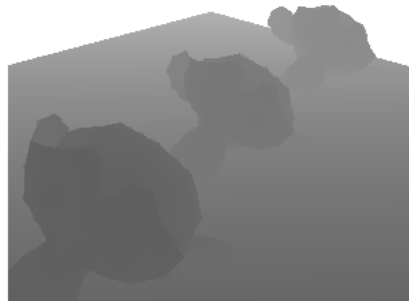


NOTA --- Il canale Z

L'immagine ottenuta mediante il **canale Z** mostra un'immagine completamente **bianca**, anche se **Z** indica la **distanza** dei vari **pixel** dalla **telecamera** e, quindi, vista la **scena** presa in considerazione, non dovrebbe produrre un'immagine **bianca**...

... in realtà, quell'immagine NON è bianca, ma le **informazioni** non sono visibili: il **canale Z** memorizza infatti i **dati** sotto forma di valori a **32 bit**, per cui è necessario convertire tali **dati** in valori a **24 bit** per poterli visualizzare correttamente.

Un buon formato per salvare l'immagine originale in **32 bit** è **OpenEXR**, mentre con un buon programma di **fotoritocco** possiamo procedere alla **conversione in 24 bit** ed ottenere, ad esempio, l'**immagine in basso a destra** (quella in basso a sinistra, completamente bianca, è l'originale salvata come file **PNG** mediante un **Nodo Output File**).



INPUT --- Image

Con tale Nodo è possibile **caricare un'immagine** ed utilizzarla all'interno del **Compositing**, ad esempio come **mappa** per l'opacità, lo sfondo o il canale Diffuse.

L'**immagine** può essere caricata da **disco**, dall'insieme delle immagini caricate in **UV-Editor delle immagini** (attraverso l'icona in basso a sinistra nel **Nodo**, "Browse ID Data"), da un **filmato** o da una **sequenza** (in questi ultimi due casi, grazie a dei pulsanti che appariranno sul fondo del Nodo, sarà possibile scegliere anche il range dei frames da utilizzare).



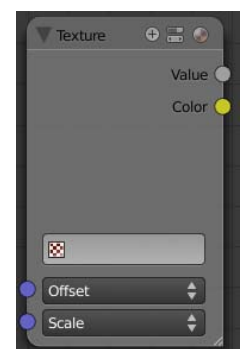
E' possibile anche recuperare le informazioni relative al solo **canale** della **trasparenza** (se previsto) del file **immagine**, attraverso la porta di output "**Alpha**", o quelle del **canale Z** (se previsto, altrimenti verrà considerato **Z uniforme** pari a distanza 1.0 dalla telecamera).

INPUT --- Texture

Texture è molto simile ad **Image**, almeno concettualmente: consente di utilizzare un'immagine, o meglio una **Texture** (quindi anche **procedurale**), tra quelle presenti nel set di **Texture** del file **.blend** utilizzato.

In questo caso è possibile regolare anche l'**offset** di spostamento della **Texture** e le sue **dimensioni**, rispettivamente mediante i selettori **Offset** e **Scale**.

Tali valori sono provvisti di **porte di input** (oltre a controlli **numerici** che consentono di impostare un **valore** ben preciso) perché il **valore** potrebbe

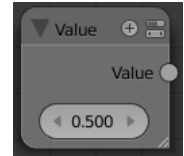


dipendere dall'**output** di altri **Nodi**, ad esempio **Nodi ColorRamp**, come vedremo.

Le **porte di output** in questo caso sono due: **value** (in bianco e nero, dipendente dalla **trasparenza Alpha** della **Texture**) e **color** (il **colore** della **Texture**).

INPUT --- Value

Questo **Nodo** restituisce un **valore numerico** (nel range 0.0 – 1.0) che può essere utilizzato come valore in input **costante** per alcune voci di altri **Nodi**.



INPUT --- RGB

Questo **Nodo** fornisce un **colore**, scelto mediante l'apposito strumento di **selezione colore** presente all'interno del **Nodo**.

E' possibile scegliere anche il grado di **trasparenza** (Alpha) del **colore**.

Il **colore** così ottenuto può essere utilizzato, mediante **filtri** (come ad esempio **Mix**) per sommare una tinta uniforme ad una texture, allo sfondo, ecc...

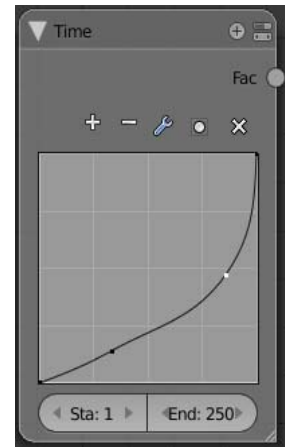


INPUT --- Time

Il **Nodo Time** restituisce, durante l'animazione, un **valore numerico** (nel range 0.0 – 1.0), che dipende dal **valore** assunto da una particolare **curva**, definita appunto nel **Nodo**, in ciascun **frame** dell'animazione.

Come è visibile in figura a lato, è possibile definire il **range** dei **frames** e l'andamento della **curva**, dal quale dipenderà anche il **valore** restituito dal **Nodo**.

Si tratta di un buon metodo per ottenere, ad esempio, delle **dissolvenze incrociate**, facendo variare il valore di **miscelazione** tra due immagini, textures o altro da 0 a 1 nell'arco di un certo numero di **frames**.



Passiamo all'insieme dei **Nodi di Output**.

OUTPUT --- Composite

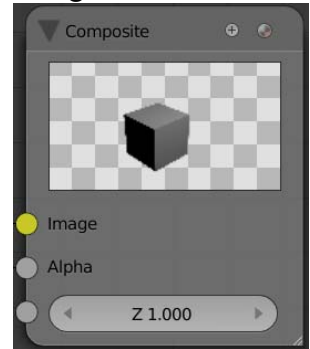
Il **Nodo Composite** rappresenta, sostanzialmente, il **rendering** vero e proprio: i **dati** in arrivo sulle **porte di input** verranno utilizzati per produrre l'**immagine finale renderizzata**, "composta".

L'**aggiornamento dell'anteprima** del risultato, visibile nel **Nodo**, viene fatto **automaticamente** da **Blender** ad ogni **rendering** o ad ogni **modifica** dell'insieme dei **Nodi** o delle loro **connessioni** nell'**editor**.

Chiaramente, se si effettua un **rendering** con la modalità “Compositing” attiva in Post-Processing e “Use Nodes” nel Node Editor, è necessario avere un **Nodo** di questo tipo connesso con qualche informazione nella **finestra Node Editor**.

Il **Nodo Compositing** è provvisto di tre porte di **input** per i **canali Image** (immagine vera e propria in RGBA), **Alpha** (mappa della trasparenza) e **Z** (informazioni sulla distanza dalla telecamera).

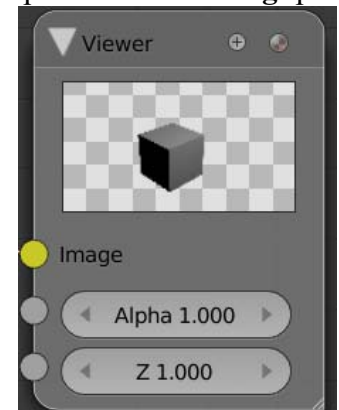
Com'è noto, per effettuare il **rendering** di una singola **immagine** si può ricorrere allo **shortcut F12** e **salvare** il risultato con **F3**, ma per realizzare una **sequenza di immagini** in un certo **range di frames** bisogna ricorrere al tasto **ANIMATION** all'interno della scheda **Render** nella **Properties Window**.



OUTPUT --- Viewer

Il **Nodo Viewer** è, come suggerisce il nome, un **visualizzatore** di anteprime di **rendering**: può essere inserito all'interno dell'insieme dei **Nodi** per visualizzare, nella finestra di **anteprima** presente nel **Nodo**, il **risultato “temporaneo”** di un'elaborazione senza dover procedere necessariamente al **rendering**.

E' provvisto delle **porte di input** per le **informazioni** relative all'immagine in RGBA (**Image**), alla trasparenza (**Alpha**) e alla distanza dalla telecamera (**Z**).

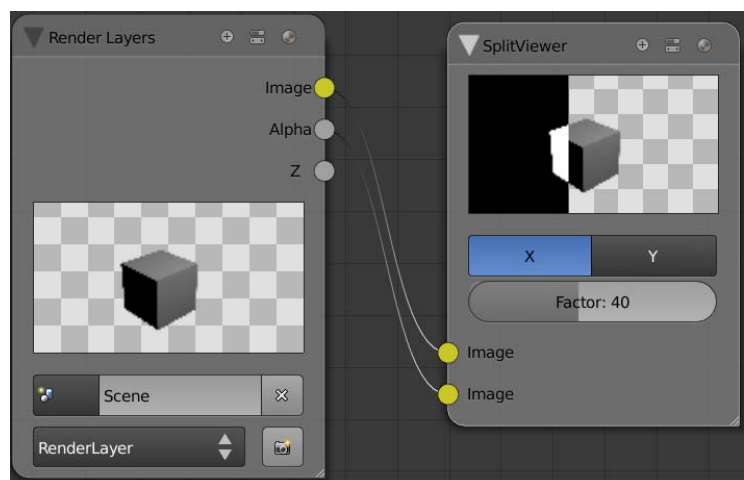


OUTPUT --- SplitViewer

Il **Nodo SplitViewer** è un **visualizzatore** di anteprime, come **Viewer**, ma prende in input **due immagini differenti** (in arrivo, ad esempio, da due diversi **canali** o **percorsi di compositing**, come mostrato in **figura**), visualizzandone una **porzione** nella finestra di **anteprima**.

La **divisione** viene fatta, di default, **verticalmente**, ma è possibile passare ad una divisione **orizzontale** selezionando il pulsante **Y** all'interno del **Nodo**.

Sempre all'interno di **SplitViewer** è presente un altro controllo, **Factor**, che determina in che misura (in **percentuale**) debba essere visualizzata **l'immagine** del primo **canale** rispetto a quella del secondo.

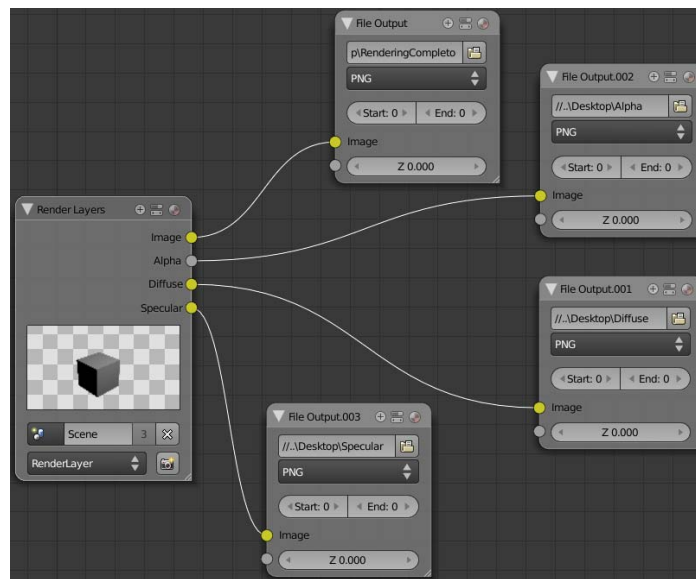


OUTPUT --- FileOutput

Il **Nodo File Output** consente di **salvare su file**, sotto forma di **immagini** o sequenze di immagini (in un range di animazione specificato con **Start** ed **End Frame**, all'interno del **Nodo**), le informazioni in arrivo sui **canali Image** e **Z**.

E' possibile scegliere tra diversi **formati**, come BMP, JPEG, PNG, OpenEXR, ecc...; per alcuni di questi, sono disponibili anche **controlli aggiuntivi**, propri del formato.

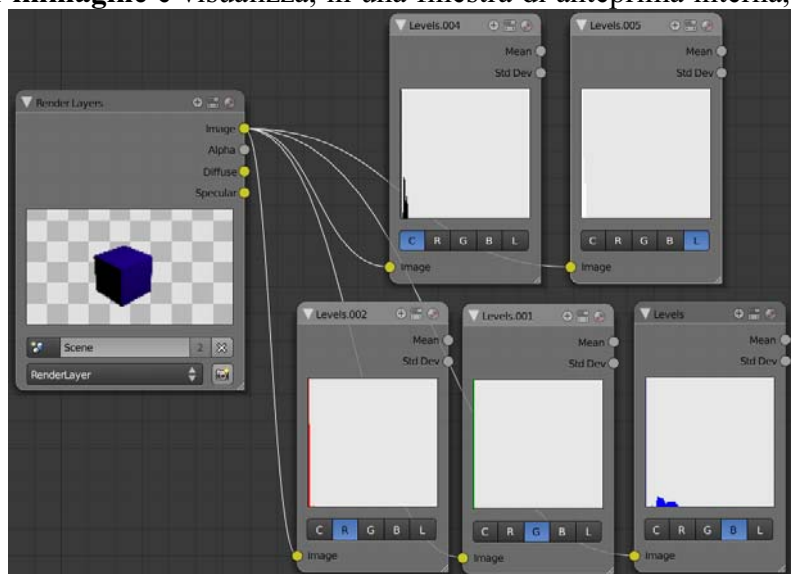
Con **File Output** è possibile quindi **salvare su file** singoli **canali** o **passi intermedi** del processo di **Compositing**; ad esempio, per salvare – oltre al **rendering** – le immagini dei **canali** Alpha, Diffuse e Specular di un Render Layer, è sufficiente configurare il **Node Editor** come mostrato in figura sotto.



OUTPUT --- Layers

Il **Nodo Levels** prende in input un'immagine e visualizza, in una finestra di anteprima interna, l'**istogramma dei valori** per un determinato **canale** (Composito, Rosso, verde (G), Blu e canale della Luminanza) dell'immagine.

E' provvisto anche di due **porte di output**, **Mean** e **Std Dev**, che restituiscono rispettivamente il valore medio e la deviazione standard (scarto quadratico medio) dell'istogramma calcolato per il **canale** selezionato.



* * *

CAPITOLO 3

Il Nodo Mix del gruppo Color

Ora che abbiamo preso in esame i **Nodi** di input e di output, possiamo iniziare la trattazione dei **Nodi** intermedi, o “**filtri**”, che consentono l’**elaborazione** delle informazioni e, quindi, il **Post-Processing** vero e proprio.

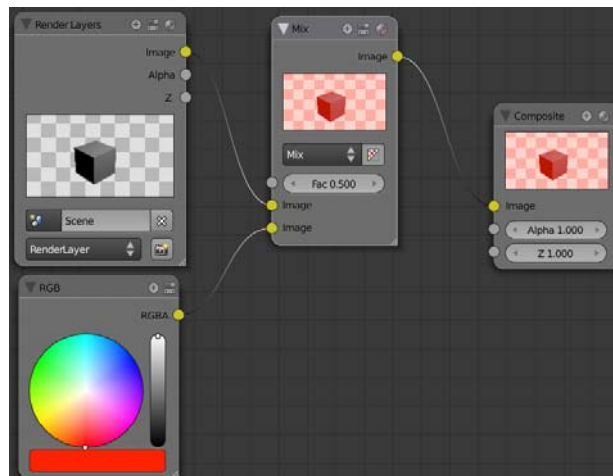
Iniziamo con il **Nodo Mix**, presente all’interno del gruppo **Color**, che mette a disposizione diverse funzioni per “**miscelare**”, come suggerisce il nome, due informazioni diverse in input per produrre un’unica **immagine in output**; si tratta, in effetti, di uno dei **Nodi** utilizzati con maggiore frequenza in fase di **Compositing**.

Per aggiungere un **Nodo Mix** al **Node Editor** è sufficiente cliccare su **Add → Color → Mix**.

Aggiungiamo anche un altro **Nodo** di input, oltre a **Render Layers**, ad esempio **RGB** con colore **rosso** e Alpha a 1, in modo da provare gli **effetti** di miscelazione di **Mix**.

Collegiamo l’**output** di **Mix** all’input **Image** di **Composite**.

Come visibile nella **figura** in basso, con queste impostazioni (**Mix** è la modalità di miscelazione di default, **Factor** 0.5 è il fattore di miscelazione predefinito) aggiungeremo una **tinta rossa uniforme** a tutta l’immagine finale da renderizzare, in quanto stiamo miscelando i due elementi con un valore di **mixaggio** pari a **0.5**; portando **Factor** a 0 o 5, varieremo l’entità della tinta rossa sullo sfondo rappresentato dall’output del **Render Layer**.

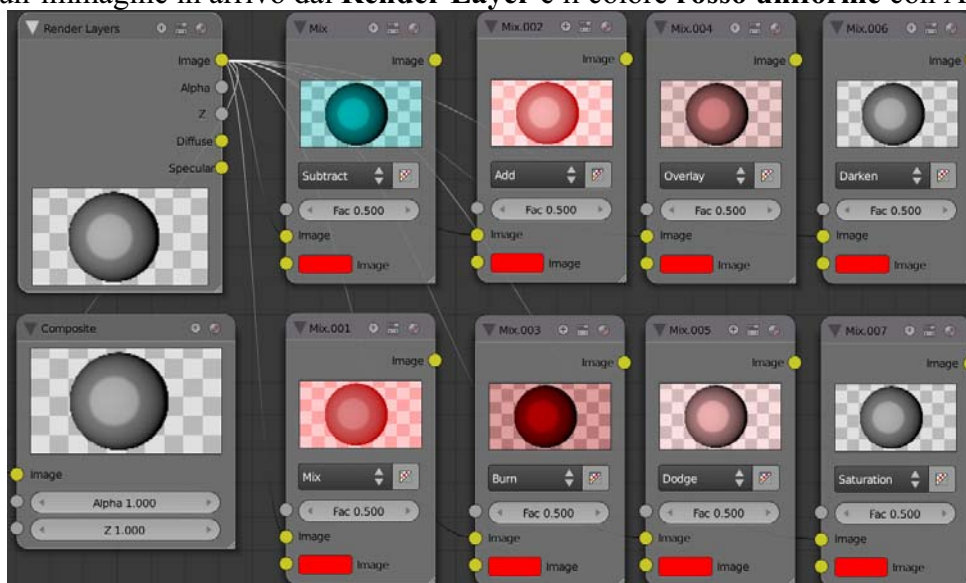


Aprendo il menù a tendina del **tipo di filtro**, in **Mix**, possiamo scegliere tra le diverse **modalità di miscelazione** messe a disposizione da **Blender 2.5**:

- **Mix** --- La prima immagine viene coperta, pixel per pixel, dalla seconda, in base al Factor e ai valori della trasparenza (Alpha).
- **Hue** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Tonalità (Hue) dei due pixel vengono mescolati, il risultato viene combinato con Intensità (Value) e Saturazione del primo pixel e riconvertito nello spazio dei colori RGB.
- **Saturation** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Saturazione dei due pixel vengono mescolati, il risultato viene combinato con Intensità (Value) e Tonalità (Hue) del primo pixel e riconvertito nello spazio dei colori RGB.

- **Value** --- I valori dei pixel vengono considerati nello spazio colore HSV; in questo caso, i valori di Intensità (Value) dei due pixel vengono mescolati, il risultato viene combinato con Tonalità (Hue) e Saturazione del primo pixel e riconvertito nello spazio dei colori RGB.
- **Color** --- Aggiunge un colore ad un pixel, in modo da fornire una determinata tinta a tutta l'immagine.
- **Burn** --- Scurisce la prima immagine basandosi sul gradiente fornito con la seconda immagine di input; il risultato generale è, quindi, un'immagine più scura.
- **Dodge** --- Schiarisce la prima immagine basandosi sul gradiente fornito con la seconda immagine di input; il risultato generale è, quindi, un'immagine più chiara.
- **Linear Light** --- Una combinazione di Burn e Dodge, a seconda del colore di base.
- **Lighten** --- Si fa un confronto pixel per pixel e si prende, dei due, quello più chiaro, applicandolo all'immagine finale.
- **Darken** --- Si fa un confronto pixel per pixel e si prende, dei due, quello più scuro, applicandolo all'immagine finale.
- **Difference** --- Da non confondere con Subtract, in quanto qui viene preso il valore assoluto della differenza, per cui il nero indica colori uguali, bianco colore opposto e i valori intermedi sono una "misura della distanza" pixel per pixel tra le due immagini.
- **Divide** --- Divide la prima immagine per la seconda, pixel per pixel e canale per canale; in caso di secondo pixel a 0.0 (nero), per evitare la divisione per 0 (impossibile) Blender lascerà il valore del pixel della prima immagine.
- **Overlay** --- Una combinazione di Screen e Multiply, a seconda del colore di base.
- **Screen** --- I valori dei due pixel vengono invertiti, moltiplicati tra di loro e il risultato viene a sua volta invertito.
- **Multiply** --- Moltiplica la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.
- **Subtract** --- Sottrae la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.
- **Add** --- Somma la seconda immagine, pixel per pixel e canale per canale, alla prima, "pesando" l'effetto con il controllo Factor. Maggiori dettagli sotto.

L'immagine seguente mostra i **risultati** ottenuti con diversi filtri **Mix** con **Factor** a 0.5 utilizzando un'immagine in arrivo dal **Render Layer** e il colore **rosso uniforme** con Alpha a 1.



