

Numero 10 - Febbraio 2011

BLENDER

Magazine Italia

COLOR CORRECTION
Preview e ispirazione

FAKE
Tecniche di illusione
nel mondo 3d

MAKING OF...
Bad Fellà e
Fermati tempo!



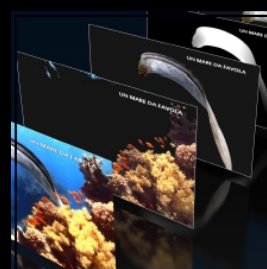
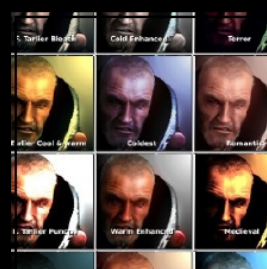
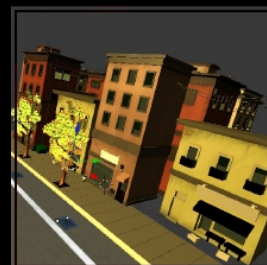
News	4
Fake: illusioni nel 3D	7
Bad Fellà - Making of	11
Bump e Normal Map	16
Color Correction	21
Un mare da favola	28
Trasparenze e alpha	33
Fermati Tempo!	37
Contest copertina BMI11	41
Gallery dell'ultimo contest	42
Gallery	43

DISCLAIMER

Blender Magazine Italia non rappresenta una testata giornalistica in quanto viene aggiornato senza alcuna periodicità. Non può pertanto considerarsi un prodotto editoriale ai sensi della legge n. 62 del 07/03/2001.

Gli autori non hanno alcuna responsabilit  sui contenuti dei siti in collegamento, sulla qualita' o correttezza dei dati. Essi si riservano la facolta' di rimuovere le informazioni, fornite da terzi, ritenute offensive o contrarie al buon costume.

Le immagini sono correlate agli argomenti di cui si scrive. Alcune, sono provenienti da Internet e quindi valutate di pubblico dominio. Qualora i soggetti proprietari fossero contrari alla pubblicazione non dovranno far altro che segnalarlo in modo da poter procedere ad una rapida eliminazione.



In questo numero

Questo numero inizia con le consuete news dal mondo Blender, ma non solo: viene segnalata la svolta opensource di Lightworks; sono online video e materiale dell'ultima Blender Conference; si danno informazioni sui prossimi open movie della Blender Foundation...

"Fake: illusioni nel 3D" è il primo di una serie di articoli di Anfeo riguardanti tecniche di illusione. Come esempio pratico del camera mapping si mostra una via con alberi e case sullo sfondo. La spiegazione continua con l'utilizzo dell'UV projection e altri piccoli trucchi per ottenere un effetto credibile.

Una delle immagini più accattivanti ed espressive che si possono trovare nei forum di Blender in Italia degli ultimi tempi è "Bad Fellà" di waaf. L'autore ci spiega come l'ha realizzata, partendo dalla modellazione, utilizzando poi lo sculpt, creando successivamente le diverse UV map per ottenere il render finale.

Chi vuole conoscere le differenze tra bump map e normal map, il loro funzionamento, i modi per realizzarle e i loro campi di impiego, può leggere l'articolo di Lell.

Chi invece è interessato a sperimentare set di nodi con il compositor di Blender per ottenere interessanti e precisi effetti di postproduzione, sarà soddisfatto leggendo l'articolo di Fernando Luceri, tanto più che l'autore rende liberamente scaricabili dal suo sito i settaggi descritti.

Insieme a Riccardo Covino ci imbarchiamo per il secondo fantastico viaggio virtuale in "Un mare da favola". Chi ha partecipato al primo, sicuramente non vorrà perdere l'occasione di prendere parte a questa appassionante esperienza.

Il tutorial di Francesco Milanese spiega approfonditamente, come si evince dal titolo, i concetti di Alpha, trasparenze Z-Transp e Raytrace con Blender 2.5

L'idea, la modellazione, i materiali e il rendering dell'immagine di copertina di questo numero sono illustrati dall'autore in un esauriente Making of.

A conclusione di Blender Magazine Italia 10, sei selezionate immagini per la gallery finale.

Blender Magazine Italia
numero 10
anno 2011

Responsabili:

Luca Pinciani (Sinistar)
Alfonso Annarumma (Anfeo)

Collaboratori:

Gikkio

Grafica:

Davide_G

Siti di riferimento:

www.blender.it
www.kino3d.com
www.blender-tutorial.com
www.blender.org

Software utilizzati:

Blender
Scribus
Pdftk
The Gimp
Openoffice



In copertina:
"Fermati tempo!"
di Pasquale Guida



Blender: le news

Ultime news dal mondo Blender e non solo

di Luca Pinciani

Prossimi open movie della Blender Foundation

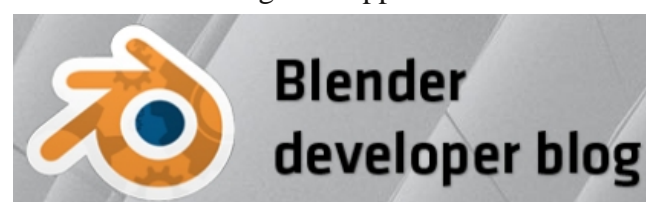
Dal suo profilo di Twitter, Ton Roosendaal ha lasciato alcune dichiarazioni relative ai prossimi open movie della Blender Foundation.

A proposito di "Mango", il prossimo progetto della BF, Ton ha detto che stanno lavorando sulla pre-pre-produzione del progetto e che l'Amsterdam film academy potrebbe aiutare economicamente la realizzazione del progetto. Il progetto dovrebbe iniziare in settembre per produrre un cortometraggio non interamente in 3D ma live action più effetti speciali.