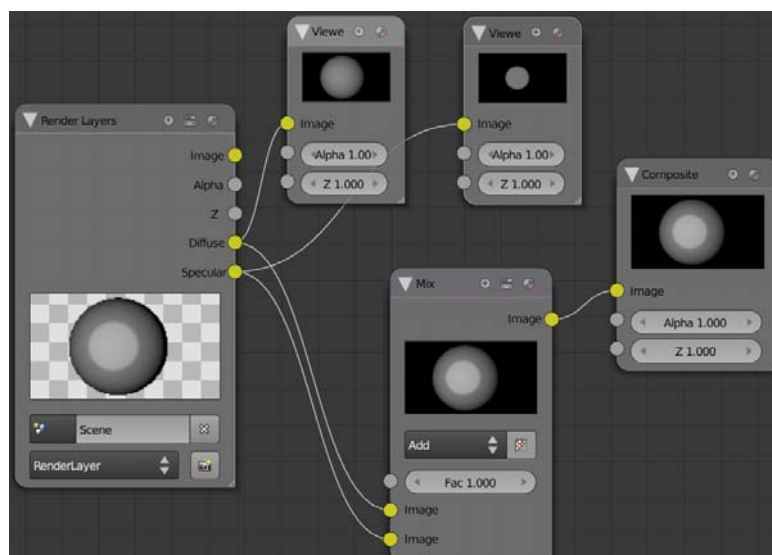
Spendiamo ora qualche parola per definire come operano i **filtri "aritmetici"** come **Divide**, **Multiply**, **Subtract** e **Add**.

Come è noto, ciascun **pixel** porta con sé le informazioni sul **colore** memorizzando quattro valori: **R** (rosso), **G** (verde), **B** (blu) e **A** (Alpha, la trasparenza); il **range di valori** va da 0.0 (nullo) a 1.0 (totale).

I **filtri** lavorano “**per pixel**” e “**per canale**”, ossia modificando il valore di ciascun **pixel** e, per ciascun **pixel**, prendendo i vari **canali** singolarmente, durante il processo di **elaborazione**; nel caso dei **filtri aritmetici**, vengono effettuate proprio le **operazioni** di moltiplicazione, sottrazione e addizione dei valori, con alcune conseguenze:

- **moltiplicare** due valori minori di 1, come nel nostro caso, porta ad un risultato che sarà 1 solo con due 1 in input (infatti, ad esempio, $0.5 \times 0.5 = 0.25$, mentre $0.25 \times 1.0 = 0.25$), per cui in generale l'effetto di Multiply sarà quello di scurire il pixel e, quindi, l'immagine in output; moltiplicare infine un qualsiasi valore per 0.0 (pixel nero) produrrà in output un pixel nero;
- **sommare** due valori con Add produrrà sempre un pixel più chiaro dei due originali (a meno che non siano entrambi a 0.0); il valore massimo è 1.0, per cui qualora la somma dovesse superare tale limite (es.: $0.5 + 1.0$), Blender imporrà come valore finale 1.0;
- **sottrarre** un valore ad un altro con Subtract porta, chiaramente, ad un valore più basso, quindi a pixel e immagini più scure; vale la considerazione del limite fatta per Add, ma verso il basso: valori minori di 0.0 (es.: $0.5 - 1.0$) verranno comunque posti a 0.0 da Blender;
- **dividere** un numero minore di 1 per un altro può portare sia a valori minori di 1 (es.: $0.6 / 0.8 = 0.75$) che maggiori di 1 (es.: $0.5 / 0.2 = 2.5$), ma in ogni caso valgono le regole di troncamento dei valori a 1.0 viste per Add.

Vediamo un **esempio** pratico di **addizione** di valori miscelando tra loro i canali **Diffuse** e **Specular** di un'immagine, impostando il **Node Editor** come visibile in **figura** sotto, in modo da ottenere un **rendering** fatto esclusivamente da questi due **canali** (per accentuare l'effetto del riflesso **Speculare**, ho utilizzato un ombreggiatore **Toon** per **Specular**).

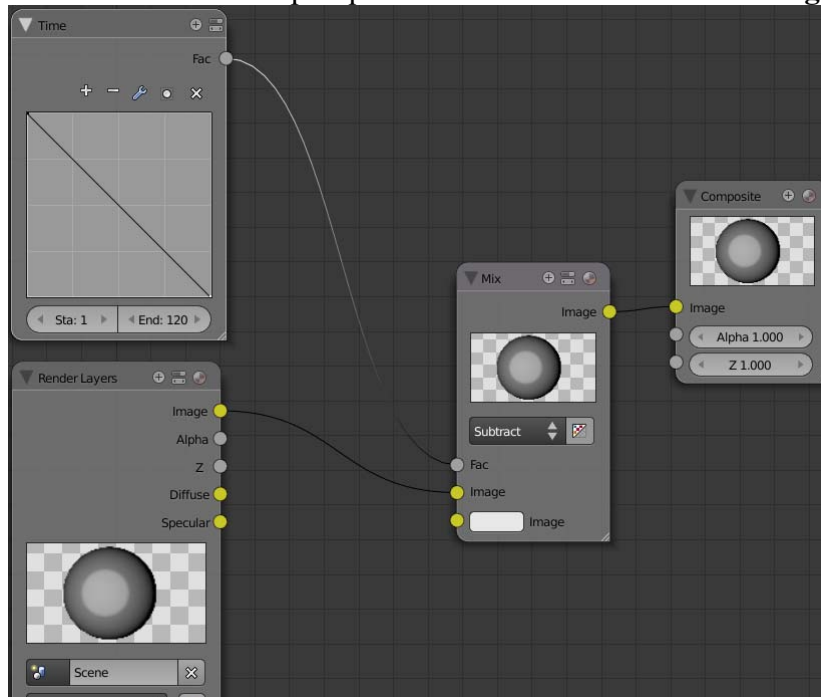


Nel **Node Editor** ho inserito anche due **Nodi Viewer**, collegandoli a **Diffuse** e **Specular** di **Render Layers**, in modo da valutare singolarmente i contributi di informazione dati dai due **canali**.

Torniamo alla trattazione generale del **Nodo Mix**.

Come avrete notato, anche per **Factor** è presente una **porta di input**: ciò significa che il **valore di miscelazione** non deve essere per forza **costante** per tutta l'immagine o per tutta l'animazione, ma può **variare**, ad esempio, in base ad una **mappa**, una **ColorBand** o al numero di **frame** corrente; vediamo subito come far variare **Factor** da 0 a 1 nell'arco di **120 frames** per rendere **più chiara** l'immagine, come in una **dissolvenza** in entrata.

La configurazione del **Node Editor** per questo **effetto** è mostrata nell'**immagine** seguente.

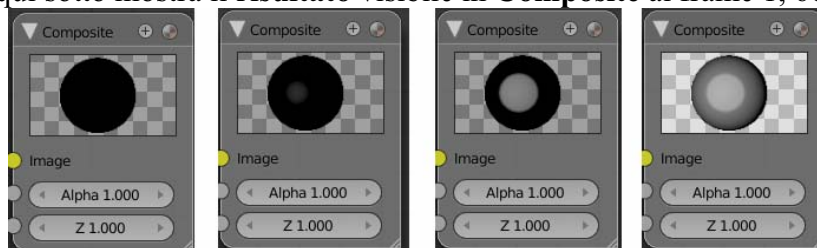


Utilizziamo un **Nodo** di tipo **Input – Time** con la **curva invertita** e colleghiamo alla porta di input **Factor** di **Mix**.

Collegiamo l'uscita **Image** del **Render Layer** alla prima porta **Image** di **Mix**, lasciando la **seconda porta vuota** ed impostando per la stessa il **colore bianco** (in assenza di immagini in input, quindi, è possibile impostare un colore uniforme per i canali in input di **Mix**).

Cambiamo il **tipo di operazione** in **Subtract**: in questo modo, al frame 1 **sottrarremo** alla prima immagine (in arrivo dal **Render Layer**) il **colore bianco uniforme**, per cui in pratica la azzereremo (sottraendo 1.0 ad ogni pixel) ed avremo un'immagine nera, ma andando avanti con i **frame** il **fattore** di miscelazione diminuirà seguendo l'andamento della curva definita in **Time**, per cui **alla fine** (al frame 120, nel nostro caso) **la sottrazione sarà nulla** e avremo l'immagine del **Render Layer** in output.

L'**immagine** qui sotto mostra il **risultato** visibile in **Composite** ai frame 1, 60, 90 e 120.



Per **accelerare** o **rallentare** la **dissolvenza**, è sufficiente **modificare la forma della curva in Time**, inserendo e spostando dei punti di controllo con un click del tasto sinistro del mouse sulla **curva**.

CAPITOLO 4

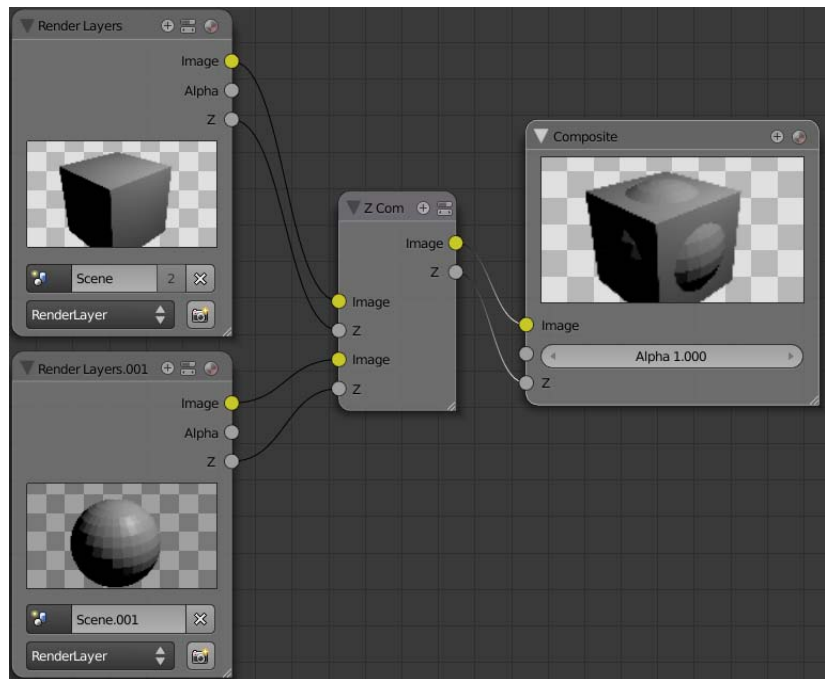
Gli altri Nodi del gruppo Color

Mix è, tra i **Nodi** del gruppo **Color**, quello che offre il maggior numero di **funzionalità**, per cui dopo averlo trattato opportunamente a parte possiamo dedicarci agli altri **Nodi** dello stesso **gruppo**, utilizzati anch'essi per l'**elaborazione del colore**.

Z COMBINE

Z-Combine prende in **input** due **immagini** provviste di informazioni riguardanti il **canale Z**, quello cioè della **distanza** di ciascun **pixel** dalla **telecamera**, e produce in **output** una nuova immagine dove ciascun **pixel** è scelto come quello “**più vicino**” alla **telecamera** confrontando i due pixel delle immagini in **input**; in caso di **parità**, viene scelto il pixel della **prima immagine** in input nel filtro.

Z-Combine restituisce in **output** anche un **canale Z** per trasmettere le informazioni sulla **profondità** per la **nuova immagine** generata, cosa che rende possibile l'utilizzo di più filtri **Z-Combine** “**in cascata**”.



L'**immagine** in alto a destra mostra un **esempio** di utilizzo di tale **filtro**: in questo caso, sono state create due diverse **Scene** all'interno dello stesso file **.blend**, ponendo il cubo e la sfera ciascuno in una propria **Scene** e **renderizzandoli separatamente**; in seguito, è stato effettuato il **rendering composito** dei due **Render Layers** utilizzando la configurazione del **Node Editor** mostrata nell'immagine (in entrambe le **Scene**, **Use Nodes** era **attivo**, ma la configurazione è stata realizzata solo per la prima **Scene**).

ALPHA OVER

Il filtro **Alpha Over** consente di utilizzare un'immagine come **sfondo** (da collegare alla prima porta **Image** in **input**) e un'immagine in **sovrimpressione** (da collegare alla seconda porta **Image** in **input**), mescolandole basandosi sul valore di **miscelazione** dato da **Factor** (eventualmente variabile, utilizzando la relativa porta in **input**) e soprattutto dal valore del **canale Alpha** per l'immagine in sovrimpressione: in caso di **Alpha** minore di 1 o, meglio ancora, nullo per un **pixel** della seconda immagine, il **pixel** risultante mostrerà un contributo dato dal corrispondente **pixel** dell'immagine di sfondo.

In caso di bagliori luminescenti (**halo**), attivare l'opzione **PreMultiply**.

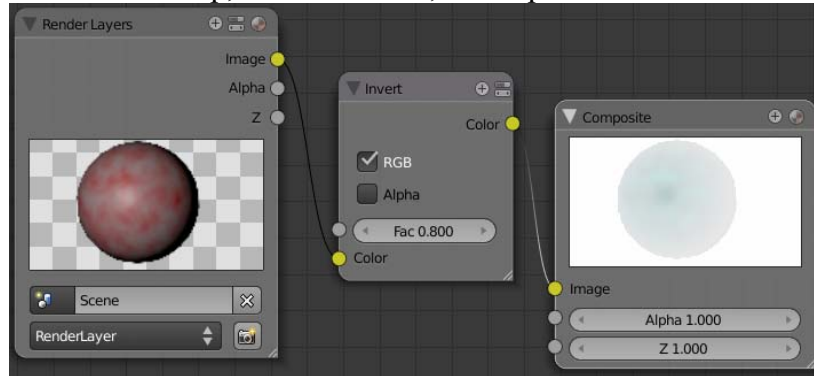


INVERT

Invert serve ad **invertire** i valori **RGB** o l'**Alpha** di un'immagine o di un colore (infatti le porte di input e output si chiamano **Color**, non **Image**), ottenendo sostanzialmente un **negativo** della stessa.

Per invertire l'**Alpha** dell'immagine in arrivo sulla porta di input **Color** è sufficiente selezionare anche la **checkbox Alpha**, mentre di default **RGB** (per invertire il **colore**) è sempre **selezionato**, ma ovviamente può essere **disattivato**.

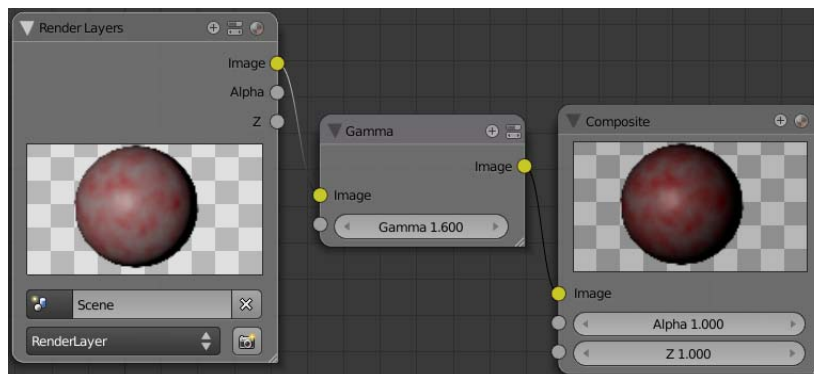
Il **fattore di influenza** dell'effetto, **Factor**, non deve essere necessariamente **costante**, per cui è possibile utilizzare una **ColorRamp**, un **Nodo Time**, ecc... per ottenere effetti differenti.



GAMMA

Tale **filtro** serve a modificare il valore per la **Gamma Correction** delle immagini.

Sono presenti solo le porte di input e output per l'immagine da elaborare e la **porta** per il **valore della correzione**, che quindi non deve essere necessariamente una **costante**.

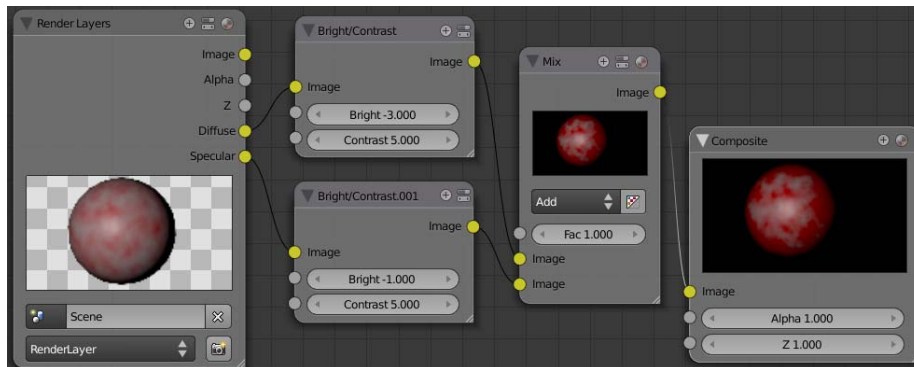


BRIGHT-CONTRAST

Bright-Contrast consente, come suggerisce il nome, di regolare la **luminosità** e il **contrasto** di un'immagine in input, restituendo in output l'immagine così modificata.

I valori di **Bright** (luminosità) e **Contrast** (contrasto) possono essere definiti come **costanti** per tutta l'immagine, mediante i selettori numerici, o **variabili**, utilizzando le **porte** di input per definire i valori con **ColorRamp**, **Time** (ad esempio, per far **aumentare la luminosità** all'inizio dell'animazione), ecc...

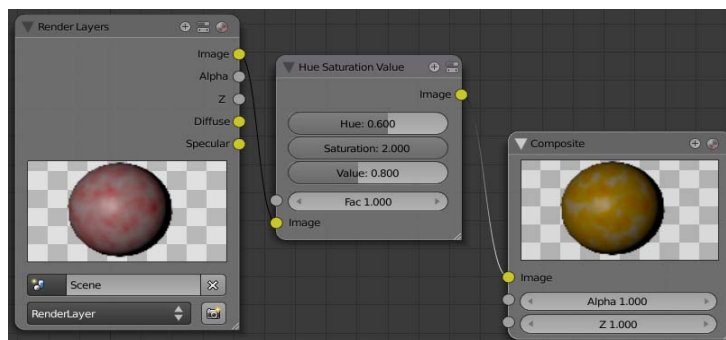
L'immagine seguente mostra una configurazione del **Node Editor** che consente di variare il **contrasto** e la **luminosità** in maniera differente per i **canali Diffuse** e **Specular**, miscelando il tutto con **Mix---Add** per ottenere l'**immagine finale**.



HSV (Hue, Saturation, Value)

Nello spazio dei colori **HSV** è possibile lavorare su **Tonalità** (Hue), **Saturazione** e **Intensità** (Value) di ciascun **pixel**.

Il filtro **Color---HSV** dei **Nodi** di **Blender** consente appunto di modificare questi **tre valori** con altrettanti **controlli numerici** (i valori, in questo caso, possono essere solo costanti), prendendo in **input** un'immagine e un **Factor** (provvisto di una porta di input, per valori variabili), restituendo in **output** l'immagine così elaborata.



RGB Curves

Lo strumento **RGB Curves** è presente in molti programmi di **fotoritocco**.

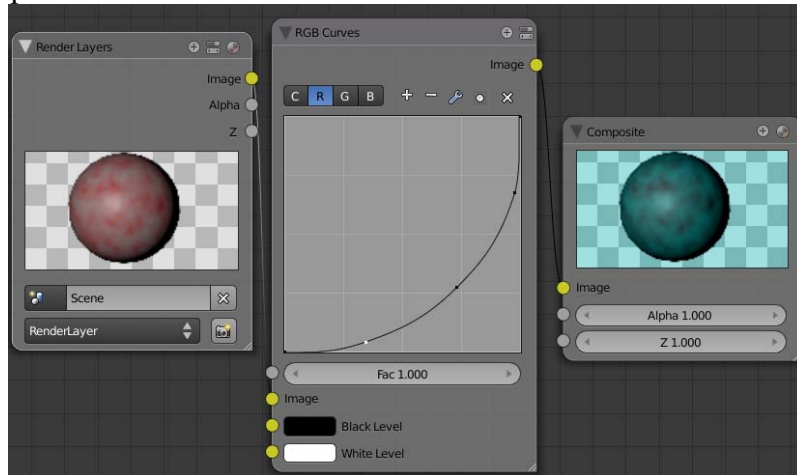
Consente di modificare le **curve di tono** dei singoli **canali RGB** o del **colore complessivo** di un'immagine; in questo modo, è possibile **aumentare** l'influenza di un colore primario, ad esempio il Rosso, per far virare verso tale colore l'immagine, oppure **ridurla**, in modo da rendere l'immagine più "fredda".

Sono presenti dei connettori per l'immagine in input e quella in output, più un paio di **porte** per definire il livello del nero e del bianco (per cui un colore o una Ramp può essere "**mappata come colore nero**" o come **bianco**, scalando linearmente gli altri colori nel nuovo **range**) e una porta per **Factor**, che definisce in che misura viene applicato l'effetto.

Si tratta quindi di un ottimo strumento per **bilanciare i toni** o conferire ad un **rendering** un'**atmosfera** particolare, rendendolo più freddo, più caldi, ecc...

Nel grafico all'interno del **Nodo**, i valori sull'asse delle **X** rappresentano il **valore in input** e quelli sull'asse **Y** i valori di **uscita**, per cui se ad esempio abbasseremo la **curva del Rosso** (inserendo un **punto di controllo al centro** della stessa con un click del tasto sinistro del mouse e trascinando verso il basso), abbasseremo il **contributo di rosso** dato all'immagine in output.

L'immagine seguente mostra l'**effetto** dovuto proprio ad una modifica della curva di tono del **canale R** (rosso), selezionato cliccando sulla casella col simbolo **R** in alto nel **Nodo**, inserendo **tre punti di controllo** per **abbassare** la curva.

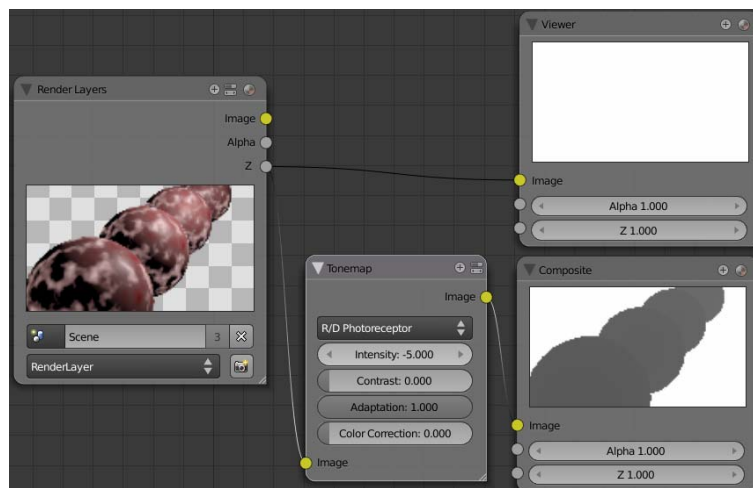


TONE MAP

Il processo di **Tone Mapping** consiste, sostanzialmente, nel **convertire** immagini ad **alto range dinamico** (High Dynamic Range Images, memorizzate ad esempio nel formato OpenEXR) in immagini con un **range dinamico minore** (Low Dynamic Range Images), visualizzabili correttamente su monitor, televisori e display.

Blender mette a disposizione, nel **Nodo Tone Map**, due **tecniche** per realizzare la conversione (**R/D Photoreceptor** e **Simple Rh**) e i controlli per regolare **parametri** come Offset, Gamma, Contrasto, ecc...

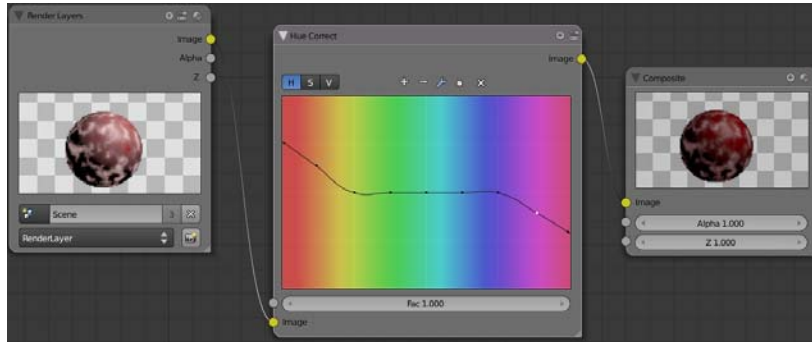
In realtà, non dobbiamo cercare delle immagini **HDR** per testare questo **Nodo**: come abbiamo visto nel **Capitolo 2**, il **canale Z di Render Layers** produce in output un'immagine a **32 bit**, non visualizzabile correttamente se non dopo opportuna compressione del **range dinamico**; ebbene, possiamo utilizzare il **Nodo Tone Map** per effettuare tale operazione, come visibile nell'immagine seguente.



HUE CORRECT

Il **Nodo Hue Correct** consente di regolare, mediante una finestra dotata di una curva con vari punti di controllo, i valori di **Hue** (Tonalità), **Saturation** e **Value** (Intensità) di un'immagine in input, producendo una nuova immagine in output.

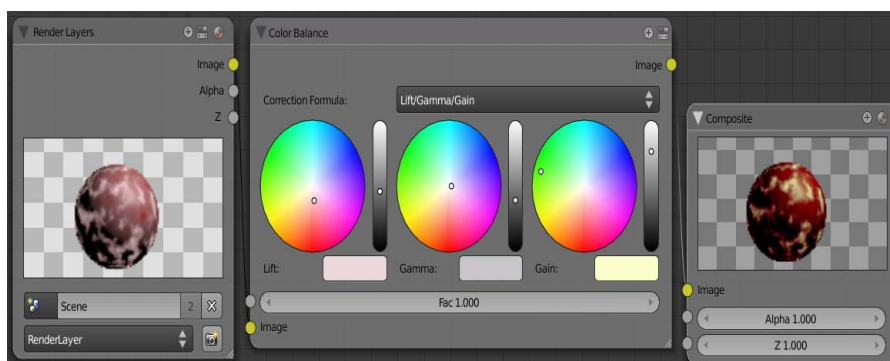
I tre **canali HSV** sono separati, nel senso che è possibile modificare la **curva** di **H** indipendentemente da quella di **S** o di **V** selezionando la **casella** con la relativa **lettera** in alto nel **Nodo**; passando da un canale ad un altro, le **modifiche** fatte al primo canale verranno comunque **preservate**, per cui è sufficiente **un unico Nodo Hue Correct** per modificare tutte e tre le curve di controllo di **HSV** contemporaneamente.



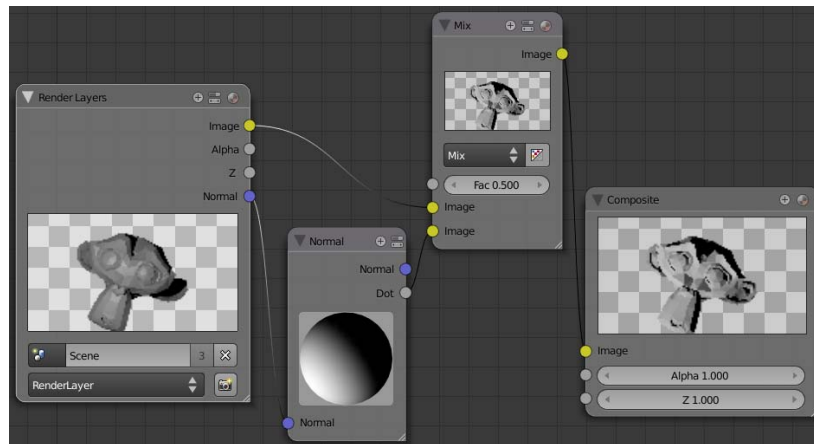
COLOR BALANCE

Il **Nodo Color Balance** mette a disposizione gli strumenti per effettuare la **correzione del colore** utilizzando il modello **Lift/Gamma/Gain** o il modello **Offset/Power/Slope**.

Non tratteremo la teoria della **correzione del colore**, limitandoci a segnalare come tale **pannello**, unitamente al **Nodo** per la correzione **HSV** e a quello del **Contrasto** e della **Luminosità**, consente un **controllo completo** sulla correzione del colore dell'immagine in fase di **Post-Processing**.



* * *

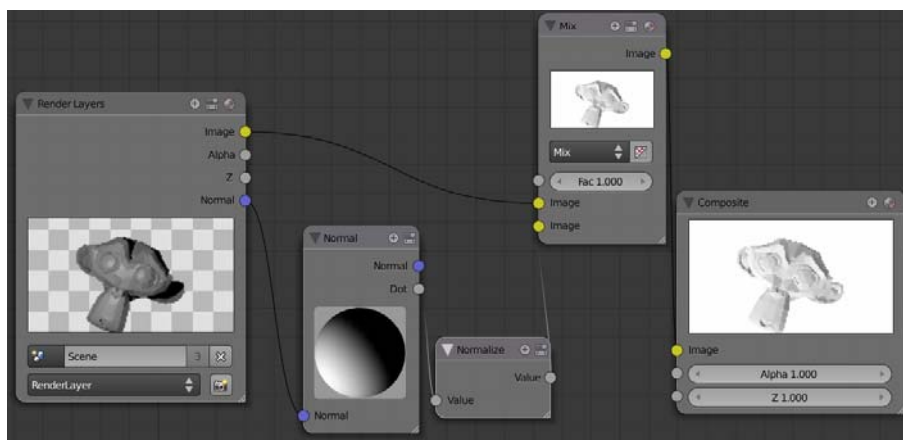


NORMALIZE

Normalize si limita a **normalizzare** (ossia a riportare nel range 0.0 – 1.0), come suggerisce il nome, i **valori in input**, restituendo il **valore** così modificato in **output**.

In assenza di un valore in input, è possibile **impostare un valore costante** da restituire in output (default: 1.0).

Nel prossimo paragrafo vedremo un esempio di utilizzo di **Normalize** (insieme ad altri **Nodi**) che ci consentirà di apprezzarne meglio l'**effetto**.



MAP TO

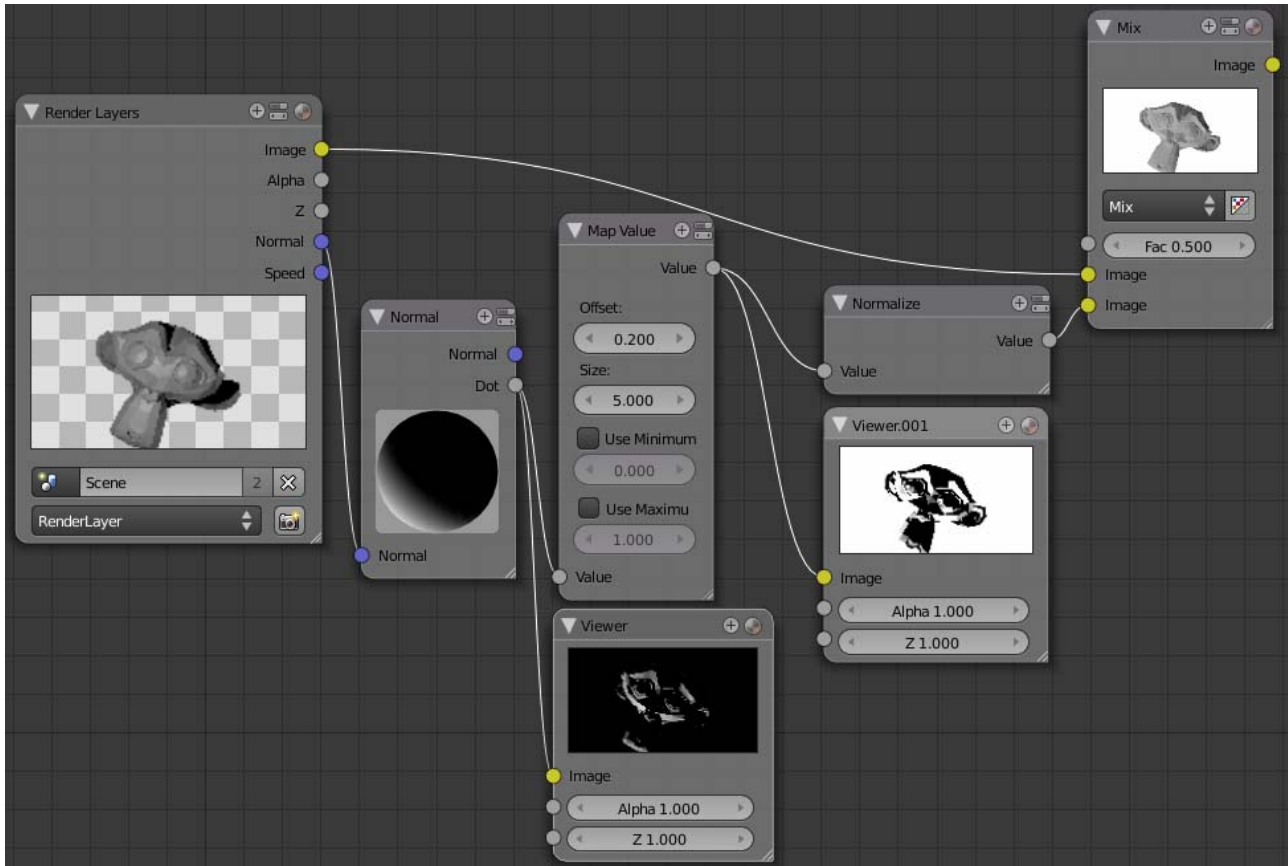
Map to prende in input dei **valori** e li **converte** (li “mappa”, appunto) in un'altra **scala**, **sommando** ad essi un valore (specificato con il campo **offset**), **moltiplicandoli** per un altro valore (specificato con il campo **size**) e **troncandoli** ad un valore minimo (**Min**) o massimo (**Max**) per mantenerli all'interno di una **scala** da noi definita.

In assenza di un **valore** in arrivo sulla porta di **input**, verrà considerato come **valore iniziale** quello specificato nel **relativo campo** (di default, 1.0).

L'immagine riportata nella pagina seguente mostra un possibile utilizzo dei **Nodi Normal**, **Map To** e **Normalize**: il valore **Normal** del **Render Layer** passa al nodo **Normal**, dove viene moltiplicato con il prodotto scalare con il vettore definito internamente al **Nodo**; il valore ottenuto

viene passato a **Map To**, che lo aumenta (offset 0.2) e lo moltiplica per 5 (size), passando il nuovo valore a **Normalize**, che lo **normalizza** e lo fornisce in **Input** a **Color---Mix**, per il **compositing** finale.

A **Normal** e **Map To** sono associati in **output** anche dei **Nodi Viewer**, per valutare gli **effetti parziali**.



* * *

CAPITOLO 6

I Nodi del gruppo Filter

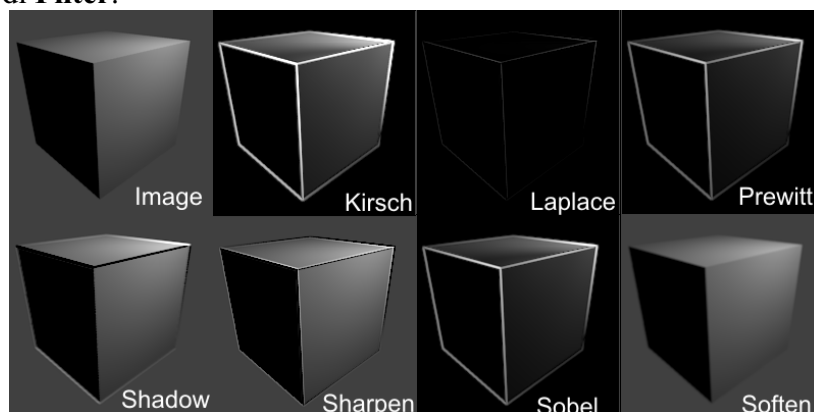
I **Nodi** del gruppo **Filter** consentono di applicare **filtri** ed **effetti** di elaborazione delle immagini digitali ai **rendering** (anche parziali), ad esempio mettendo in evidenza gli **spigoli** o **sfocando** (blurring) l'immagine.

FILTER

Il **Nodo Filter** mette a disposizione diverse opzioni di **filtraggio** dell'immagine fornita in input:

- **Soften** aggiunge un po' di blurring, di sfocatura all'immagine (Soften significa letteralmente "ammorbidisci");
- **Sharpen** è l'opposto di Soften: mette in evidenza gli spigoli;
- **Laplace** implementa una maschera di convoluzione per l'edge detection basata sull'operatore Laplaciano, per cui produce un'immagine dove sono visibili solo gli spigoli;
- **Sobel** è un altro operatore di edge detection, ma a differenza di Laplace il kernel di convoluzione deriva dal Gradiente;
- **Prewitt** è simile a Sobel, anche concettualmente, ma cambiano i valori della maschera di convoluzione e, quindi, gli effetti;
- **Kirsch** è un altro filtro non lineare di edge detection, per evidenziare cioè gli spigoli;
- **Shadow** scurisce gli spigoli, mettendoli in evidenza.

L'immagine seguente mostra il **rendering** di una **scena** e le immagini ottenute applicando alla stessa i vari **filtri** di **Filter**.



BLUR

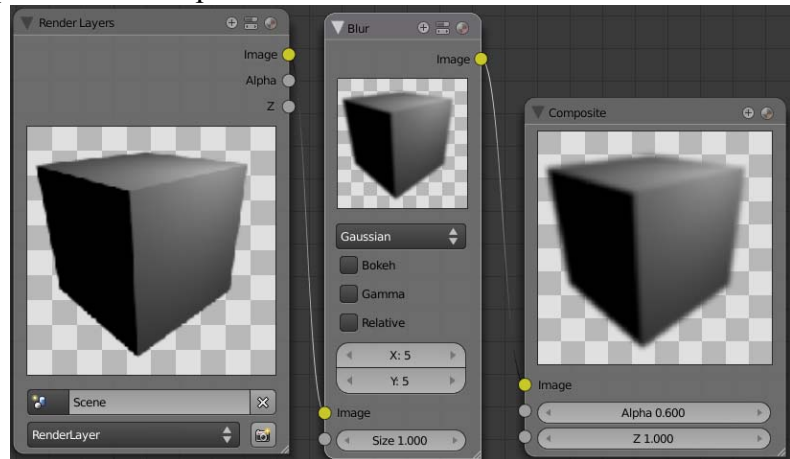
L'effetto di **blurring** consiste nello **sfocare** un'immagine o parte di essa, in genere per simulare effetti come la **profondità di campo** o il "mosso" degli oggetti in **movimento**.

Nella sua versione base, implementata appunto dal **filtro Blur**, è sufficiente impostare (nei campi X e Y del **Nodo**) la **dimensione della maschera** che opererà pixel per pixel e il **tipo di filtro** da utilizzare (tipicamente, Gaussian).

La funzione **Bokeh**, se selezionata, impone l'utilizzo di una **maschera circolare** per il **filtro**; **Gamma**, se selezionata, effettua la **Gamma Correction** dell'immagine prima di applicare il **filtro di blurring**.

Selezionando **Relative**, si avrà la possibilità di esprimere i valori delle **dimensioni X e Y** della maschera di **blurring** in **percentuale**.

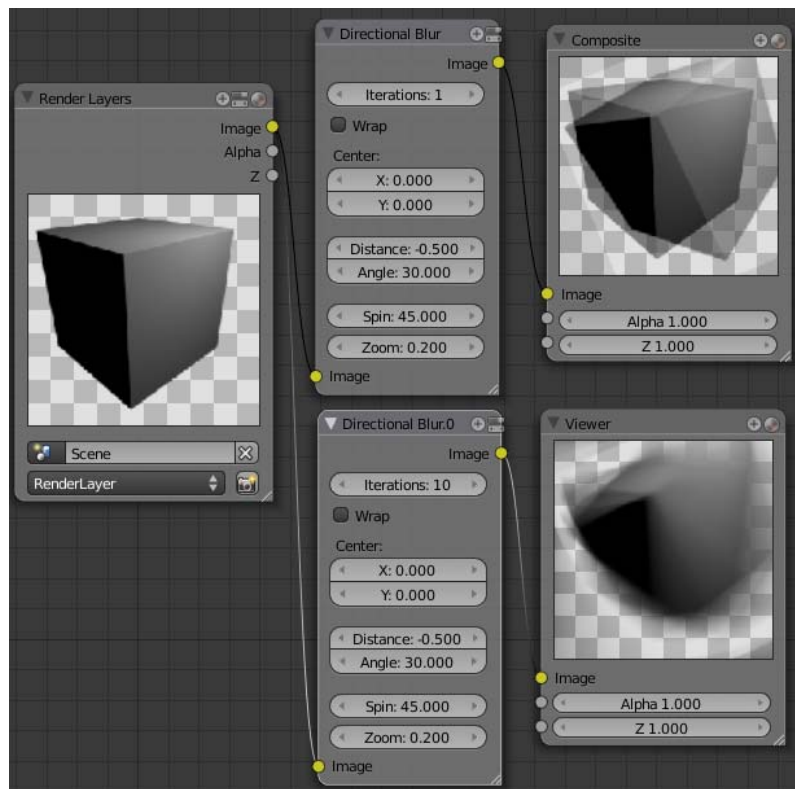
L'immagine a lato mostra un **esempio** di applicazione del **filtro di blurring** ad un'immagine.



DIRECTIONAL BLUR

Il **blurring** realizzato con **Directional Blur** consiste in un insieme di **copie dell'immagine originale** (il numero di copie è definito dal parametro **Iterations**, iterazioni), **spostate** lungo una **direzione** (definita mediante i parametri **Distance** e **Angle**) ed eventualmente **zoomate** (parametro **Zoom**) o **ruotate** (parametro **Spin**, espresso in **gradi**).

L'immagine a lato mostra due risultati ottenuti con due filtri **Directional Blur** che differiscono, nelle impostazioni, solo per il numero di **Iterations**, applicati alla stessa **immagine in input**.



BILATERAL BLUR

Il filtro di **Blurring Bilaterale** effettua un **blurring adattivo** (cioè che si “adatta” alle caratteristiche dell'immagine passata in input) di **alta qualità**, ottimo in presenza di rendering con effetti quali **Raytracing** e **Ambient Occlusion** o per simulare **ombre** in presenza di **riflessioni** e **rifrazioni**.

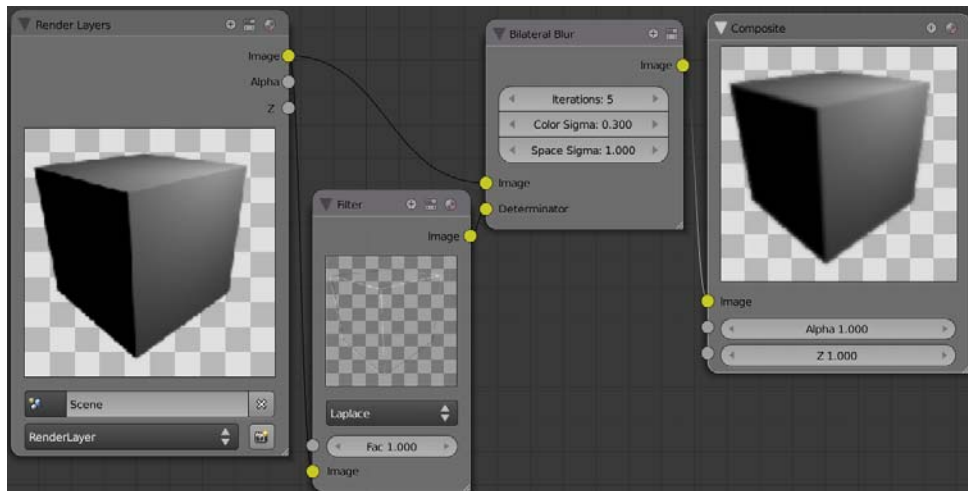
Il **secondo canale** di **input** presente per il **Nodo, Determinator**, non è obbligatorio, ma può servire (specialmente in caso di **immagini molto rumorose** o ricche di **dettagli**) per definire i

bordi e gli **spigoli** sui quali effettuare il **blurring**, altrimenti il **filtro** si baserà sugli **spigoli** che individuerà nell'**immagine** passata in **input** col primo canale.