Oltre a queste informazioni, Ton ha anche parlato del progetto successivo a Mango: molto più ambizioso. Il nome sarà "Gooseberry", ovvero il nome inglese di un frutto di bosco, l'uva spina. Il progetto potrebbe produrre un film realizzato completamente in Blender dal Blender Institute con la collaborazione di alcuni studi grafici nel mondo.

Blog per gli sviluppatori di Blender

La Blender Foundation ha rilasciato un nuovo sito web dedicato agli sviluppatori di Blender.

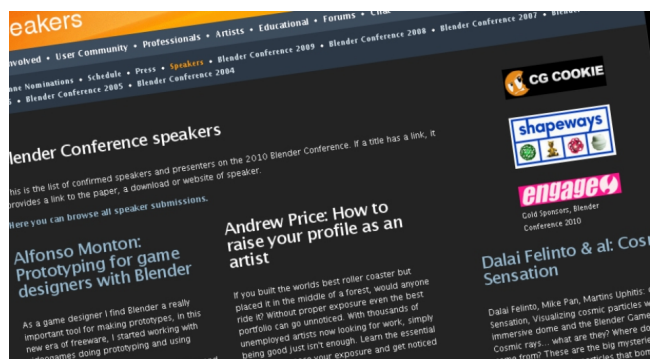


Si tratta di un blog raggiungibile all'indirizzo <http://code.blender.org>. Chiunque abbia accesso all'svn per bf-blender può ottenere un account per scrivere articoli sul sito. Ton Roosendaal si augura che il blog possa migliorare la comprensione di Blender, possa fornire una migliore documentazione e dare una visione del software a vecchi e nuovi sviluppatori.

Online le risorse della BC2010

Vi siete persi la Blender Conference 2010? Sono

online i video e buona parte del materiale usato dagli speaker. Come ogni anno, anche per questa edizione, gli interventi sono stati molti e interessanti. Dall'integrazione del multitouch in Blender 2.5 alla grafica dell'art director di Sintel, da come creare una render farm alla realtà aumentata, dall'utilizzo di Blender in un workflow con Adobe Flash al making of del corto "Froggy is back"... e tanto altro ancora!



Potete vedere tutti i video della BC2010 nell'apposita playlist di YouTube:

<http://www.youtube.com/user/BlenderFoundation?feature=mhum#g/c/83F5B6F78A5BA316>

Il materiale degli speaker è invece visionabile su Blender.org:

<http://www.blender.org/community/blender-conference/speakers/>

Videogioco "Dead Cyborg"

Endi, un esperto utente di Blender, sta realizzando un curatissimo videogioco di una qualità molto elevata. Il gameplay è simile alle vecchie avventure testuali, ma la grafica è 3D.



Il giocatore può esplorare diversi ambienti, raccogliere oggetti ed eventualmente utilizzarli ai

fini del gioco. Lo sviluppo del videogioco è basato sulle donazioni degli utenti, per cui attualmente è uscito solo un video trailer visibile all'indirizzo <http://www.youtube.com/watch?v=euGnVNNfMhE&hd=1>

Lightworks è ora opensource



Come annunciato ad aprile dello scorso anno, da novembre 2010 Lightworks è opensource. Per chi non lo conoscesse, Lightworks è un potente e professionale software di video editing non lineare. Attualmente vi è solamente la versione per i sistemi Windows. Dal sito ufficiale www.lightworksbeta.com si possono raggiungere tutti i contenuti legati al software. Dalla sezione "Downloads" è ovviamente possibile scaricarlo, ma anche trovare guide per iniziare subito ad utilizzarlo. C'è inoltre la sezione "Forum" con post nuovi quasi ogni giorno. Per chi è ancora indeciso e provarlo o meno, può controllare le caratteristiche del software nella sezione "Features" oppure leggere la sezione "Spotlight" che racconti di alcuni film prestigiosi per i quali si è utilizzato Lightworks.

Creare un videogioco per Android con grafica in Blender

Un interessante articolo di Rubén López spiega brevemente ma in modo chiaro e preciso come creare un videogioco 2D per Android utilizzando

Blender per disegnare la grafica. L'articolo è nato dall'esperienza pratica dell'autore che ha realizzato "Egg Savior", un simpatico videogioco con protagonista un gallo. Quest'ultimo, come spiegato nell'articolo, è stato modellato e animato in Blender per ottenere delle sprite 2d per il videogioco. Ecco il link all'articolo:

<http://organicandroid.blogspot.com/2010/11/creating-2d-games-with-android-and.html>

Renderer Mitsuba



Un nuovo motore di render si affaccia sulla scena opensource: Mitsuba, rilasciato sotto licenza GPL. E' disponibile per Windows, Linux e OSX e tramite un plugin è possibile utilizzarlo con Blender.

Si può utilizzare sia da una interfaccia con linea di comando e sia da frontend più intuitivo.

Lunga è la lista delle features supportate: per saperne di più, il sito ufficiale è www.mitsuba-renderer.org

Altre news in breve:

- Ad inizio gennaio è stata rilasciata l'ultima versione di Blender, la 2.56a beta: poche le novità ma molti i bug fix.

- Sono usciti due nuovi libri in italiano su Blender: "Materiali e Textures in Blender 2.5 Volume 1: i Materials" di Francesco Milanese e "Grafica 3D con Blender" di Francesco Siddi.

Inoltre, da pochi giorni, il libro "Blender per l'architettura" di Ilario De Angelis è opensource. Come si evince dal titolo, il libro tratta dell'utilizzo di Blender 2.49 in ambito architettonico.

- Burster è un interessante plugin per i browser

che permette di visualizzare all'interno di una pagina web dei contenuti realizzati in Blender, come, ad esempio, un videogioco che utilizza il Blender Game Engine. Buster è stato rilasciato con licenza GPL.