Il parametro **Iterations** definisce il numero di applicazioni dell'**effetto**, il cui risultato può essere definito ancora più in dettaglio variando il valore del parametro **Space Sigma**.

Il parametro **Color Sigma** è un valore di soglia che stabilisce quando le **differenze tra i colori** di due pixel adiacenti debbano essere considerate come **spigoli**.

L'immagine seguente mostra un'applicazione di tale **filtro** dove per il canale **Determinator** è stata passata, in **input**, l'immagine ottenuta applicando un filtro **Laplaciano** con **Filter** all'immagine originale.

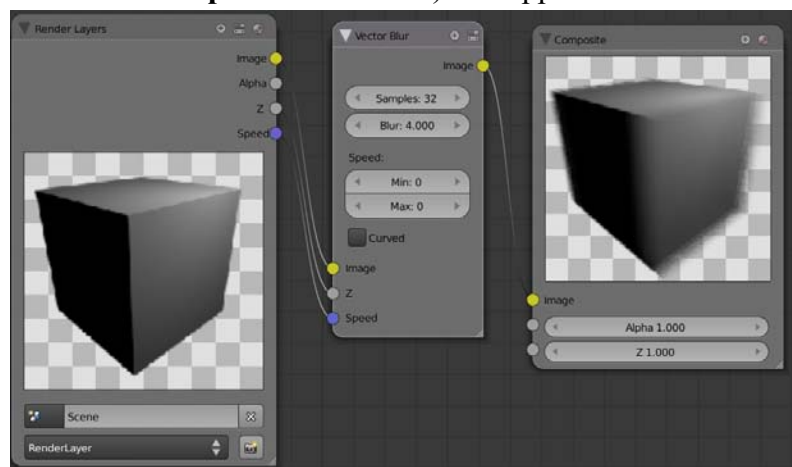


VECTOR BLUR

Vector Blur prende in input le informazioni sul **movimento** degli **oggetti** presenti nella scena, ottenendole dal canale **Speed** di **Render Layers** (NOTA: bisogna attivare **Vector** in **Layers – Passes**, nella scheda **Render**, all'interno della **Properties Window**) ed applica un effetto di **blurring** agli **oggetti in movimento**.

Si tratta di un metodo meno corretto ma più veloce del **Motion Blurring** implementato da **Blender** mediante **Full Sample Motion Blur**, nella scheda **Render** all'interno della **Properties Window**.

I parametri **Samples** (campioni) e **Blur** servono a definire la **qualità** e l'**intensità** dell'effetto, mentre **Speed Min** e **Speed Max** servono per **limitare gli effetti** in presenza di oggetti troppo lenti o troppo veloci, che potrebbero introdurre **effetti indesiderati** nell'immagine finale; il parametro **Curved**, infine, se selezionato effettuerà l'interpolazione tra i punti chiave del **blurring** utilizzando una curva **Bezier**, non in maniera **lineare**.



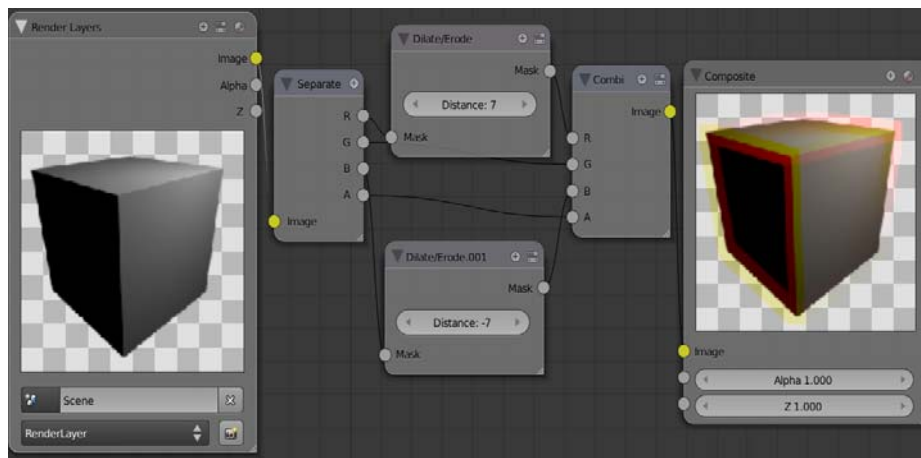
DILATE / ERODE

Dilate/Erode è un operatore **morfologico** (ossia che modifica la **forma** e l'aspetto di un oggetto) che però ha effetto solo su un determinato **canale RGBA** dell'immagine.

Il parametro **Distance** serve ad indicare se il **filtro** dovrà **espandere** (con valori maggiori di 0) l'**area di influenza** del colore o **restringerla** (con valori minori di zero).

Nell'immagine seguente viene mostrata la configurazione del **Node Editor** e il risultato ottenuto **dilatando** il canale **rosso** con **Distance** pari a 7 ed **erodendo** il canale **blu** con **Distance** pari a -7, applicando tale **effetto** ad una **scena** comprendente tre sfere e un plane completamente bianchi.

(**NOTA:** per separare e ricombinare i **canali RGBA** dall'immagine originale sono stati utilizzati i **Nodi Separate RGBA** e **Combine RGBA** del gruppo **Convertor**, che verrà discusso in maniera più approfondita nel **prossimo capitolo**).



DEFOCUS e la profondità di campo

Defocus è un ottimo **filtro** per implementare **effetti** come la **profondità di campo** o, in generale, la **sfocatura** di certe parti dell'immagine (ad esempio, passando una **Color Ramp** o un **Nodo Time** al canale **Z-Buffer**, invece del valore **Z** del **Render Layer**).

NOTA: per poter utilizzare **Defocus** per la profondità di campo, è necessario regolare il **punto focale** della **telecamera** attiva impostando un valore per il parametro **Distance** nella scheda **Object Data** della telecamera, nella **Properties Window**.

Bokeh Type e il valore dell'angolo indicano la **forma** che dovrà avere l'**otturatore** (il numero di lamelle, da 3 a 8, oppure perfettamente circolare) e l'**angolo di fase** dello stesso (rotazione intorno all'asse perpendicolare uscente dal piano dell'otturatore).

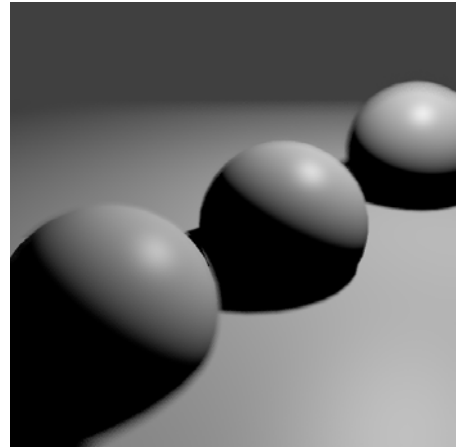
Gamma Correction, se selezionato, attiverà la **Gamma Correction** prima e dopo l'applicazione dell'**effetto**.



fStop indica l'**apertura (f)** della **lente** e consente, quindi, di regolare la **profondità di campo**; il valore 128 indica il valore massimo, con tutti gli oggetti a fuoco.

Da notare che una variazione di **fStop** **NON** ha effetti sulla **quantità di luce** che entra nella telecamera (e quindi sulla **luminosità** dell'immagine renderizzata), come avviene invece con le macchine fotografiche reali.

L'immagine visibile in alto è stata realizzata ponendo il **punto focale** della telecamera in corrispondenza della sfera centrale e con **fStop** pari a 3.



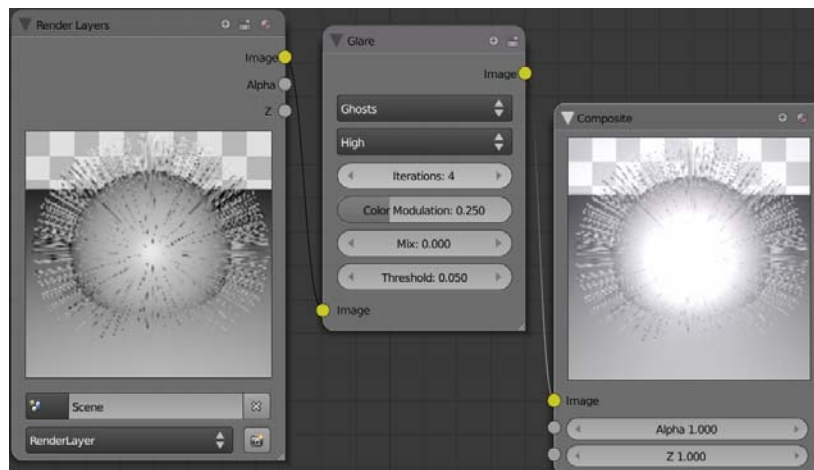
GLARE

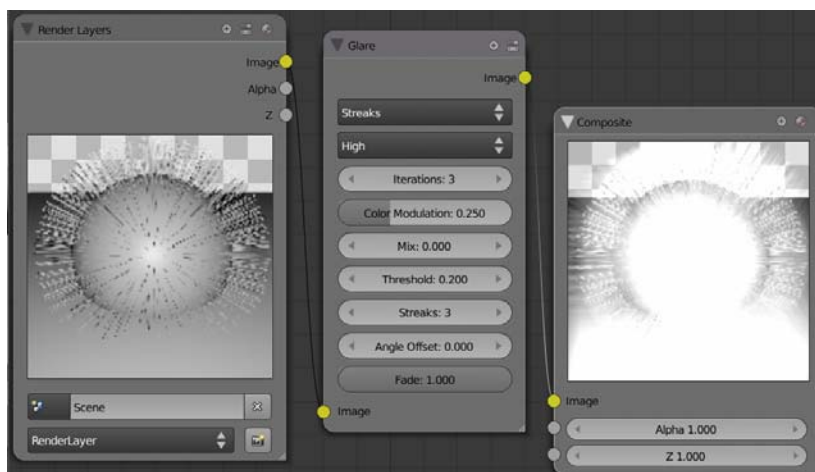
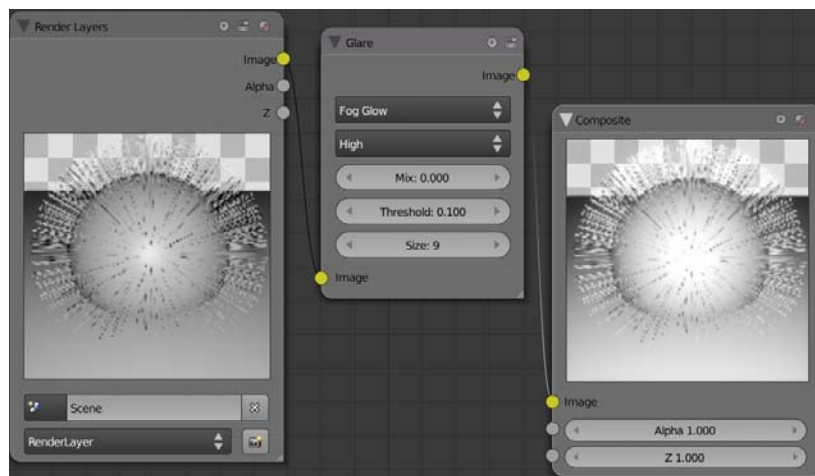
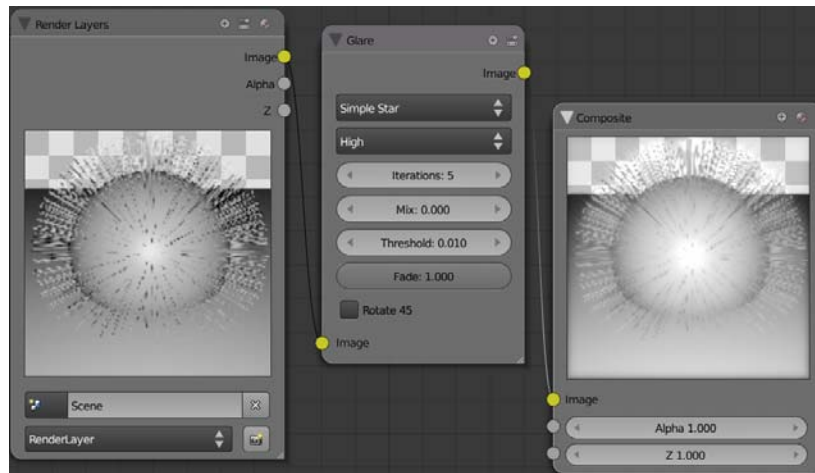
Il **Nodo Glare** consente di aggiungere effetti quali **bagliori**, **glows** e **halo** (*glare* vuol dire proprio “bagliore”); in particolare, i tipi di **Glare** messi a disposizione da **Blender** sono:

- **Fog Glow**;
- **Simple Star**;
- **Streaks**;
- **Ghosts**.

I **parametri** da impostare variano a seconda del **tipo di effetto**, ma principalmente bisogna tenere conto di **Iterations** (il numero di applicazioni dell'effetto) e **Threshold** (valore di soglia che indica di prendere in considerazione per l'effetto i **pixel più luminosi** di tale valore).

Le **immagini** seguenti mostrano alcune applicazioni del **Nodo Glare** con tipi ed impostazioni differenti.





* * *

CAPITOLO 7

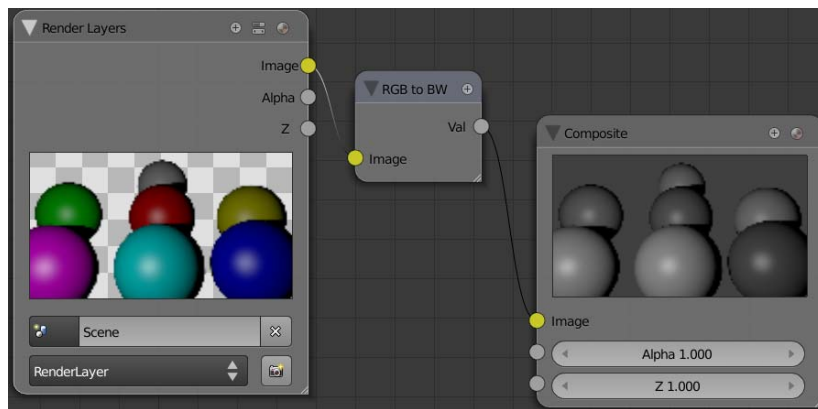
I Nodi del gruppo Convertor

I **Nodi** del gruppo **Convertor** consentono di **modificare le informazioni** relative ai **colori** delle immagini passate in **input**, ad esempio effettuando **operazioni aritmetiche** sulle stesse, **estrapolando canali** (RGBA, YCbCrA, ...) o **combinandoli** per formare un'immagine, impostando un **canale Alpha** per un'immagine ed altro ancora.

RGB to BW

RGB to BW fa quello che suggerisce il nome: **converte** un'immagine a **colori** in una in **bianco e nero** (o, meglio, in scala di **grigi**).

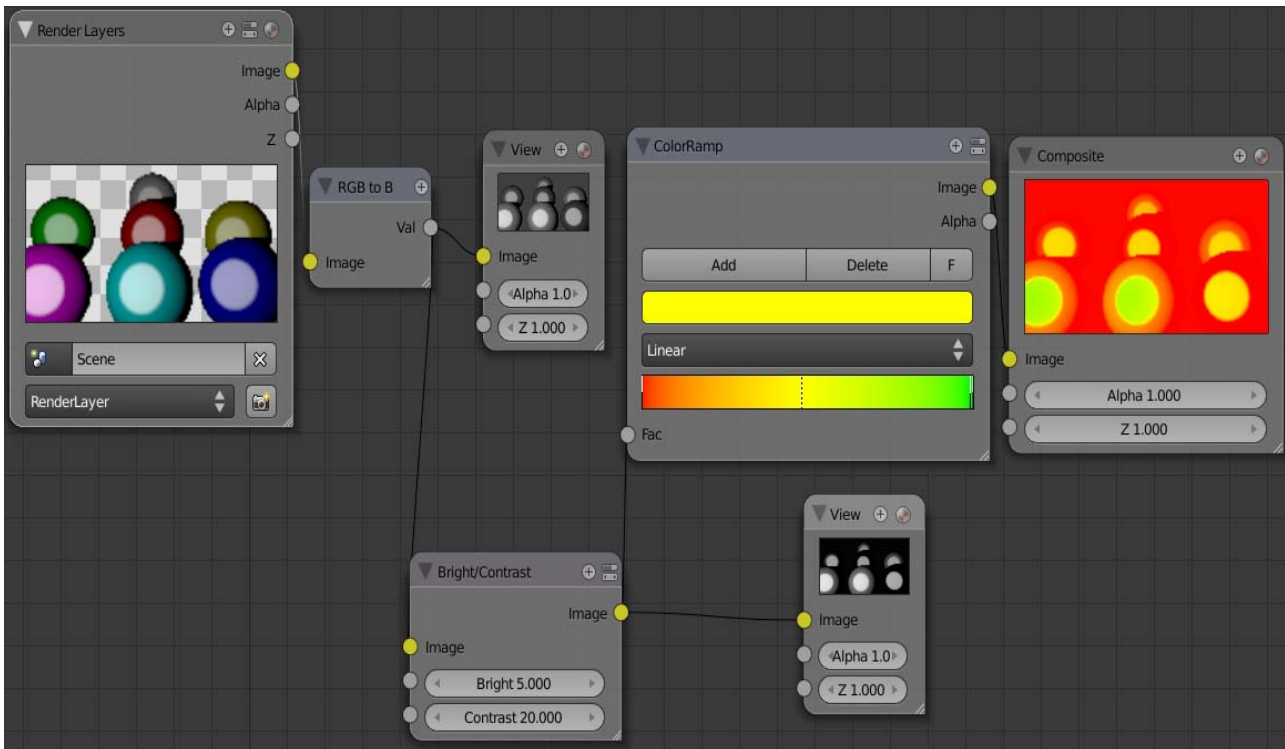
Le uniche due **porte** presenti sono **Image** in input e **Val** in output, non ci sono altri **controlli** o **impostazioni**.



COLOR RAMP

La **Color Ramp** dell'**Editor dei Nodi** è uguale in tutto e per tutto a quella utilizzata con i **Materials** e le **Textures** in **Blender**; anche le **funzioni** sono sostanzialmente identiche: fornisce una “**striscia**” di **colori** (con **Alpha** definibili a piacere) **interpolati**, da utilizzare ad esempio per **colorare** altri **elementi**.

L'applicazione più semplice che possiamo pensare consiste nel **cambiare il colore di un'immagine** basandoci su una **mappa** di colori in **bianco e nero**: faremo corrispondere il colore rosso alle parti nere dell'immagine, il giallo al grigio e il verde al bianco, utilizzando lo schema mostrato nella **figura** nella pagina seguente per l'**Editor dei Nodi** (ho inserito un **Nodo Brightness-Contrast** per aumentare il **contrasto** dell'immagine in bianco e nero in arrivo da **RGB to BW**; i **risultati parziali** sono comunque visibili grazie ai **Nodi** di output **Viewer**).



SET ALPHA

Il **Nodo Set Alpha** prende in **input** un'immagine e ne restituisce in **output** un'altra creata aggiungendo all'immagine in input un **canale Alpha**.

Tale **Nodo** può tornare utile con immagini che non sono provviste di **canale Alpha**, come ad esempio quelle salvate in formato **JPEG**.

Il valore di **Alpha** può essere una **costante** (definita mediante il campo numerico) o una **variabile**: è presente infatti una porta in input su **Alpha**, per cui è possibile far variare il valore della trasparenza con una **Color Ramp**, oppure nel tempo con il **Nodo Time**, ad esempio per far apparire o sparire un'immagine.

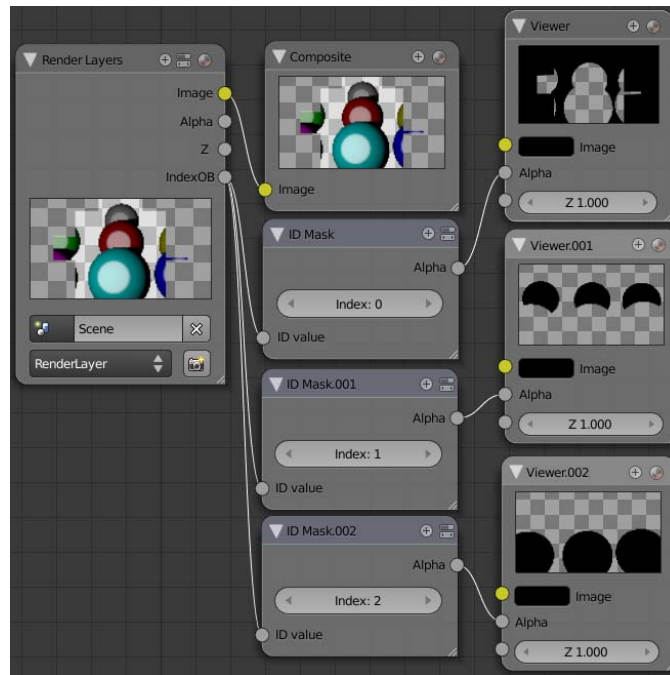


ID MASK

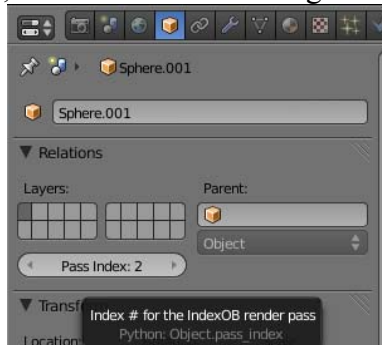
Il **Nodo ID Mask** consente di generare un'immagine con trasparenza creata ponendo a **nero** gli oggetti che hanno un determinato **ID** (un valore **identificativo numerico**, intero) e **trasparenti** tutti gli altri **pixel**.