Per provarlo o vedere video di esempio, è possibile visitare il sito ufficiale www.geta3d.com

- Nuova versione, sia per Windows che per Linux, per l'ottimo Voodoo Camera Tracker: <http://www.digilab.uni-hannover.de/docs/manual.html>



Fake: illusioni nel 3D

Tecniche di illusione nel mondo del 3D

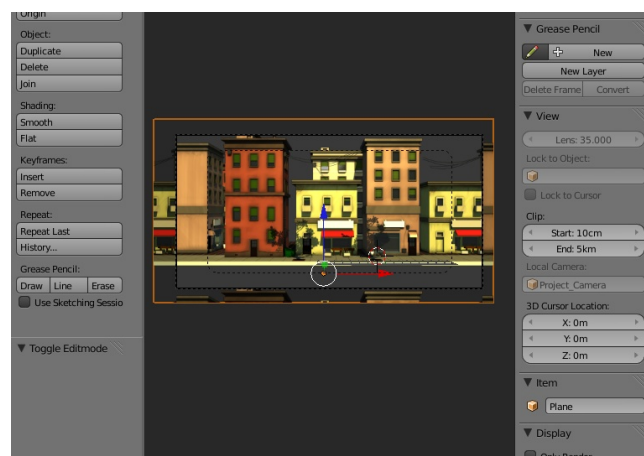
di Anfeo

Quando si lavora con la grafica digitale bisogna tenere conto di elementi di finzione o illusori. Il primo grande inganno a cui sottoponiamo chi guarda le immagini e i video che creiamo è la falsa concezione di tridimensionalità nei nostri lavori. Ricordiamoci che il mezzo con cui trasmettiamo è sempre bidimensionale, monitor, TV, cinema, stampe ecc... Ok, esiste il cinema 3D, i videogiochi, le partite, ma anche qui alle volte vengono usati stratagemmi per far sembrare l'immagine in 3D, però per non complicare troppo il discorso non ne terremo conto.

Questo sarà il primo di una serie di articoli dedicati al mondo del FAKE, dove quello che vediamo in realtà è puro inganno, come il tunnel disegnato da Willie il Coyote sulla montagna per far schiantare Beep Beep... o aspettate, per Beep Beep era reale...

Camera Mapping

Uno dei modi più comuni di simulare qualcosa che non c'è, è quello di applicare un'immagine bidimensionale ad un elemento piatto e posizionarlo come sfondo. Questa tecnica si usa da sempre, dal teatro ai primi film fino a svilupparsi nell'odierna tecnica computerizzata del Camera Mapping.



Il Camera Mapping si può effettuare in molti programmi, ma noi vedremo come farla in Blender con un esempio pratico, tanto per capire in che modo ci può aiutare effettivamente in un nostro progetto.

NB: Si daranno per scontate alcune funzioni base di Blender per non appesantire ulteriormente la lettura.

Prima di tutto ecco di cosa abbiamo bisogno:

Un'immagine da proiettare.

Un oggetto che funzioni da proiettore.

Un oggetto sul quale viene proiettata l'immagine.

Abbastanza semplice, ma cosa succede esattamente? In pratica utilizzando il modificatore UV Projector, l'oggetto che riceve la proiezione crea un UV map che adatta l'immagine dal punto di vista dell'oggetto che proietta. Usando quindi una Camera possiamo spalmare ciò che vediamo attraverso di essa, sull'oggetto.

Questo è tutto quello che ci serve, perché la tecnica del Camera Mapping è proprio quello, proiettare un'immagine complessa su una superficie meno complessa, ma che a prima vista ci ricordi la profondità e la complessità che dimostra l'immagine proiettata.

La tecnica del Camera Mapping può essere usata con Rendering, ma anche con foto e illustrazioni. Viene utilizzata molto nel Cinema in accoppiata con i Matte Paint, ovvero i classici dipinti/fotomontaggi di paesaggi, che vengono proiettati su delle superfici solide e quando si sposta l'inquadratura ci danno quel senso di profondità tipica di un'immagine tridimensionale.

L'inquadratura infatti è fondamentale, perché da essa dipende la riuscita del Camera Mapping. Questa tecnica va pianificata in funzione della ripresa che vogliamo effettuare, perché si dovrà tenere conto del movimento della Camera per evitare sgradevoli sorprese.

Analizziamo quindi questo progetto per una scenografia di un esterno.

La ripresa di questo esterno prevede una carrellata di una serie di palazzi che termina in una palestra.

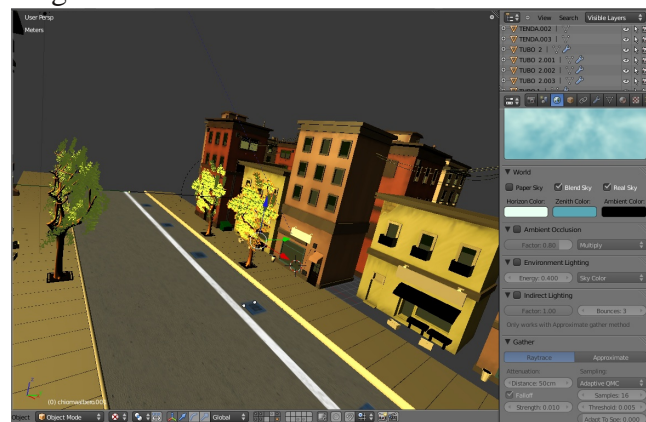
Siccome la complessità dei palazzi e dell'ambiente esterno in generale renderebbero troppo lunghi i tempi di rendering si è deciso di usare un semplice piano sul quale mostrare il rendering dei palazzi mentre la Camera fa la sua carrellata.

Inserire link a video.

Utilizzando il Camera Mapping la cosa si è evoluta un po' e da qualche palazzo ora si ha l'impressione di essere in una città.

Il Rendering.

La modellazione dell'isolato è opera di Gikkio su concept di SandroP, mentre il texturing è stato eseguito da Dawith.



Dopo aver posizionato una Sun di un giallo molto caldo, ho posizionato la Camera per eseguire poi un rendering con Ambient Occlusion.

I due alberi sono stati renderizzati su un Layer a parte, vedremo in seguito il perché.



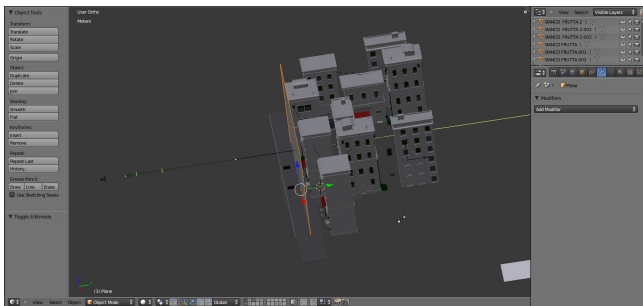
Alla fine abbiamo due immagini di 4000x2000px in formato PNG con canale Alpha.
(in figura Gli Alberi)



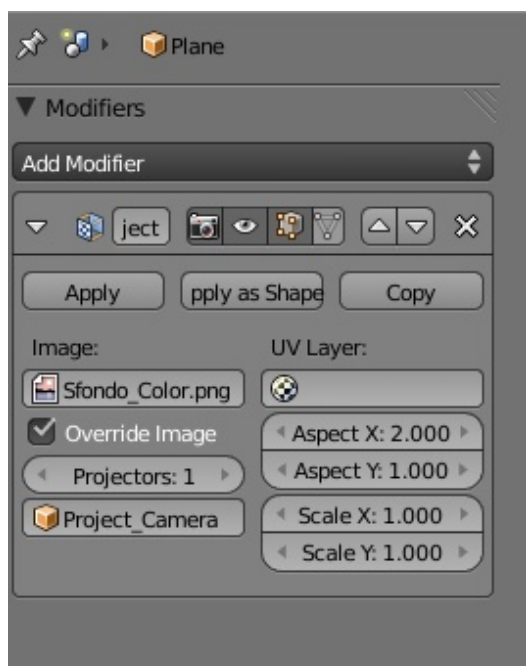
(in figura lo Sfondo)

UV Project

Utilizzando come proiettore la stessa Camera usata per il rendering ho creato un Plane e gli ho applicato il modificatore UV Project e l'ho posizionato circa all'altezza del marciapiedi, nascondendo i palazzi in un altro livello in quanto ora superflui.

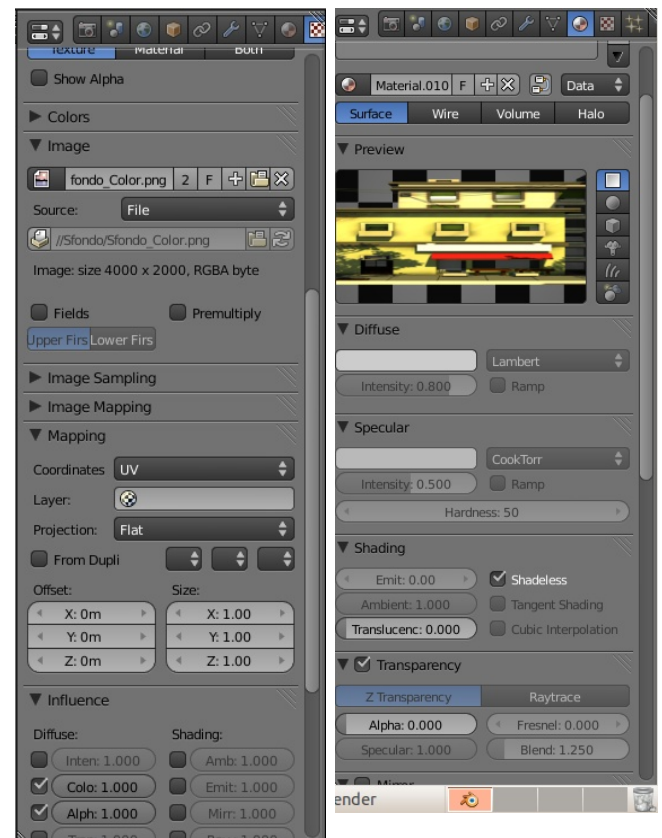


Settare l'UV Project è abbastanza semplice, dobbiamo scegliere l'oggetto che proietta (la Camera) e l'immagine da proiettare.



Siccome l'oggetto che proietta è la Camera, se non abbiamo alterato la scala di essa, Blender la considera come un quadrato, quindi dobbiamo alterare l'Aspect Ratio del modificatore. L'immagine è 4000x2000px quindi il rapporto è 2:1, quindi inseriamo 2 nel campo AspX.

Creo un materiale Shadless con Alpha=0 per il Plane gli ho applicato come texture in modalità UV il rendering dei palazzi, sia su Color che su Alpha.



Andando in modalità GLSL possiamo vedere in anteprima le texture con l'Alpha direttamente in 3D view.

Ora l'immagine è mappata sul Plane proprio come la si vede in Vista Camera. Se così non fosse, probabilmente dovete eseguire un Unwrap del Plane.