Si tratta quindi di un ottimo **Nodo** per ottenere **immagini-maschere**, tuttavia per poter usufruire di questa funzionalità bisogna tenere a mente due cose:

- è necessario **attivare**, nei **Render Passes**, l'opzione **Object Index** e collegare la porta di **output IndexOB** di **Render Layers** alle porte di **input ID Value** dei vari **Nodi ID MASK**, come mostrato in figura nella pagina seguente;



- L'ID di ciascun oggetto è, di default, 0, ma può essere modificato accedendo al pannello **Object** nella **Properties Window** ed inserendo un valore diverso nel campo **Pass Index**, nella sezione **Relations**, come mostrato nella figura seguente.



NOTA: anche lo sfondo ha un ID numerico, con valore 0.

MATH

Il **Nodo Math** mette a disposizione diverse **funzioni matematiche** per variare un **valore** in input.

L'input è un **valore**, non un'immagine: per poter elaborare un'immagine, è necessario **scomporla nei suoi canali** (vedremo a breve i **Nodi** del tipo **Separate** e **Combine**) ed operare su ciascun **canale** con un **Nodo Math**.

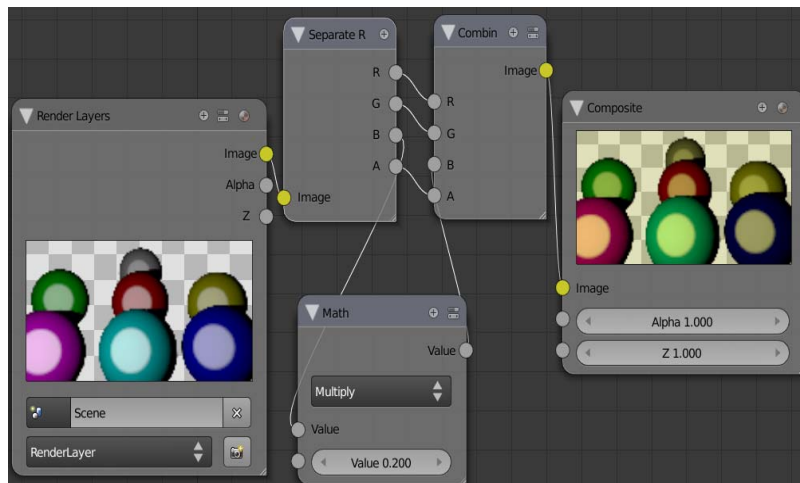
La nota positiva è che possiamo utilizzare le **funzioni** matematiche di **Math** anche su altri **valori**, come ad esempio l'output in arrivo da una funzione **Time**, se abbiamo bisogno di **mappare l'output di Time** in un **range** diverso da 0.0 – 1.0, quando il **Nodo Map Value di Vector** non è sufficiente per questo scopo (ad esempio, utilizzando funzioni trigonometriche o l'elevamento a potenza).

Più in dettaglio, le **funzioni matematiche** messe a disposizione dal **Nodo Math** sono:

- **Add** --- Addizione, somma.
- **Subtract** --- Sottrazione.
- **Multiply** --- Moltiplicazione.
- **Divide** --- Divisione.
- **Sine** --- Seno.
- **Cosine** --- Coseno.
- **Tangent** --- Tangente.
- **Arcsine** --- Arcoseno.
- **Arccosine** --- Arcocoseno.
- **Arctangent** --- Arcotangente.
- **Power** --- Elevamento a potenza.
- **Logarithm** --- Logaritmo.
- **Minimum** --- Prende il valore minimo da un insieme in input (es.: il pixel più scuro).
- **Maximum** --- Prende il valore massimo da un insieme in input (es.: il pixel più luminoso).
- **Round** --- Arrotonda un valore.
- **Less Than** --- Restituisce 1 se il valore in input è minore del valore specificato nel Nodo, 0 altrimenti. Buono per creare immagini-mappe a partire da una soglia.
- **Greather Than** --- Restituisce 1 se il valore in input è maggiore del valore specificato nel Nodo, 0 altrimenti. Buono per creare immagini-mappe a partire da una soglia.

NOTA: le **funzioni trigonometriche** seno, coseno e tangente accettano **valori** nel **range 0 - 2π** .

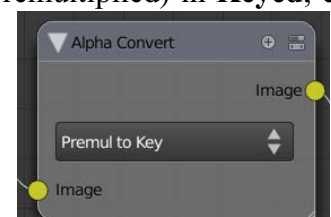
L'immagine seguente mostra la configurazione del **Node Editor** per **moltiplicare** per 0.2 il **canale Blu** di un'immagine.



ALPHA CONVERT

Questo **Nodo** consente di convertire le **immagini premoltiplicate** (Premultiplied) in **Keyed**, e viceversa, a seconda della modalità scelta con l'apposito **selettore**.

Sono presenti solo **due porte**: una per l'immagine in **input** e l'altra per l'immagine restituita in **output**, non ci sono altri **parametri** o **impostazioni**.

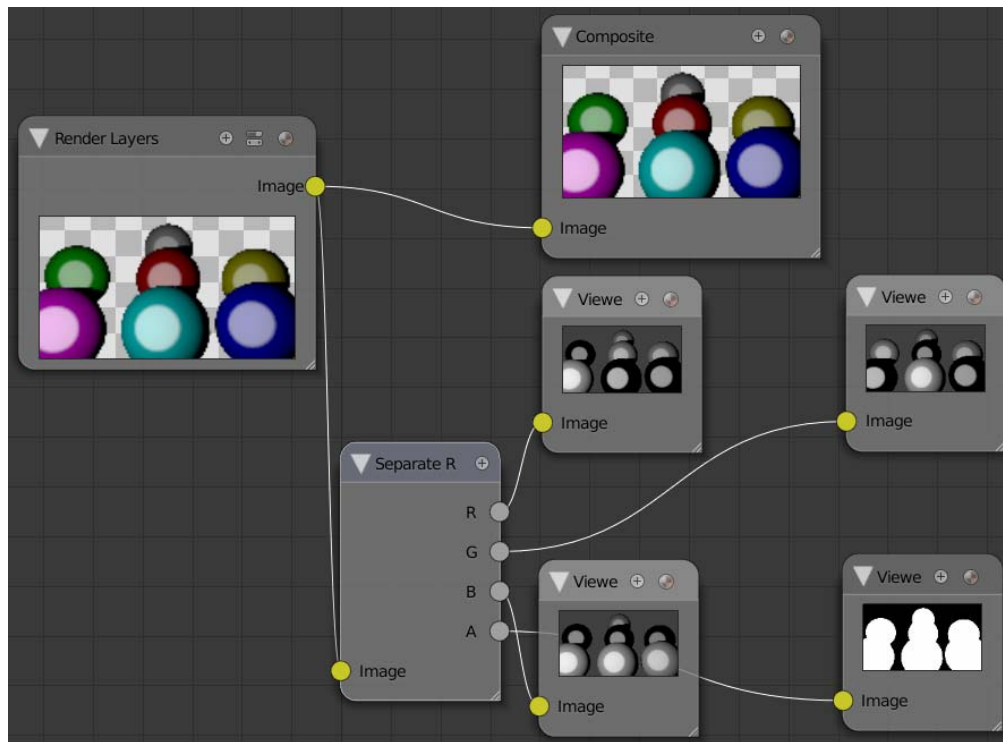


I Nodi del tipo Separate e Combine

I Nodi **Separate** e **Combine** consentono di **suddividere** un'immagine nei suoi **canali principali** (Separate) o, al contrario (Combine), di **creare** un'immagine a partire dalle informazioni relative ai **singoli canali**.

Separate e **Combine** sono tanti perché è possibile scomporre o ricomporre le immagini in **spazi di colore differenti**:

- **RGBA**, ossia i tre colori primari Rosso, Verde e Blu e la trasparenza Alpha.
- **HSVA** (Hue, Saturation, Value, Alpha: Tonalità, Saturazione, Intensità, Trasparenza).
- **YUVA**, luminosità (Y), cromaticanza (U, V) e Alpha.
- **YCbCrA**, luminosità (Y), cromaticanza (Cb, Cr) e Alpha.



* * *

CAPITOLO 8

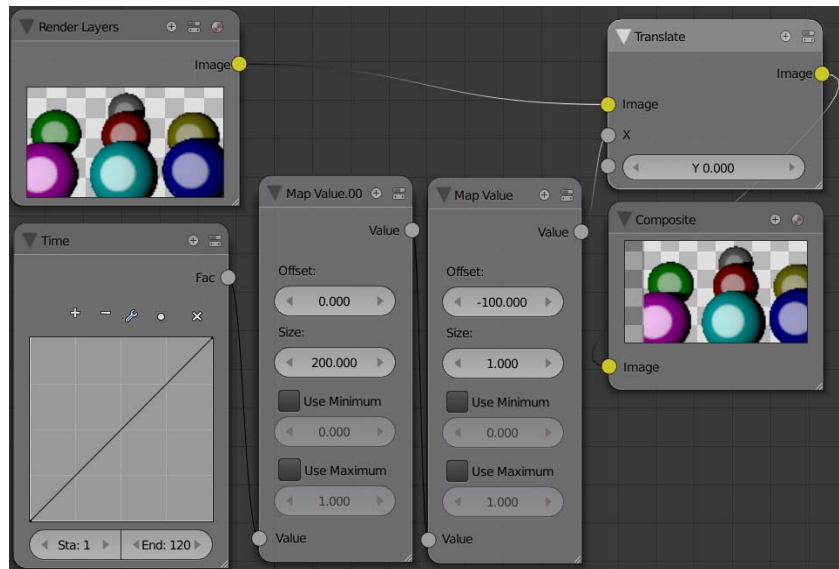
I Nodi del gruppo Distort

I **Nodi** del gruppo **Distort** consentono di **trasformare** le **immagini**, ad esempio traslandole, ruotandole o riflettendole rispetto ad un asse di simmetria.

TRANSLATE

Translate consente, come suggerisce il nome, di **traslare**, ossia **spostare** lungo uno o due assi (X, orizzontale, e Y, verticale) un'immagine.

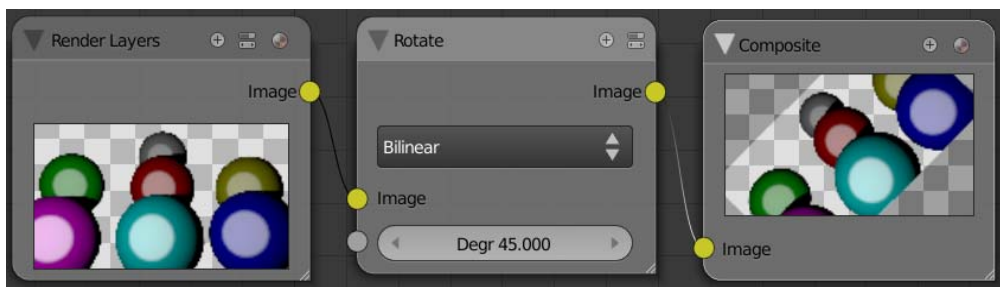
La configurazione del **Node Editor** mostrata in figura in basso, ad esempio, effettua la **traslazione** da -100 a +100 pixel lungo l'asse X nell'arco di **120 frames**; il valore in output di **Time**, infatti, viene **mappato** nel range -100, +100 grazie a due **Nodi Map Value** di **Vector** e collegato in input al **canale X** del **Nodo Translate**.



ROTATE

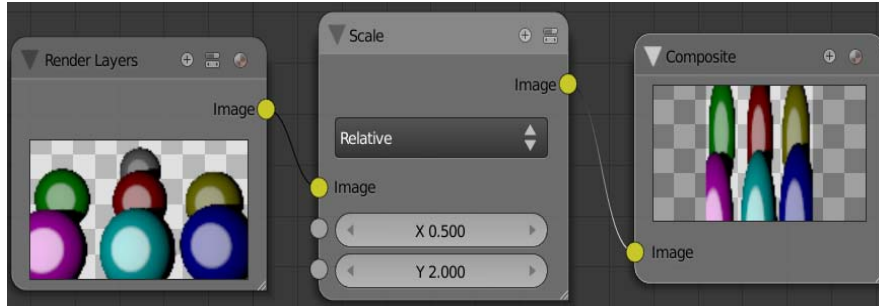
Rotate effettua una **rotazione** dell'immagine intorno al suo **centro**, per un valore – in **gradi** – espresso mediante il campo **Degr**, che è provvisto di una **porta di input**, per cui ad esempio è possibile effettuare l'**animazione di una rotazione** mediante un **Nodo Time**, come visto per **Translate**.

All'immagine ruotata viene applicato un **filtro**, di default **Bilinear** (le altre opzioni sono **Nearest** e **Bicubic**, ma **Bilinear** in genere va più che bene).



SCALE

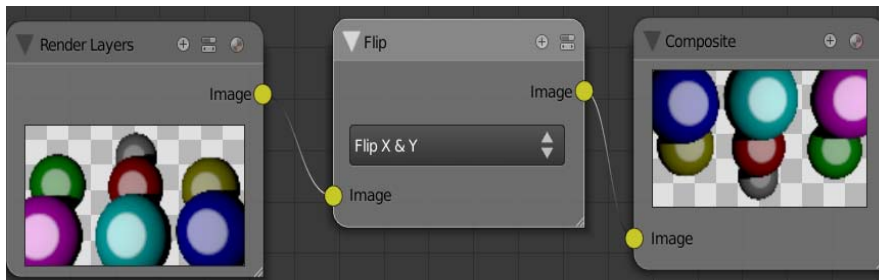
Scale effettua il **ridimensionamento** di un'immagine rispetto ai **due assi**, anche con valori differenti per X e Y, **rimpicciolendola** (con valori < 1) o **ingrandendola** (con valori >1), come visibile nella figura seguente.



I valori dello **scaling** possono riferirsi allo spazio delle dimensioni relative dell'immagine, assolute, della scena o del rendering; la selezione avviene mediante il selettore **Space** presente all'interno del **Nodo**, con voce selezionata di default **Relative**.

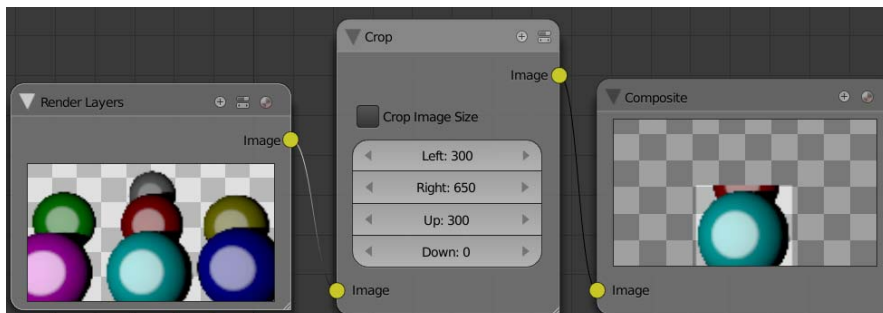
FLIP

Flip **riflette** l'immagine rispetto all'asse X (orizzontale), all'asse Y (verticale) o a entrambi, scegliendo la modalità mediante l'apposito **selettore** presente all'interno del **Nodo**.



CROP

Crop consente di **estrapolare** una sotto-area rettangolare dall'immagine di partenza, specificando i punti di inizio e fine **selezione** con **Left** (punto in cui iniziare la selezione in orizzontale), **Right** (punto in cui terminare la selezione in orizzontale), **Up** (margine superiore della selezione) e **Down** (margine inferiore della selezione).



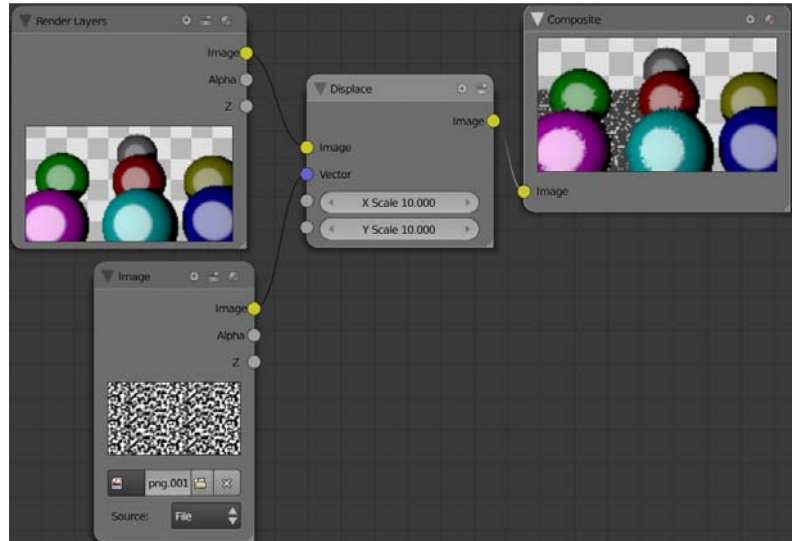
Selezionando la casella **CROP**, la selezione ottenuta verrà **centrata** nell'immagine in **output**.

DISPLACE

Displace serve a “spostare” i pixel dell’immagine di base basandosi sui valori dei pixel di un’immagine in scala di grigi.

L’immagine da modificare va collegata alla porta di input **Image** del **Nodo**, mentre alla porta di input **Vector** va collegata l’immagine da utilizzare per il **displacement** dei pixel.

I valori **X Scale** e **Y Scale** (costanti o variabili, in quanto provvisti di **porte di input**) servono ad **accentuare** o a **ridurre** l’effetto **displacement**.



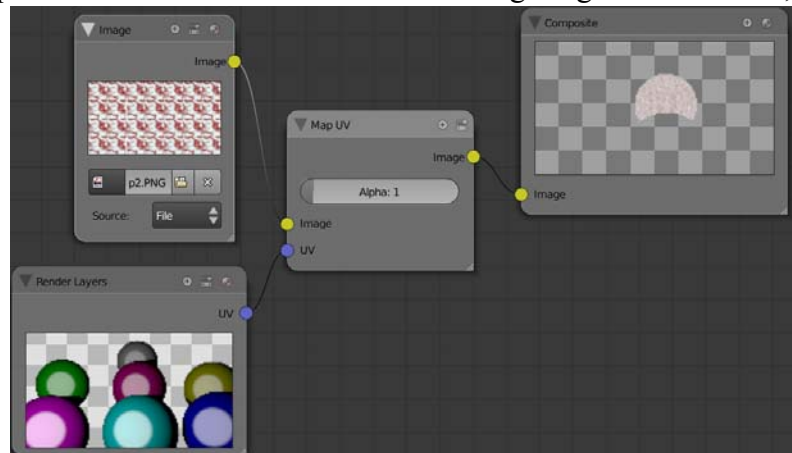
Per poter applicare correttamente l’effetto, le due immagini devono avere le **stesse dimensioni** (nell’immagine sopra, il **Node Editor** configurato per usare **Displace** con un’immagine per il canale **Vector** che però è più piccola dell’immagine da modificare, per cui viene applicata solo ad una **porzione** di quest’ultima).

MAP UV

Il **Nodo Map UV** permette di applicare una nuova **texture** ad un’immagine già renderizzata, **sostituendo le textures** presenti.

La nuova **Texture** va collegata alla porta di input **Image** di **Map UV**, mentre il canale **UV** proveniente da **Render Layers** (UV deve essere attivato in **Render Passes**, nella scheda **Render**, nella **Properties Window**) va collegato alla porta di input **MAP UV** del **Nodo**.

Ovviamente, gli oggetti presenti nella scena devono essere provvisti di layers **UV** e di **Textures** mappate con coordinate **UV** su tali layers.



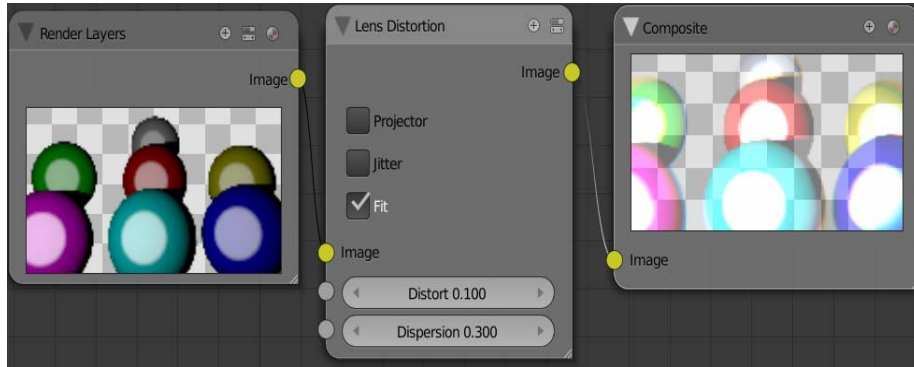
Nell’immagine in alto si vede l’applicazione dell’effetto ad una scena nella quale solo una sfera (quella centrale) era provvista di **layer UV** e **Texture** associata al **Material**.

Utilizzando in maniera opportuna il **Nodo ID Mask**, è possibile isolare i vari oggetti e sostituire le **textures** una per una.

LENS DISTORTION

Il **Nodo Lens Distortion** consente di implementare alcuni effetti di **distorsione** dovuti alla presenza di **lenti**, come ad esempio la distorsione lenticolare vera e propria o l'aberrazione cromatica.

Il valore della **distorsione lenticolare** può essere regolato agendo sul parametro **Distort**, mentre il parametro **Dispersion** serve a regolare l'**aberrazione cromatica**.