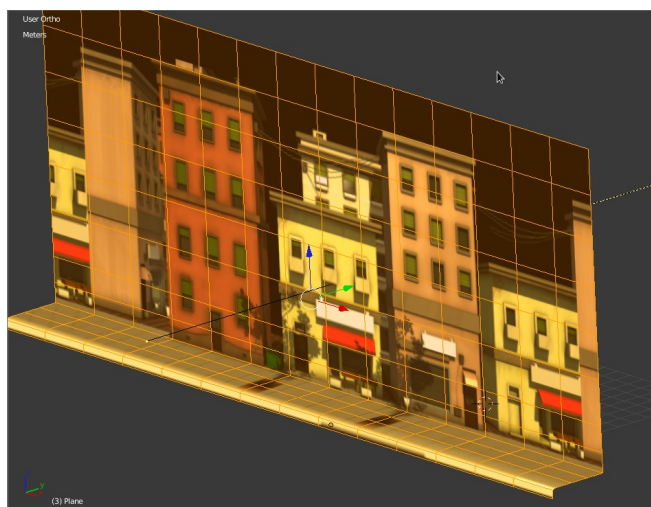
Dare profondità

Il trucco non funziona ancora, manca un tocco di profondità, quindi ho estruso in fuori la parte del marciapiedi.

L'immagine non viene distorta, ma continua ad essere proiettata dalla Camera, cosa che non sarebbe successa con una semplice UV map.

L'unico difetto del modificatore UV Project è che se l'immagine inizia a prendere forme strane basta semplicemente aggiungere qualche loop alla mesh e tutto si sistema.

(in figura: Errore UV)

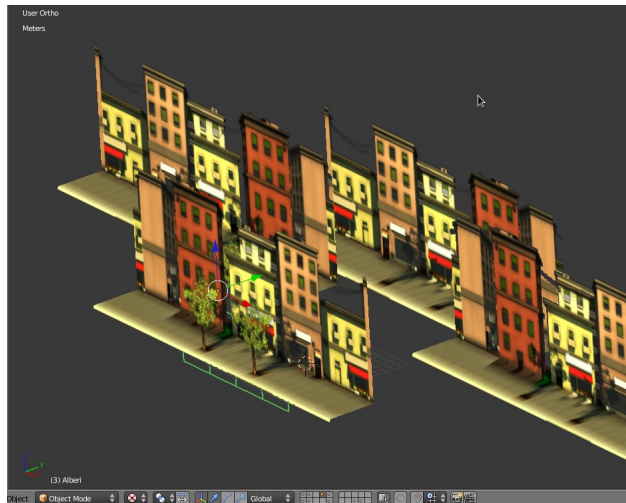


(in figura: Errore UV Corretto)

Volendo è possibile modellare anche i palazzi per dare loro un tocco di profondità in più, ma vista la semplicità della ripresa non è strettamente

necessario.

Alla fine è bastato creare un plane anche per gli alberi e poi rendere più popolosa la città duplicando semplicemente la fila di palazzi.



Ora abbiamo un'immagine 2D che sembra 3D, in più ci sono tutta una serie di vantaggi:

I tempi di rendering sono ridotti drasticamente, il raytracing non è interessato dal materiale Shadless quindi non vengono calcolate ombre e Ambient Occlusion per quegli oggetti.

C'è a disposizione un'immagine 2D da 4000x2000px da modificare direttamente in Gimp, possiamo aggiungere delle scritte alle insegne o altri particolari che avrebbero richiesto un certosino lavoro di texture e materiali.

Questa tecnica ha i suoi limiti, ad esempio non è possibile ruotare l'inquadratura perché altrimenti si capirebbe subito che lo sfondo è finto. Infatti come detto sopra, tutto richiede un'attenta pianificazione e determinare prima come sarà l'inquadratura e dopo scegliere il tipo di oggetto da modellare.

Non mi sono dimenticato della strada, in realtà per essa ho utilizzato la tecnica del Bake Full Render, che spiegherò un'altra volta.



Bad Fellà - Making Of

Un progetto completo, dalla modellazione al rendering finale

di Waaf

Salve a tutti!

Molti di voi mi conoscono come “waaf” nelle community italiane. Non sono bravo in preamboli e presentazioni varie, quindi perché non iniziare subito?

Navigando in rete per trovare ispirazione per modellare qualcosa mi sono ritrovato in una delle migliori pagine per quello che riguarda references, in particolar modo di natura fantasy-horror <http://www.characterdesigns.com/>

Qui ho scelto il soggetto che mi interessava. Avevo in mente qualcosa che ricordasse sia il Jack Nicholson di Shining che il Malcolm McDowell di Arancia Meccanica.

Scaricate le references ho iniziato a modellare poligono per poligono, fino ad ottenere la giusta forma ed una buona qualità dei poligoni (fig 1 e 1bis)

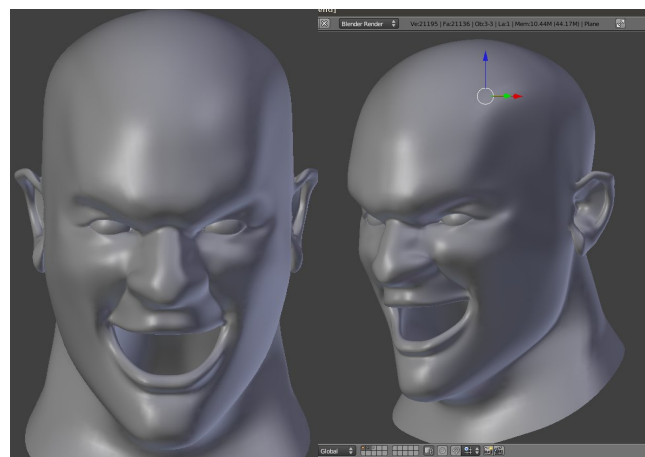


fig.1 e 1bis

Fatto questo ho iniziato a scolpire (fig. 2) usando la build specifica per lo sculpt che, anche se piena di bug, mi ha permesso di raggiungere buoni livelli di dettaglio (fig.3). Ho dato abbastanza importanza alla zona degli occhi, che di solito è

quella più comunicativa (fig. 4 e 5). A tal proposito ricordo che è fondamentale inserire due sfere provvisorie per semplificarsi di molto la gestione delle proporzioni e le fasce muscolari.