Selezionando **Jitter** si otterrà un risultato più “rumoroso”, in quanto i canali-colore **RGB** verranno separati in maniera **casuale**.

Selezionando **Fit** l'immagine in output verrà **scalata** per adattarla alle dimensioni del **frame**.

Selezionando infine **Projector**, la distorsione lenticolare di base verrà **disattivata** e verrà invece introdotto un **offset** orizzontale per i singoli **canali colore RGB**, simulando un proiettore non calibrato (da cui il nome).

* * *

CAPITOLO 9

I Nodi del gruppo Matte

I **Nodi** del gruppo **Matte** consentono, in generale, di creare delle **maschere di opacità** (Matte, appunto) per variare la **trasparenza** dei pixel in base, ad esempio, al valore di luminosità o di canale-colore degli stessi.

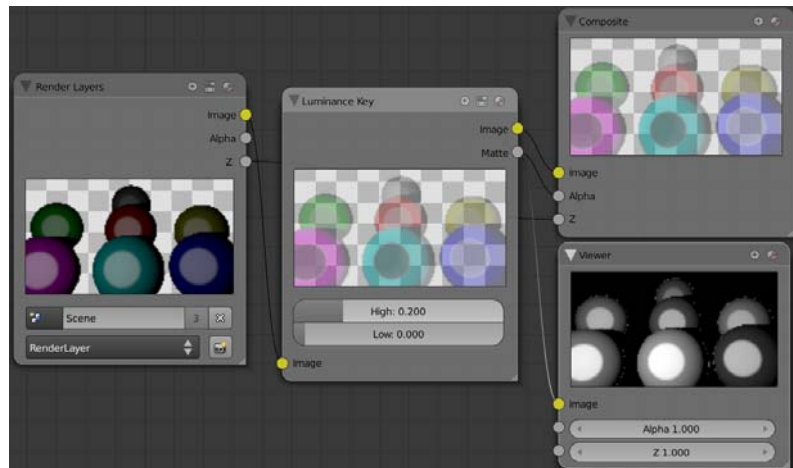
LUMINANCE KEY

Luminance Key prende in input un'immagine e, in base a due valori di **soglia**, considera certi pixel come **opachi** e altri come **trasparenti** o **semi-trasparenti**, restituendo in output l'immagine elaborata e un'immagine-mappa (**Matte**) del canale **Alpha** così elaborato.

Si tratta quindi di un **filtro** utilissimo per **separare**, in un'immagine, oggetti molto **chiari** rispetto ad uno sfondo o ad altri elementi più **scuri** (es.: fiamme, esplosioni, ...).

La distinzione viene effettuata in base al **valore di luminosità** di ciascun pixel.

Le due soglie sono **High** e **Low**: tutti i pixel con valori di luminosità maggiore di **High** verranno considerati completamente **opachi**, mentre tutti quelli con valori minori di **Low** verranno considerati completamente **trasparenti**; i pixel con valori compresi nel **range Low – High** avranno valori di **Alpha** variabili.

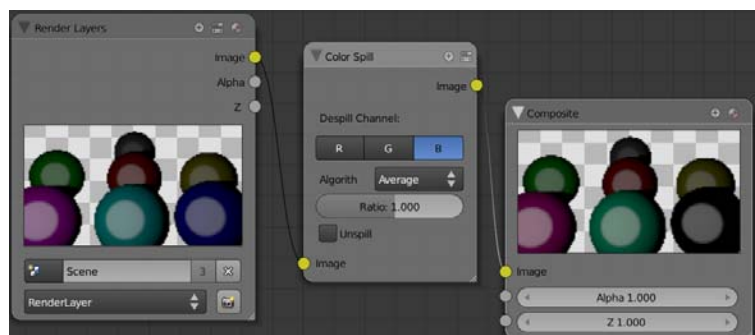


Mantenendo **High** a 1 e **Low** a 0, i valori di **Alpha** verranno mappati **linearmente** sui valori della **luminosità** dei pixel, per cui solo i pixel neri saranno completamente trasparenti e solo i pixel bianchi saranno completamente opachi.

COLOR SPILL

Il **Nodo Color Spill** implementa un filtro che, presa un'immagine in input, **riduce** l'influenza di uno dei tre colori primari (**RGB**), restituendo l'immagine così elaborata in output.

Per effettuare il processo **inverso**, ossia **compensare** uno o più canali colore, è sufficiente selezionare **Unspill** ed inserire manualmente, nei pulsanti che appariranno a video, i valori per i tre canali **RGB**.

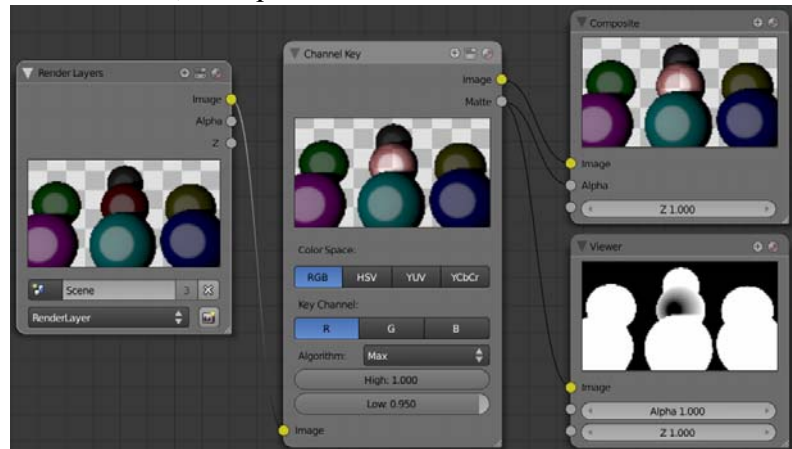


CHANNEL KEY

Channel Key si comporta come **Luminance Key**, ossia considera certi **pixel** come trasparenti, semitrasparenti o opachi a seconda del loro valore, ma questa volta considerando il valore di un **canale-colore**.

Il **Nodo** prende in input un'immagine e restituisce in output l'immagine col nuovo **Alpha** e l'immagine-mappa (**Matte**) per la trasparenza (**Alpha**) elaborata.

All'interno del **Nodo** è possibile scegliere lo **spazio colore** nel quale effettuare la selezione del canale (RGB, HSV, YUV, YCbCr), quindi il canale da considerare e i valori **High** e **Low**: **High** indica il valore del colore superato il quale il pixel verrà considerato come completamente opaco (**Alpha** a 1), mentre **Low** indica il valore al di sotto del quale un pixel verrà considerato completamente trasparente (**Alpha** a 0), mentre a valori intermedi corrisponderanno **Alpha** (e, quindi, trasparenze) intermedie.

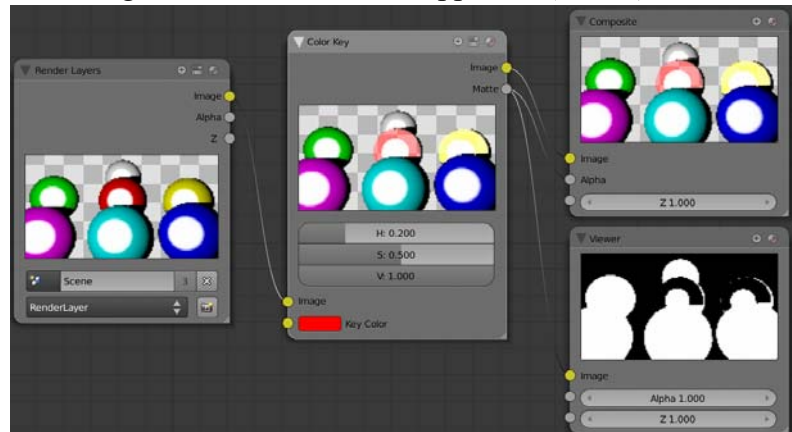


COLOR KEY

Color Key prende in input un'immagine e un **Colore** (anche una **ColorRamp**, mediante la seconda porta di ingresso) e restituisce l'immagine elaborata e una mappatura (**Matte**) del canale **Alpha** (trasparenza) elaborato.

I pixel dell'immagine originale con **colore simile** al colore di confronto avranno valori di **Alpha** variabili da 0 a 1 (a seconda della distanza del loro colore dal colore di confronto), mentre tutti gli altri saranno opachi (**Alpha** a 1).

I tre valori **Hue**, **Saturation** e **Value** (Tonalità, Saturazione e Intensità) sono tre valori di **tolleranza** per specificare quando considerare un colore "simile" al colore di confronto.

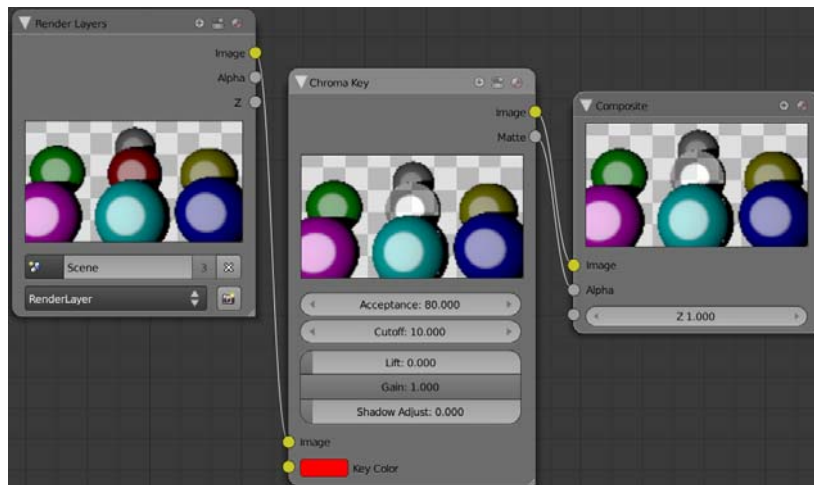


CHROMA KEY

Il **Nodo Chroma Key** consente di considerare certe parti di un'immagine in **sovrapposizione** come **trasparenti**, scegliendo quei pixel in base ad un **colore** (valore cromatico) o un'immagine inviata sul secondo canale in input.

L'output è costituito dall'immagine con il nuovo **Alpha** e dall'immagine-mappa (**Matte**) del canale **Alpha** elaborato.

I valori **Acceptance** e **Cutoff** sono delle soglie: **Acceptance** serve per determinare fino a che punto (con che tolleranza) considerare *simili* due pixel in base al colore, mentre **Cutoff** serve a determinare quando considerare *uguali* due pixel.



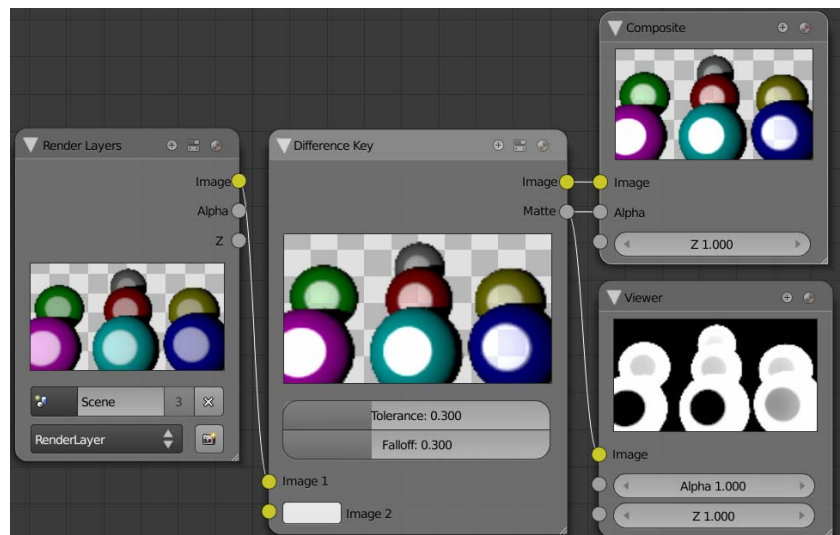
DIFFERENCE KEY

Difference Key permette di effettuare la **differenza** pixel per pixel tra un'immagine e un colore o tra due immagini, restituendo una nuova immagine ottenuta dalla prima ma variando l'**Alpha** per i pixel che coincidevano (o quasi, utilizzando **Falloff**).

L'immagine di partenza deve essere collegata alla porta di input **Image 1**, mentre per **Image 2** è possibile scegliere un colore (default) oppure un'immagine.

Collegando ad **Image 2** la stessa immagine fornita in input per **Image 1**, si avrà in output un'immagine completamente **trasparente** (la differenza tra due pixel corrispondenti delle due immagini sarà nulla).

Il valore **Tolerance** indica, come suggerisce il nome, un valore di **tolleranza** per la differenza tra due **pixel**, per scegliere se considerarli o meno **uguali** (e quindi porre l'**Alpha** a 0).

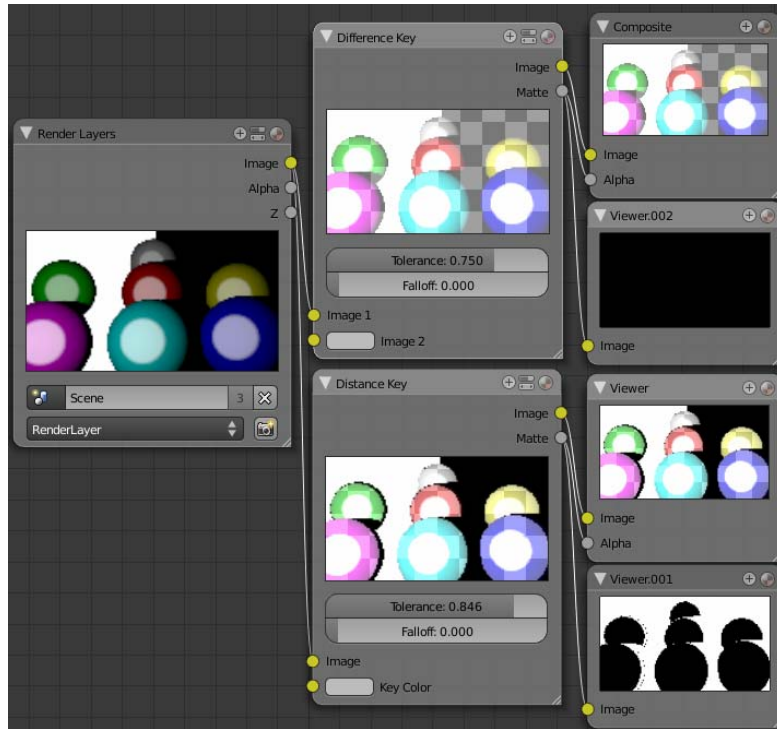


Il valore **Falloff** (lett.: caduta) serve a far variare un po' il valore dell'**Alpha**, per evitare passaggi netti da **Alpha 1** ad **Alpha 0** lungo la frontiera tra pixel "uguali" e pixel "diversi".

Il **Nodo** restituisce in output l'immagine elaborata e la maschera (**Matte**) ottenuta per l'**Alpha**; collegando ad un nuovo **Nodo** l'output si avrà l'immagine di partenza con la trasparenza applicata, come visibile nella figura in alto, che mostra proprio questa configurazione nel **Node Editor**.

DISTANCE KEY

Il comportamento del filtro implementato dal **Nodo Distance** è molto simile a quello visto con **Difference**, ma con una... differenza sostanziale: mentre in **Difference** il colore di confronto veniva **sottratto** al colore di ciascun pixel dell'immagine e, in base al valore ottenuto, veniva calcolato l'**Alpha** per i vari pixel; in **Distance** viene considerata la **differenza in valore assoluto** tra il colore di confronto e il colore del pixel considerato.



L'immagine mostra i risultati ottenuti con **Difference** e **Distance** (con valori di colore di confronto, tolleranza e falloff uguali) applicati alla stessa immagine in input.

* * *

CAPITOLO 10

Raggruppamenti e dipendenze cicliche

Finora abbiamo lavorato con **compositing semplici**, fatti di pochi **Nodi** e poche **connessioni**, ma in situazioni più complesse le cose potrebbero complicarsi un po' e rendere **illeggibile** la finestra **Node Editor**; fortunatamente per noi, in **Blender** è possibile selezionare un insieme di **Nodi** (con le relative connessioni) e **raggrupparli**, creando cioè un unico **Nodo** particolare, detto **Gruppo**, che nasconde le componenti interne e le connessioni mostrando solo le **porte** di input e output, in modo da risparmiare spazio sul **Node Editor** ed evitare **confusione** nella lettura dello schema.

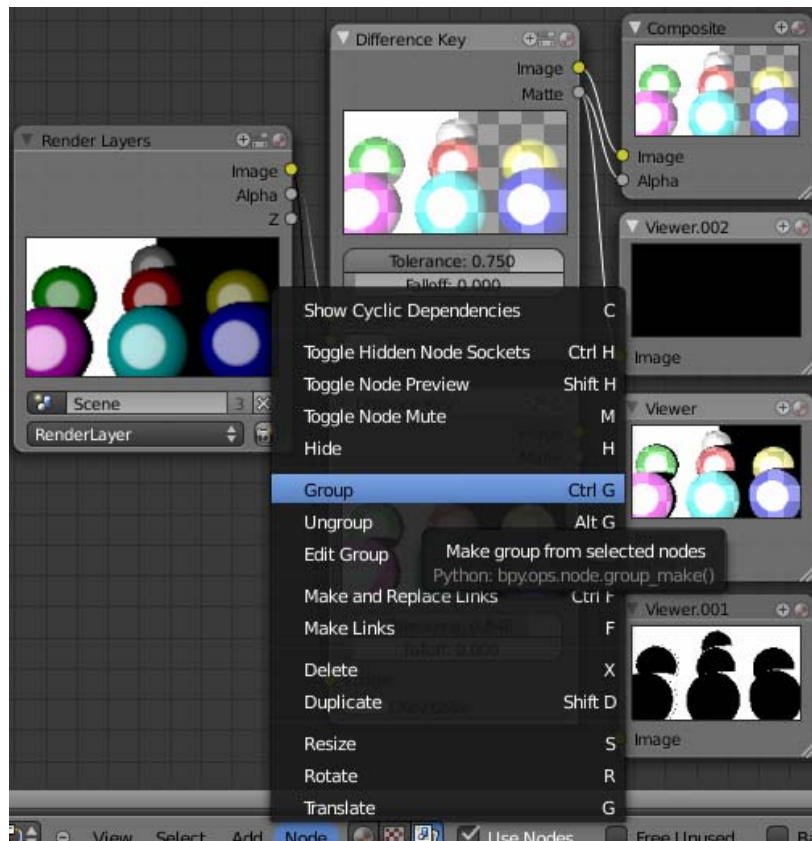
Per creare un **Gruppo**, selezioniamo un **insieme di Nodi** (ad esempio con la maschera rettangolare) e premiamo **CTRL-G**, oppure scegliamo la voce “**Group**” dal menù **Node**, come mostrato nella seconda figura di questa pagina.

I **Nodi** originali scompariranno, lasciando il posto al **Nodo Gruppo** appena creato, come visibile nell'immagine seguente.



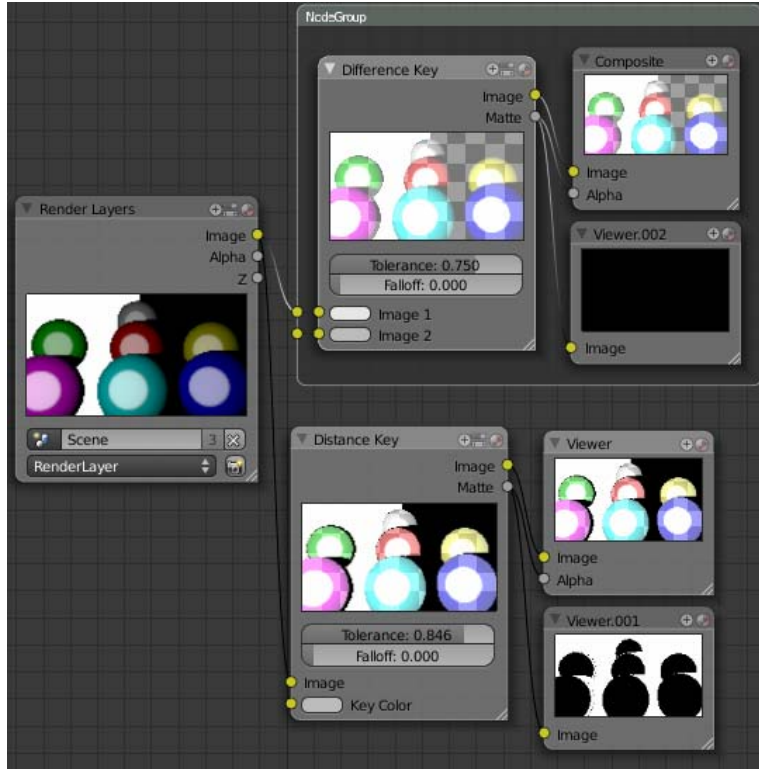
E' possibile, ed anzi è consigliabile, cambiare il **nome** per tale **Gruppo**, scegliendo magari un nome significativo.

Possiamo **cancellare** un **Gruppo**, ricreando i **Nodi** e le connessioni originali, selezionandolo e premendo **ALT-G**, oppure scegliendo la voce “**Ungroup**” dal menù **Node**.



Per modificare un **Gruppo**, ad esempio cambiando i collegamenti o inserendo nuovi **Nodi**, non è necessario scioglierlo, modificarlo e ricostruirlo: basta premere **TAB** (o scegliere la voce **“Edit Group”** nel menù **Node**) per operare in **modalità editing** (un po’ come avviene con la modellazione delle **mesh**) e premere nuovamente **TAB** o **“Edit Group”** per tornare alla visualizzazione classica.

La visualizzazione del **Gruppo** in modalità **editing** è mostrata nell'immagine presente nella pagina seguente.



Prima di chiudere questo breve capitolo, parliamo di una funzionalità utile per scovare eventuali **dipendenze cicliche** all'interno dell'**Editor dei Nodi**: la funzione **Show Cyclic Dependencies**, richiamabile con l'omonima voce nel menù **Node** o con lo shortcut **C**.

Tale funzione mette in evidenza, se presenti, le **dipendenze cicliche** (esempio base: l'output di un **Nodo Mix** è l'input di uno dei suoi canali) consentendoci di eliminarle prima di avviare il **rendering con loop infiniti**.

* * *

APPENDICE

Esempi pratici di Compositing con i Nodes

Dato che nei capitoli precedenti abbiamo visto alcuni esempi di utilizzo dei **Nodi** e delle loro funzionalità per ottenere alcuni **effetti**, è giunto il momento di analizzare alcune **applicazioni pratiche**.

* * *

Sfocare le ombre con Blur

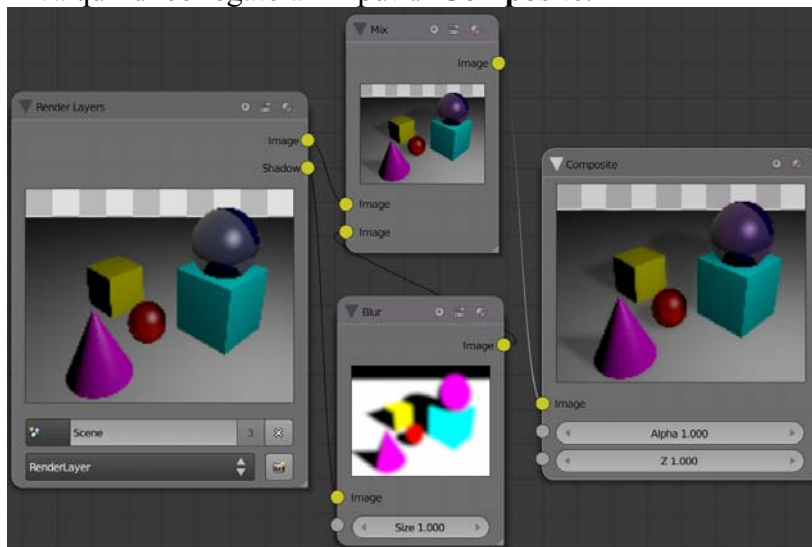
Un buon metodo per **sfocare le ombre** prodotte dagli oggetti consiste nell'applicare al canale **Shadow** di **Render Layers** un filtro di **blurring** (il classico **Blur** andrà più che bene).

Per prima cosa, dobbiamo **escludere le ombre** dall'immagine restituita sul canale di output **Image** da **Render Layers**; per far ciò, clicchiamo sul simbolo della macchina fotografica posto accanto a **Shadow** in **Render Passes** (scheda **Render** nella **Properties Window**), ma selezioniamo la casella di spunta posta a sinistra di **Shadow**, in modo di avere la porta di output **Shadow** nel **Nodo Render Layers**.

Le **ombre** infatti vengono comunque calcolate da **Blender** e sono disponibili in **Render Layers**, solo che con queste impostazioni le abbiamo **escluse** dall'immagine temporanea.

Configuriamo i **Nodi** come visibile nella figura seguente, ossia collegando l'uscita **Shadow** di **Render Layers** ad un filtro **Blur** (ed impostando i vari parametri dello stesso come preferiamo), per poi collegare l'immagine in uscita da tale filtro ad un **Nodo Mix** con funzione **Multiply**, in modo da moltiplicare le ombre sfumate all'immagine originale in arrivo da **Render Layers**.

L'output di **Mix** va quindi collegato all'input di **Composite**.



Una piccola considerazione sul **Nodo Mix**: in genere, per mescolare due canali di un'immagine (ad esempio, **Diffuse** e **Specular**) si ricorre alla funzione **Add**, la somma dei valori, che però non è appropriata con le ombre; le **ombre**, infatti, devono **scurire i pixel che influenzano**, per cui si fa ricorso alla funzione **Multiply**, il prodotto, visto che il prodotto di due numeri minori di 1 darà in input un numero più piccolo del minore dei due, ossia **un pixel più scuro** di quello di partenza.

* * *

Sfocatura animata

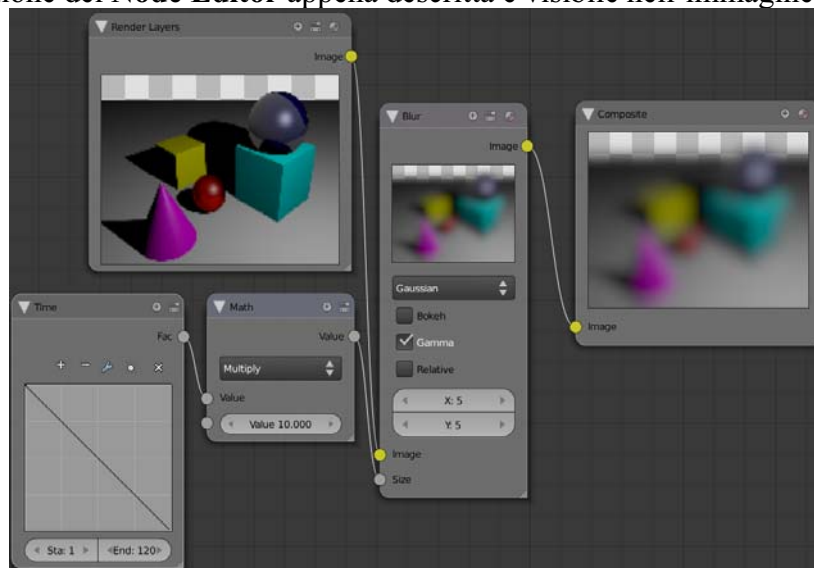
Utilizzando i **Nodi Time** e **Map Value**, possiamo aggiungere un **effetto di blurring** con valore **decescente** nell'arco di un certo numero di **frames** all'immagine, in modo da realizzare l'**animazione** di una sfocatura che a poco a poco sparisce, effetto utile ad esempio per un personaggio che inizia a riprendersi dopo una caduta o una batosta e inizialmente vede tutto sfocato (nel paragrafo successivo, "**L'ubriaco**", viene presentata una variante più articolata di questo effetto).

Poniamo un **Nodo Blur** tra l'immagine in uscita da **Render Layers** e la porta di input **Image** di **Composite**, specificando dei valori di base per X e Y (la dimensione del **blurring**).

Inseriamo anche un **Nodo Time**, **invertendo** la diagonale, in modo da avere valore massimo (1.0) al primo frame e valore minimo (0.0) all'ultimo frame (che va specificato nel campo **End** del **Nodo**).

Il valore ottenuto con **Time** non è però certo un buon fattore di moltiplicazione, per cui tra **Time** e la porta di input "**Factor**" del **Nodo Blur** precedentemente creato inseriamo un **Nodo Math** con funzione **Multiply**, ponendo il valore ad esempio a 10; in questo modo, l'output di **Time** varierà tra 10 e 0 e **scalerà linearmente** le dimensioni della **maschera di blurring** definita in **Blur**.

La configurazione del **Node Editor** appena descritta è visibile nell'immagine seguente.



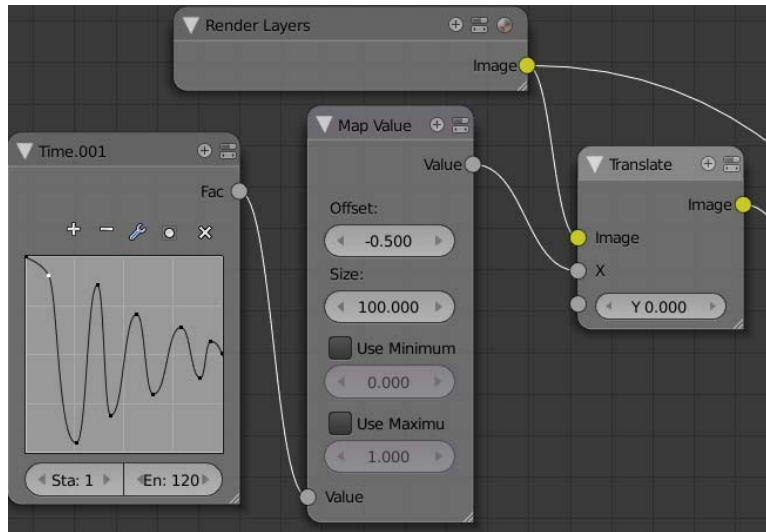
L'ubriaco (sfocatura e immagini sdoppiate).

A partire dall'effetto appena visto della **sfocatura animata** possiamo realizzarne uno ancora più interessante con **immagini sdoppiate**, **traslate** e **sfocate** che, dopo un certo numero di frames, si ricongiungono mettendosi perfettamente a **fuoco**.

Per realizzare questo effetto dobbiamo far uscire dalla porta di output **Image** del **Render Layer** due connessioni: la **prima** servirà per l'immagine **fissa**, **sfocata**, di sfondo, mentre la **seconda** servirà per l'immagine **sfocata** e in **movimento**; per il momento, comunque, vediamo quali sono i **Nodi** che ci servono per la **traslazione animata**.

Inseriamo all'interno del **Node Editor** un **Nodo Time**, un **Nodo Map Value** e un **Nodo Translate**, collegandoli in **sequenza** (collegando l'output di **Map Value** alla porta **X** di **Translate**).

Invertiamo la curva del **Nodo Time** e facciamole assumere un andamento **oscillatorio smorzato** che deve terminare con valore **Y** pari all'incirca a **0.5** (vedremo presto perché), inserendo alcuni punti di controllo e spostandoli a piacere, come mostrato nella figura seguente.



Configuriamo il **Nodo Map Value** con **offset** pari a **-0.5** (per cui il valore al termine dell'animazione sarà 0.0) e **Size** ad esempio **100** o **160**, a seconda di quanto volete accentuare l'effetto.

Alla porta di input **Image** di **Translate** colleghiamo una delle immagini in arrivo direttamente dal **Render Layer**.

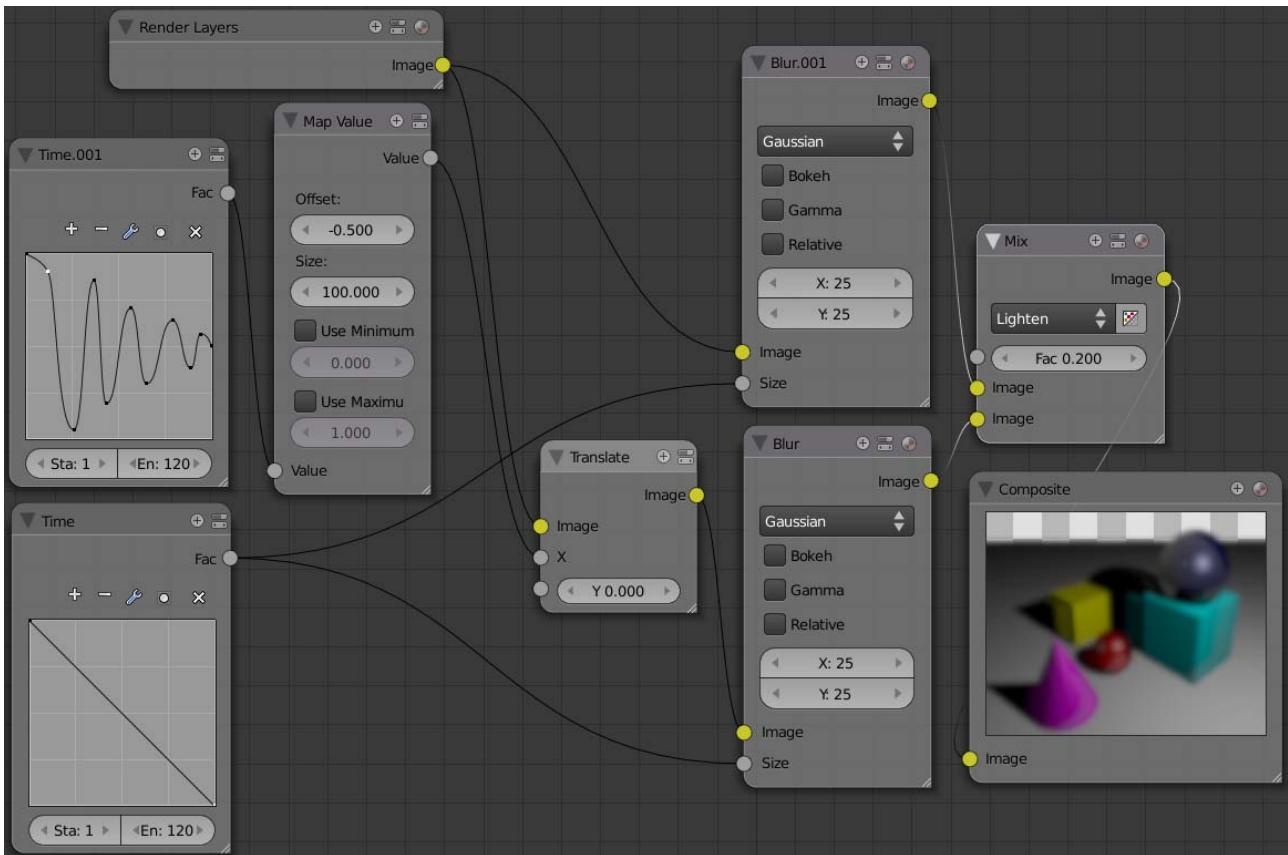
Per regolare l'effetto di **Blurring**, inseriamo un altro **Nodo Time**, **invertendone** la curva, all'interno del **Node Editor**; questo **Nodo** controllerà i canali **Size** dei due **Nodi Blur** da inserire.

Anche se l'effetto di **blurring** da applicare alle due immagini (la prima in arrivo direttamente dal **Render Layer**, la seconda in arrivo da **Translate**) è praticamente lo stesso, dobbiamo inserire due **Nodi Blur** con le stesse impostazioni (ad esempio, 25 sia per **X** che per **Y**): uno collegato a **Render Layer** e l'altro a **Translate**.

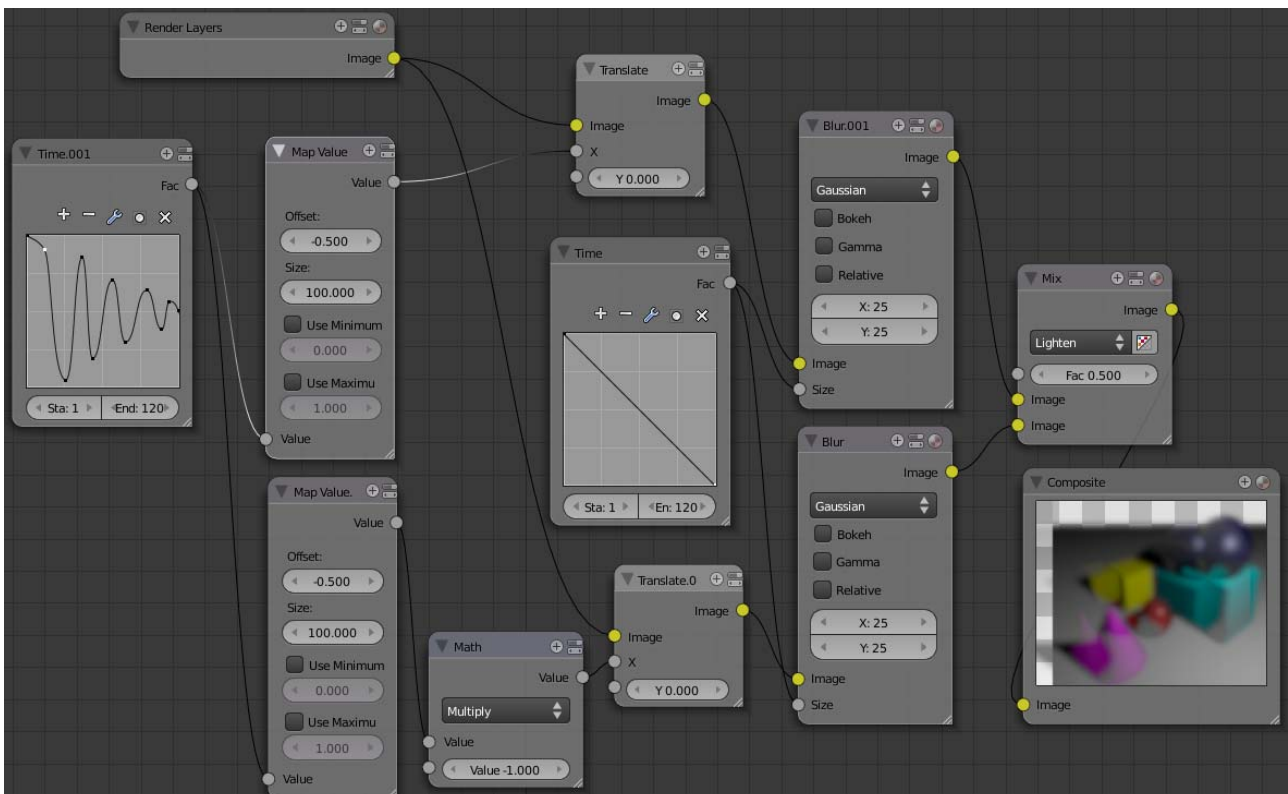
Per far **variare** nel tempo il valore del **blurring**, fino ad annullarlo al termine dell'animazione, colleghiamo l'output dell'ultimo **Nodo Time** creato alle porte di input **Size** dei due **Nodi Blur**.

Abbiamo quindi le due **immagini sfocate**, dobbiamo solo **mescolarle** insieme: aggiungiamo un **Nodo Mix** con funzione **Lighten** e **Factor** basso, ad esempio 0.2, collegandogli in input le immagini in arrivo dai due **Nodi Blur** e collegando l'output direttamente al **Nodo Composite**.

L'immagine riportata nella pagina seguente mostra la configurazione del **Node Editor** così ottenuta.



Una variante di quest'effetto consiste nel **traslare, animando** lo spostamento nel tempo, anche l'immagine di **sfondo**, utilizzando ad esempio la stessa **curva temporale** vista precedentemente ma **invertendone** i risultati. Tale **schema** è visibile nella figura seguente.



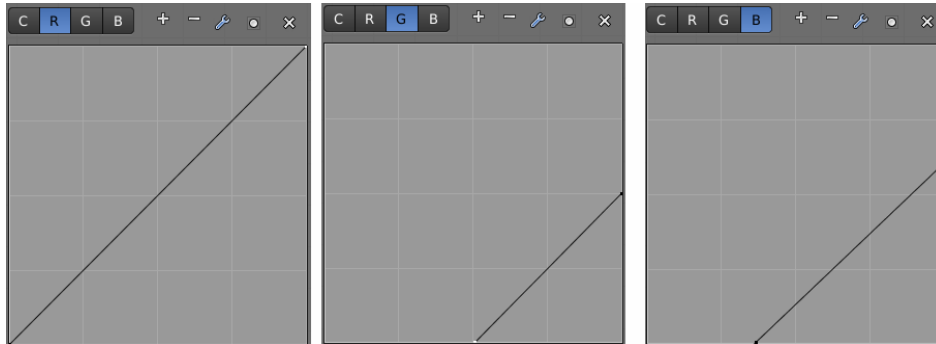
* * *

Atmosfere più fredde o più calde con le curve RGB.

Per conferire un'atmosfera o una **tonalità** più calda o più fredda ad una **scena**, è possibile ad esempio modificare le **curve colore (RGB)** dell'immagine.

Per effettuare questa operazione è sufficiente inserire un **Nodo RGB Curve** tra l'immagine restituita in **Output** da **Render Layers** e il canale di input **Image** di **Composite**.

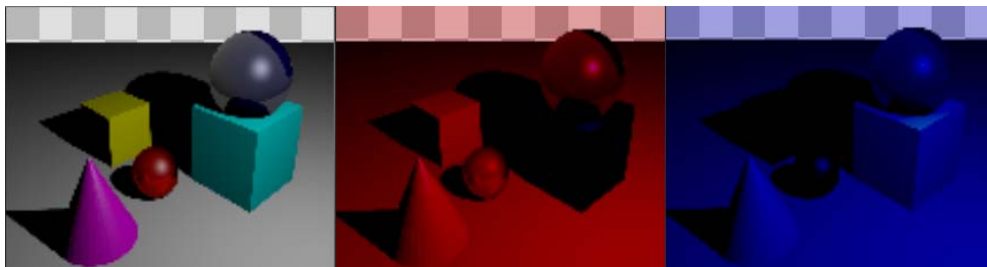
Per avere **atmosfera** più calde, si consiglia di “traslare” lungo l'**asse X**, verso destra, le **curve** (beh, le diagonali) dei canali **G** e **B**, traslando un po' di più **G** rispetto a quanto fatto con **B** e lasciando immutata la curva del **rosso**, come visibile nella figura seguente.



Tale operazione consiste quindi nell'annullare (linea piatta) i valori in input bassi per **G** e **B** e nel limitare superiormente i valori in uscita.

Per ottenere **atmosfera** o tonalità più **fredde**, invece, si consiglia di traslare (come descritto precedentemente) la curva del **Rosso** e un po' meno quella del **Verde**.

L'immagine seguente mostra l'immagine originale e due **rendering** ottenuti rispettivamente rendendo più calda e più fredda l'immagine con le **tecniche** appena descritte.



Questo è solo un modo per **cambiare l'atmosfera** generale di una **scena** operando sui colori: **Blender** mette a disposizione molti **Nodi** per controllare luminosità, contrasto, saturazione, tonalità ed altri **parametri**.