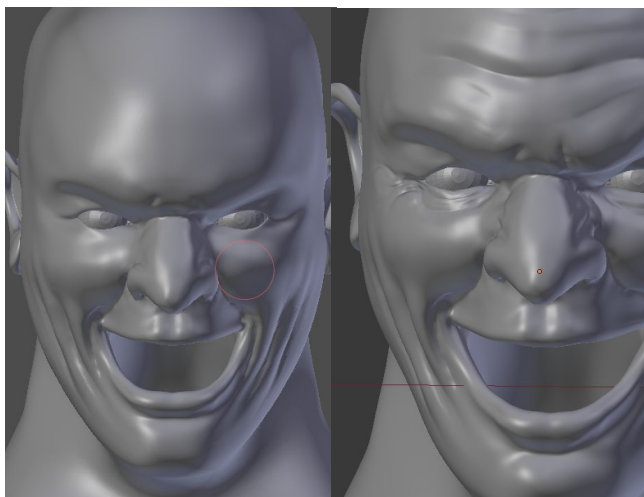


fig.2 e 3

E' possibile visionare timelapse sia del processo di modellazione che di quello di sculpting:

<http://vimeo.com/17431431>

<http://vimeo.com/17554503>.

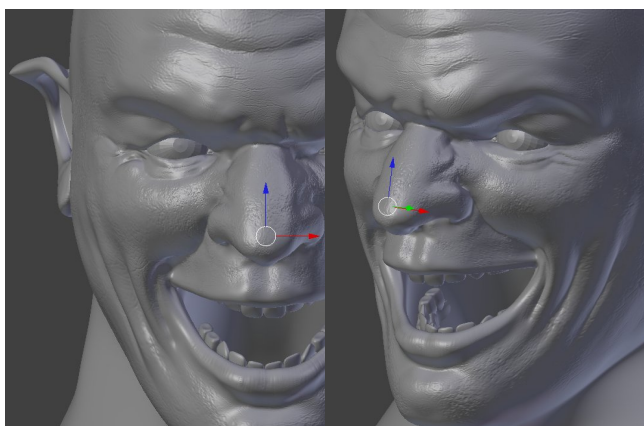


fig.4 e 5

Una volta terminata la modellazione del volto, ho creato i bulbi oculari nella classica versione con occhio interno e parte esterna trasparente. Per denti e gengive, invece, ho importato una mesh che avevo fatto tempo addietro e che tengo pronta per simili necessità. L' ho modificata fino ad avere una linea generale che mi soddisfacesse abbastanza (fig. 6).

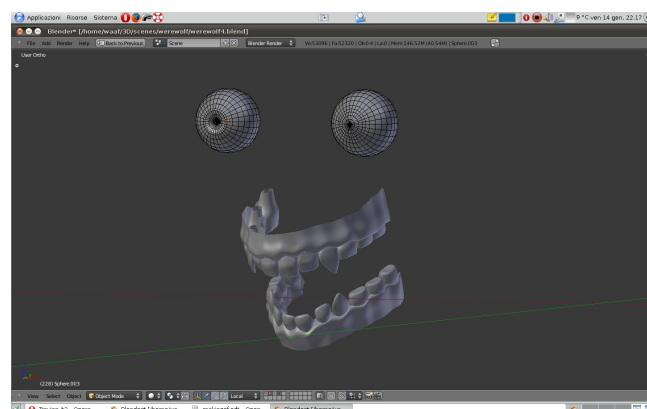


fig. 6

Si è passato quindi alla fase di retopo. Non c'è molto da dire su questa fase, se non che ho cercato più possibile di far seguire ai poligoni l'andamento delle fasce muscolari (fig. 7).

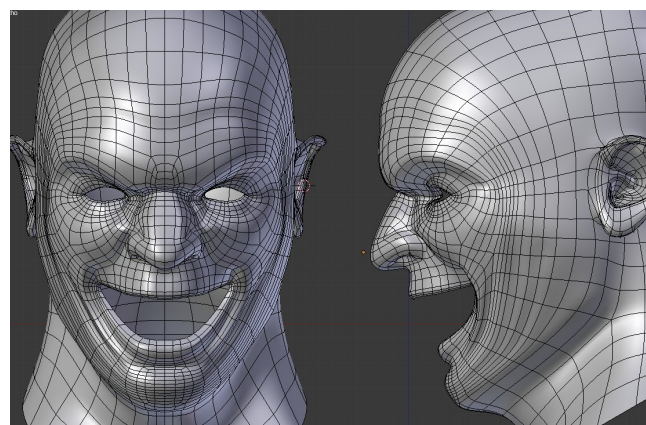


fig. 7

Per quello che riguarda le textures, ho scelto di utilizzare l'approccio classico.

Con la reference frontale aperta in GIMP ho modificato la grandezza della superficie a 6000x4000.

Ho quindi aperto come nuovo livello la reference laterale e la ho posizionata in modo che l'occhio visibile fosse sovrapposto all'occhio destro della front. Quindi con la gomma ho cancellato la parte dell'immagine side che andava a coprire la front. Ho copiato il livello e, con il mirror, l'ho specchiato ripetendo poi il processo con la parte sinistra del volto (fig.8).

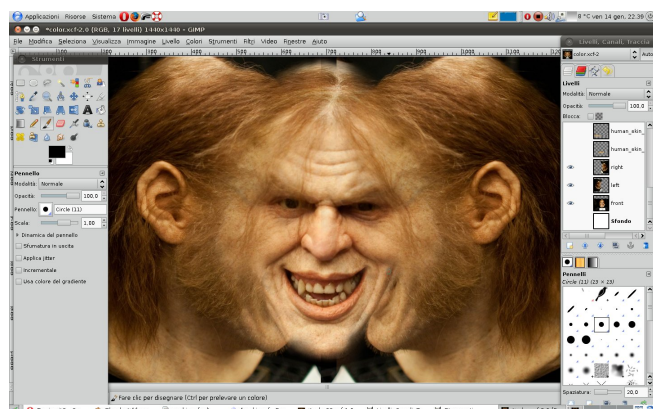


fig. 8

Il resto è stato un lavoro di cerotto e di clone. Ho anche aggiunto come livello una texture pelle generica, che andasse a coprire quelle zone dove era assente la pelle del tizio (fig. 9).



fig. 9

Alla fine ho usato questa texture solo per i sistemi hair, mentre per il canale diffuse ho tolto tutti i capelli sostituendoli con una texture pelle generica.

Nella creazione della texture color va posta attenzione nel rimuovere le zone speculari presenti nella fotografia (es. sulle labbra o sulla fronte), e tutti quei punti dove si hanno cambiamenti di colore troppo netti.

Terminato il processo di creazione della texture color, la ho importata nell' UV editor ed ho iniziato il processo di unwrapping.

Nulla di così complicato. Ho pinnato i vertici “chiave”, quelli che dovevano coincidere con la texture, ovvero occhi, bocca, narici ecc.

L' Unwrap ha fatto il resto (fig.10).

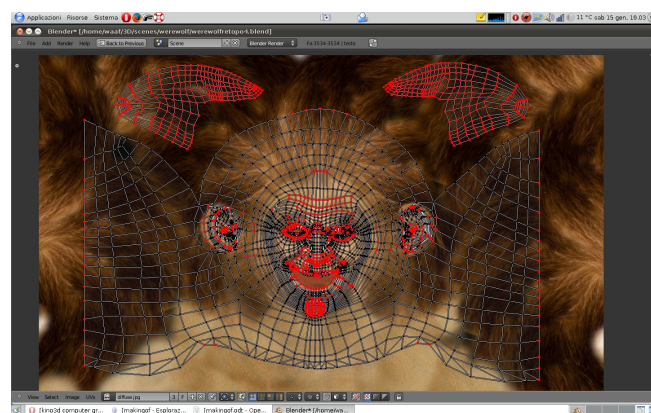


fig. 10

Una volta fatto questo ho bakerizzato la normal map (fig.11) e l'ambient occlusion map (fig.12). La mappa di spec è stata ottenuta lavorando di burn e di pennello in GIMP. Qui la principale accortezza deve essere quella di tenere a mente quali sono le parti che rifletteranno maggiormente la luce, e quali invece sono più opache.



fig. 11



fig. 12

E sono passato alla gestione dei materiali (fig.13).

Per la testa il materiale è relativamente semplice (fig.14). Il canale diffuse lo ho praticamente

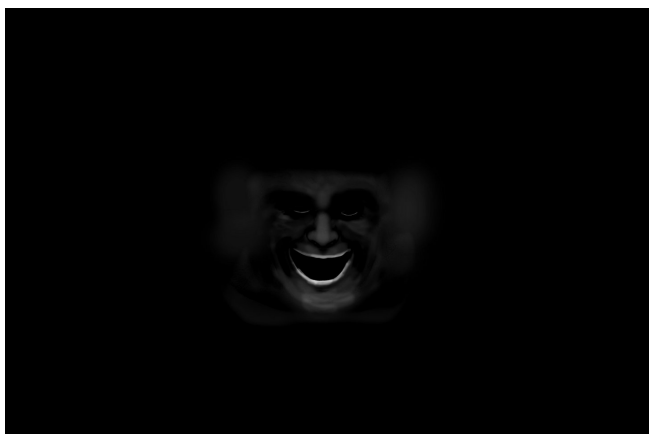


fig. 13

ignorato. Nel canale specular (CookTorr) ho settato a zero l'intensity per farla gestire dalla mappa di spec, ed ho tenuto un valore relativamente basso di Hardness. Lavoro di tentativi per arrivare alla giusta quantità di SSS.

Per quanto riguarda le textures il discorso si fa leggermente più complesso, ma non troppo.

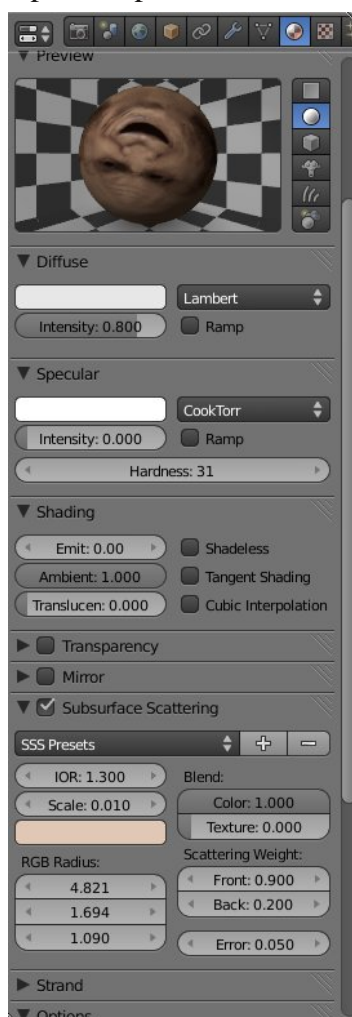


fig.14

La texture spec è stata applicata al canale con un valore intensity di 0,40.

Per ottenere un bump decente ho dovuto applicare sia la mappa di AO con un valore norm di 5 che la normal map con valore sempre di 5. Purtroppo con il SSS il bump quasi si annulla, quindi sono necessari valori molto alti.