* * *

Fornire il bagliore GLARE solo a determinati oggetti della scena

Come abbiamo visto nei capitoli precedenti, grazie al **Nodo ID Mask** e alla proprietà **ID Pass** degli oggetti, è possibile assegnare un **identificativo numerico** agli **oggetti** della scena ed ottenere un'immagine-mappa della trasparenza (**Alpha**) per separarli dal **rendering** finale.

In questo **tutorial** vedremo come applicare l'effetto **Glare** (ma il procedimento è lo stesso per qualsiasi effetto) solo ad un oggetto.

Assegniamo all'oggetto da isolare un **ID Pass** diverso da quello degli altri (ad es.: 1), come discusso nel **capitolo 7**, e selezioniamo la casella **Object Index** in **Layers-Passes**.

Del **Nodo Render Layers** ci servono i canali di output **Image** e **IndexOB**. Colleghiamo il canale **IndexOB** ad un **Nodo ID Mask**, impostando nello stesso il valore dell'**ID** dell'oggetto da isolare (nel nostro caso, 1).

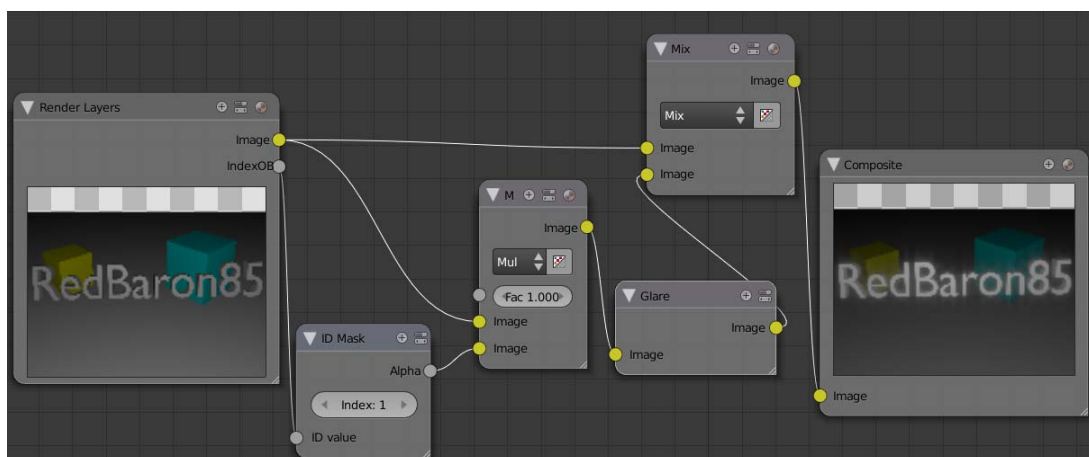
L'output di **ID Mask** è un'immagine-mappa della trasparenza (**Alpha**) dove solo i pixel dell'oggetto isolato sono a 1, mentre tutti gli altri sono a 0.

Per ottenere una nuova immagine **RGBA** isolando l'oggetto dall'originale, inseriamo un nuovo **Nodo Mix** con operazione di moltiplicazione e **Factor** a 1, collegandogli in input l'immagine in arrivo da **Render Layers** e la maschera ottenuta con **ID Mask**.

Moltiplicando un'immagine per una mappa dell'**Alpha** otterremo una nuova immagine identica alla prima ma **opaca** solo in corrispondenza dei **pixel bianchi** della mappa **Alpha** in input.

All'immagine ottenuta in output da questo **Nodo Mix** possiamo quindi collegare il **Nodo** dell'effetto che vogliamo implementare (nel nostro caso, **Glare**) e collegare l'output di quest'ultimo ad un nuovo **Nodo Mix**, che questa volta avrà come operazione l'addizione e come **Factor** 0.5.

L'output di questo **Nodo Mix** potrà essere quindi collegato ad altri **Nodi** o direttamente a **Combine**.



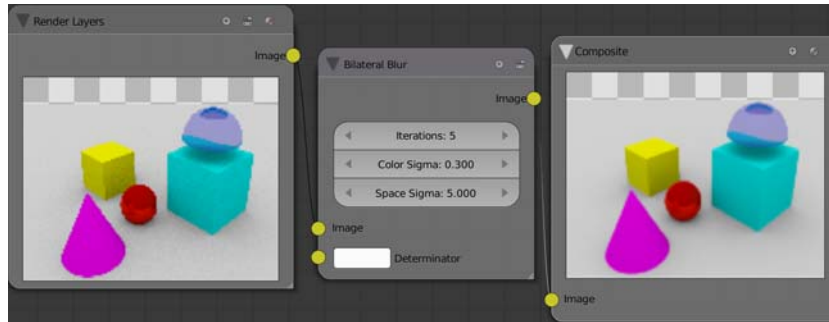
* * *

Bilateral Blur per i rendering AO

I **Rendering** prodotti con **Ambient Occlusion**, specialmente in modalità **Raytrace**, sono spesso “rumorosi”.

In questo caso, il filtro di **blurring** consigliato è **Bilateral Blur**, in quanto effettua una correzione **adattiva** dell’immagine a seconda della presenza di spigoli, rumore e dettagli.

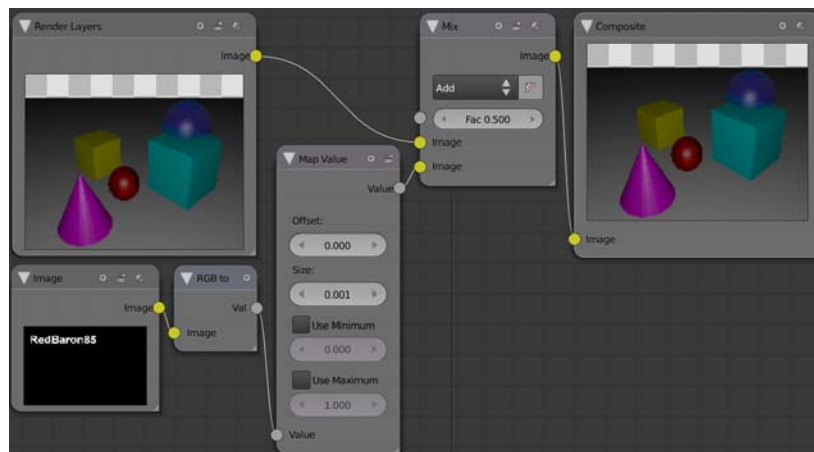
Il più delle volte le impostazioni di default per **Color Sigma** e **Space Sigma** andranno più che bene, per cui è necessario impostare il numero di **iterazioni** ed eventualmente il **Determinator**.



* * *

Watermarking delle immagini

Per “firmare” le proprie immagini (**watermarking**), è possibile aggiungere all’immagine di partenza l’immagine con la **firma**, opportunamente trattata; in particolare, dovremo caricare l’immagine con la firma in un **Nodo Image**, convertirla in **scala di grigi** (con un **Nodo RGB to BW**), ridimensionarne il valore con un **Nodo Map Value** (ad esempio, con Size a 0.001) e **sommarla** all’immagine originale con un **Nodo Mix---Add**.



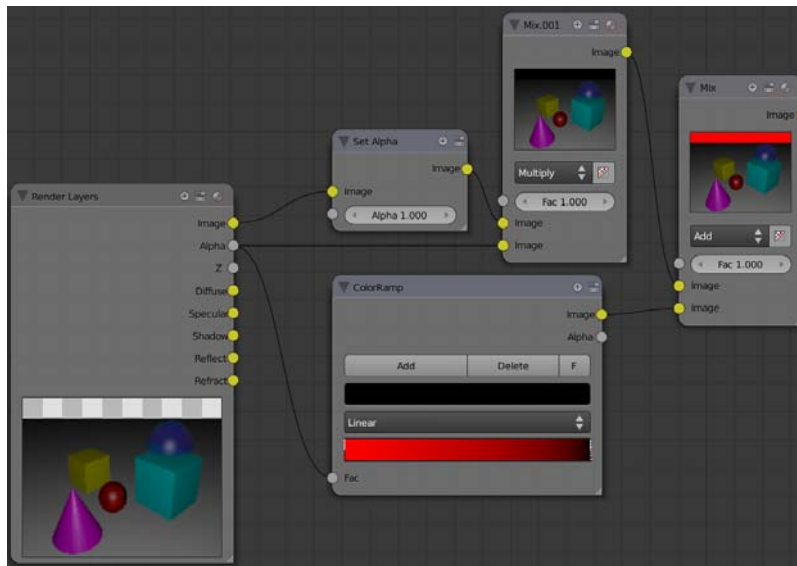
* * *

Cambiare, colorandolo, lo sfondo trasparente

Negli esempi visti fino a questo punto, lo **sfondo** è sempre **trasparente**.

Possiamo **colorare** di un’unica tinta lo **sfondo** utilizzando la **Color Ramp** sul canale **Alpha** per cambiare il **colore** delle aree nere (trasparenti) e bianche (opache) e **mixare** l’immagine ottenuta con **Mix---Add** (attenzione a **Factor**, che deve valere 1) con l’immagine in uscita da **Render Layer** con l’**Alpha** portato a 1 mediante un **Nodo Set Alpha**.

L'immagine seguente illustra quanto detto.



Ovviamente è possibile raggiungere lo stesso scopo con altri **Nodi** e altre combinazioni, questo è solo uno dei modi possibili.

* * *

Rendering Stereoscopico (basi)

In **Blender** è possibile realizzare degli **stereogrammi**, immagini realizzate con tecniche di **stereoscopia**. Per creare gli stereogrammi dovremo produrre il **rendering** di una **scena** da **due punti di vista** diversi; questo effetto si ottiene grazie a due telecamere i cui piani immagine sono posti sullo stesso piano, mentre il **punto di vista** è **traslato** di poche unità¹.

Per visualizzare queste immagini in **3d** con i classici **occhialini colorati**, infine, dovremo **cambiarne i colori**: vediamo come fare.

Dopo aver realizzato la scena da **renderizzare**, pronta per il **Rendering**, creiamo una seconda **Scene** all'interno dello stesso file **.blend** **copiando** al suo interno tutti gli elementi presenti nella prima **Scene**, telecamera inclusa; **spostiamo** però leggermente quest'ultima lungo l'**asse X locale**, in modo da mantenere il suo piano immagine sullo stesso piano di quella presente nella scena principale, ma col punto di osservazione leggermente spostato, come già detto. In questo modo stiamo guardando **la stessa scena** da un **punto di vista** leggermente diverso.

Assicuriamoci di aver attivato l'uso dei **Nodi** per il **Compositing** nel **Node Editor** di ciascuna delle due **Scenes** e torniamo al **Node Editor** della prima **Scene**.

I due **Render Layers** associati alle due **Scenes** producono in output, sul canale **Image**, i due **Rendering**, pronti per essere fusi nell'**immagine stereoscopica**.

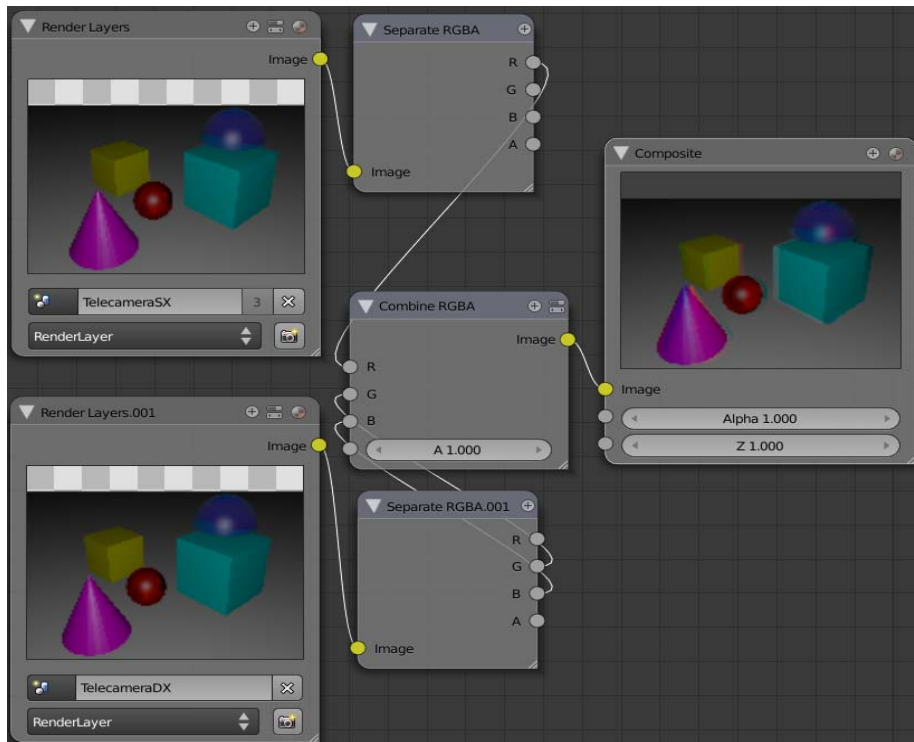
Per effettuare correttamente quest'operazione, dobbiamo **rimuovere il blu e il verde** dall'immagine di **sinistra**, mentre dall'immagine di **destra** dobbiamo **rimuovere il rosso** (questo, almeno, per vedere l'immagine con occhialini con lenti dai colori complementari a quelli detti, ossia **lente rossa** per l'occhio **sinistro** e **ciano** per l'occhio **destro**); per far ciò, utilizziamo i nodi

¹ Tuttavia, il calcolo del valore esatto, così come altri elementi di teoria della stereoscopia, esula da questa trattazione.

Separate RGB sulle immagini ottenute in output dai due **Render Layers**, inseriamo un **Nodo Combine RGB** e colleghiamo l'output **R** che arriva dalla telecamera **sinistra** e gli output **verde e blu** che arrivano dalla telecamera **destra** alle relative porte di input del **Nodo Combine**.

L'output del **Nodo Combine** è l'immagine stereoscopica e può essere modificata ulteriormente, se lo desideriamo, o inviata direttamente al **Nodo Composite**.

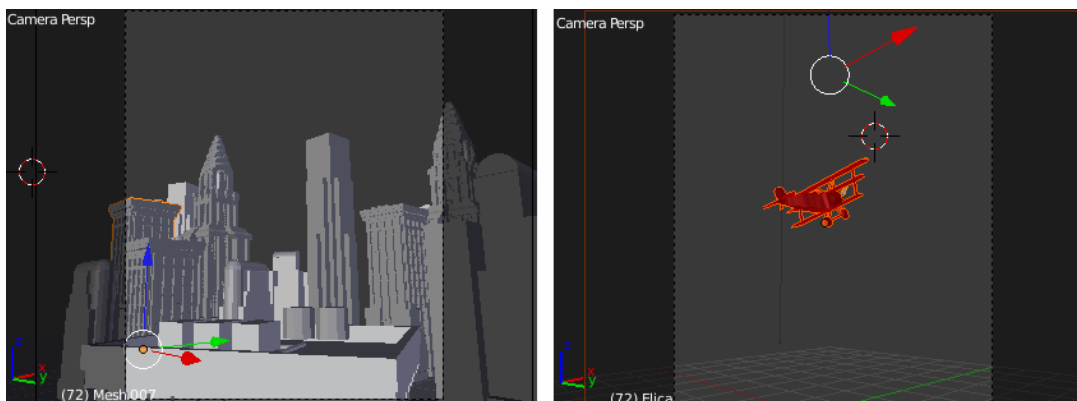
L'immagine seguente mostra lo schema nel **Node Editor** per la configurazione appena descritta.



* * *

La copertina di questo ebook

Per realizzare l'immagine utilizzata come **copertina** di questo **ebook**, ho creato due **Scenes** all'interno dello stesso file **.blend**, inserendo nella prima **Scene** ("Metropoli") i modelli 3D dei grattacieli e, nella seconda **Scene** ("Aereo") il modello 3D del Fokker DR-1, l'aereo del **Barone Rosso**.



Ovviamente, in entrambe le **Scenes** il compositing con i **Nodes** è attivo e le **telecamere** si trovano nella **stessa posizione**; cambiano, invece, i **metodi di illuminazione**: nessuna fonte di luce, solo **Ambient Occlusion**, in **Metropoli**, mentre in **Aereo** vi è una fonte di luce **Point** posta in alto, sopra il modello.

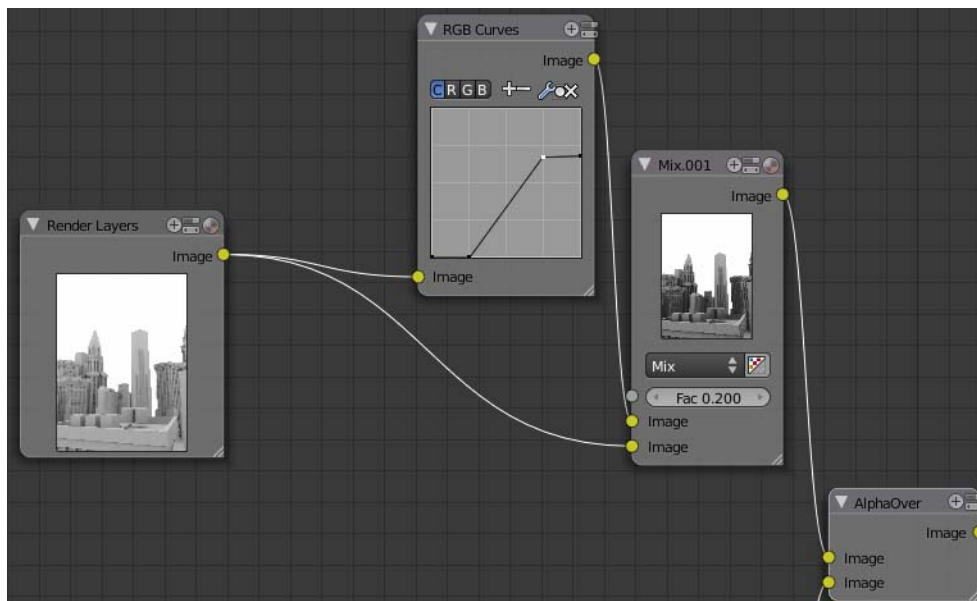
In **Metropoli**, il cielo di **sfondo** è **bianco**, mentre in **Aereo** lo **sfondo** è **trasparente**: l'immagine (filtrata) dell'aereo verrà infatti applicata in **sovrimpressione** con un **Nodo AlphaOver** all'immagine (filtrata) della città.

L'output prodotto con **AlphaOver** verrà quindi inviato direttamente al **Nodo Combine**.

Sono presenti ovviamente **due Render Layer**, uno per **Scene**.

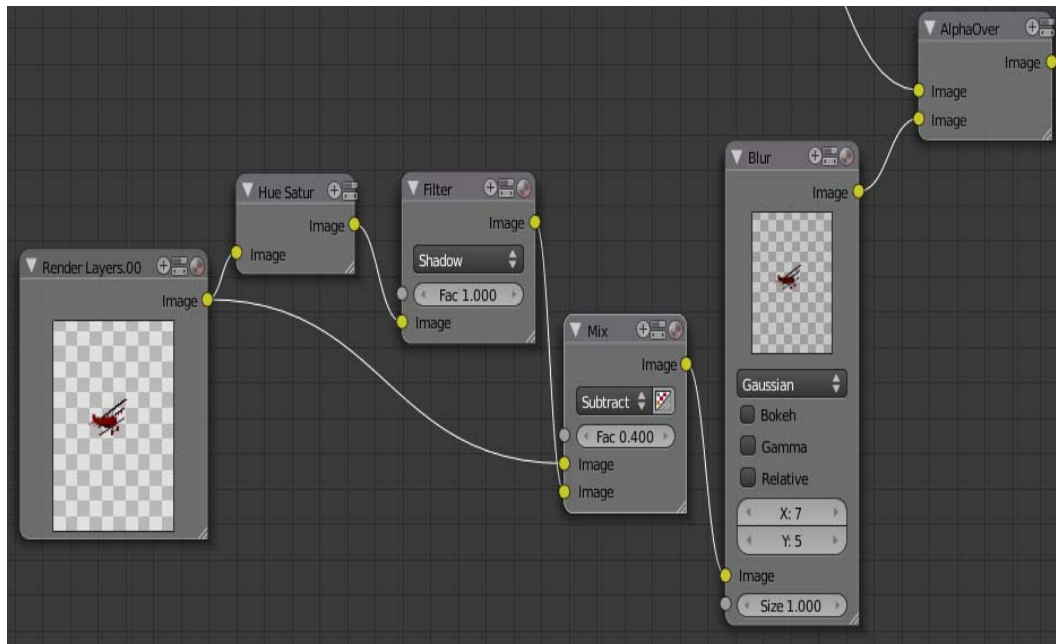
Il **Render Layer** di **Metropoli** ha, sulla porta di output **Image**, due connessioni in uscita: la prima va ad un **Nodo RGB Curves** dove la curva **C** (composita) assume una forma quasi a scalino, per realizzare un effetto **soglia**, mentre la seconda connessione va ad un **Nodo Mix---Mix** con **Factor** 0.2, che mescola l'immagine originale con quella ottenuta dal **Nodo RGB Curves**.

Il percorso dell'immagine di sfondo dal **Render Layer** al **Nodo AlphaOver** è visibile nell'immagine seguente.



Anche dalla porta di output **Image** del **Render Layer** relativo alla **Scene Aereo** escono due connessioni: la prima porta l'immagine ad un **Nodo HSV** per variare tonalità, saturazione e intensità, poi ad un **Nodo Filter-Shadow** con **Factor** a 1 per accentuare gli spigoli; l'immagine così ottenuta viene inviata ad un **Nodo Mix-Subtract** con **Factor** 0.4 per essere appunto sottratta all'immagine originale, in arrivo direttamente dal **Render Layer**.

Il percorso dell'immagine dell'aereo dal **Render Layer** al **Nodo AlphaOver** è visibile nell'immagine riportata nella pagina seguente.



* * *

* * *
* * *