Per la texture diffuse sono ricorso ai nodi (fig. 15).

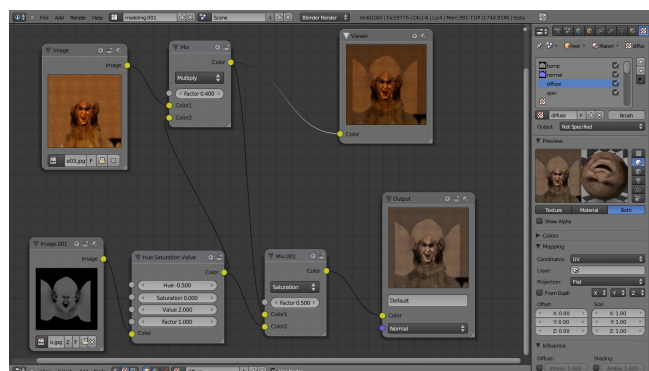


fig.15

Non sono un esperto nell'uso dei nodi, quindi probabilmente molti storceranno il naso vedendo come ho gestito la texture...

gli occhi si dividono in occhio interno ed occhio esterno. Per l'occhio interno (fig.16) ho usato una normale texture nel canale diffuse ed una procedurale cloud per il bump.

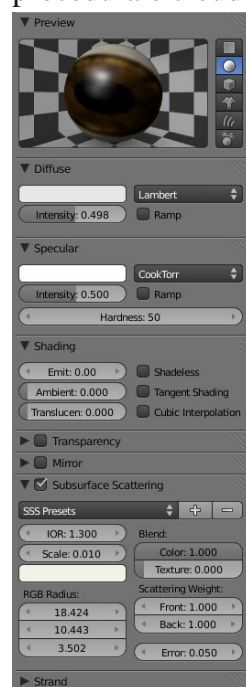


fig.16



fig.17

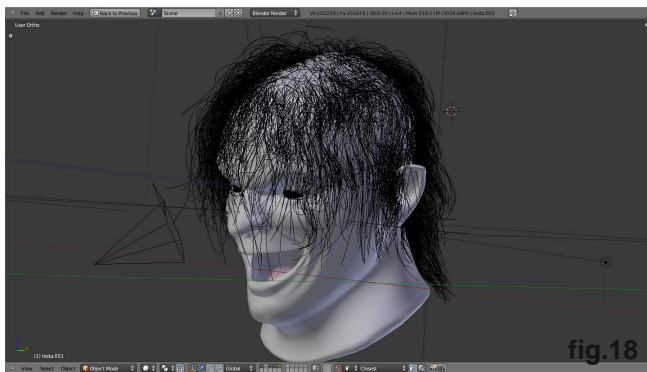


fig.18

L'occhio esterno invece ha solo una procedurale cloud nel canale norm. Il resto è lavoro di mirror e trasparenza (fig.17) si è giunti quindi ai sistemi particellari hair.



fig.19

Per i capelli ho usato 4 sistemi diversi, pettinati manualmente (fig.18).

Altri due sistemi per le basette (fig.19). Ad entrambi ho applicato un materiale (fig 20) con la diffuse map creata con ancora presenti i capelli, in modo da colorare le particelle realisticamente.

Un altro sistema hair è stato usato per la barba, un'altro per le sopracciglia ed infine uno per le ciglia. Questi tre, anch'essi pettinati a mano, hanno materiali molto generici.

Ho poi ripreso in mano denti e gengive, lavorando un po' di sculpt per renderli più "grotteschi" (fig.21), specialmente la parte superiore.

Prima di iniziare ad illuminare ho creato la secrezione lacrimare degli occhi (fig.22). Sembra un dettaglio inutile, ma il riflesso che da un materiale trasparente applicato a quelle mesh aumenta di molto la sensazione di "vita" del modello.

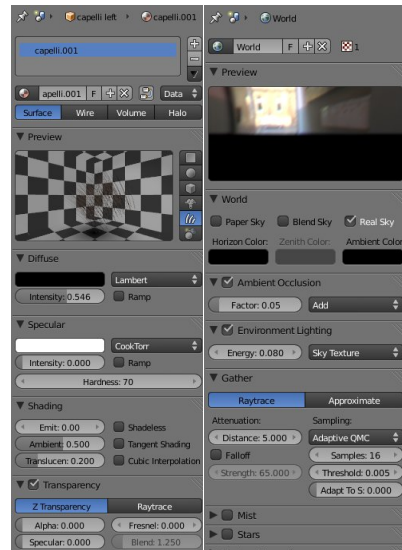


fig.20

fig.23

Parlando dell'illuminazione, iniziamo con una bella mappa HDRI nel "world".

L'illuminazione di questa l'ho tenuta parecchio bassa (fig.23), in modo da poter ottenere variazioni di chiaroscuro che rispondessero alle mie esigenze.

Le luci hanno subito molte variazioni, sia di tipologia, che di posizione ed intensità. Diciamo

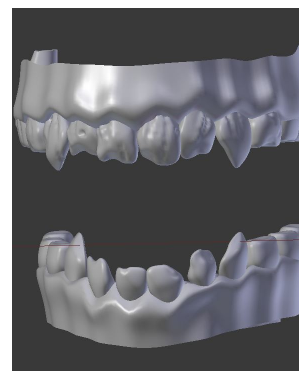


fig.21

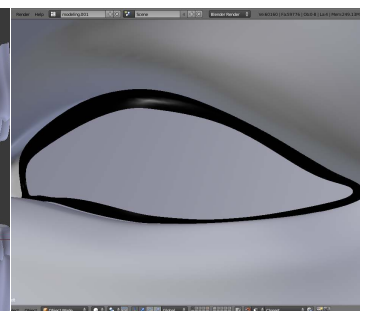
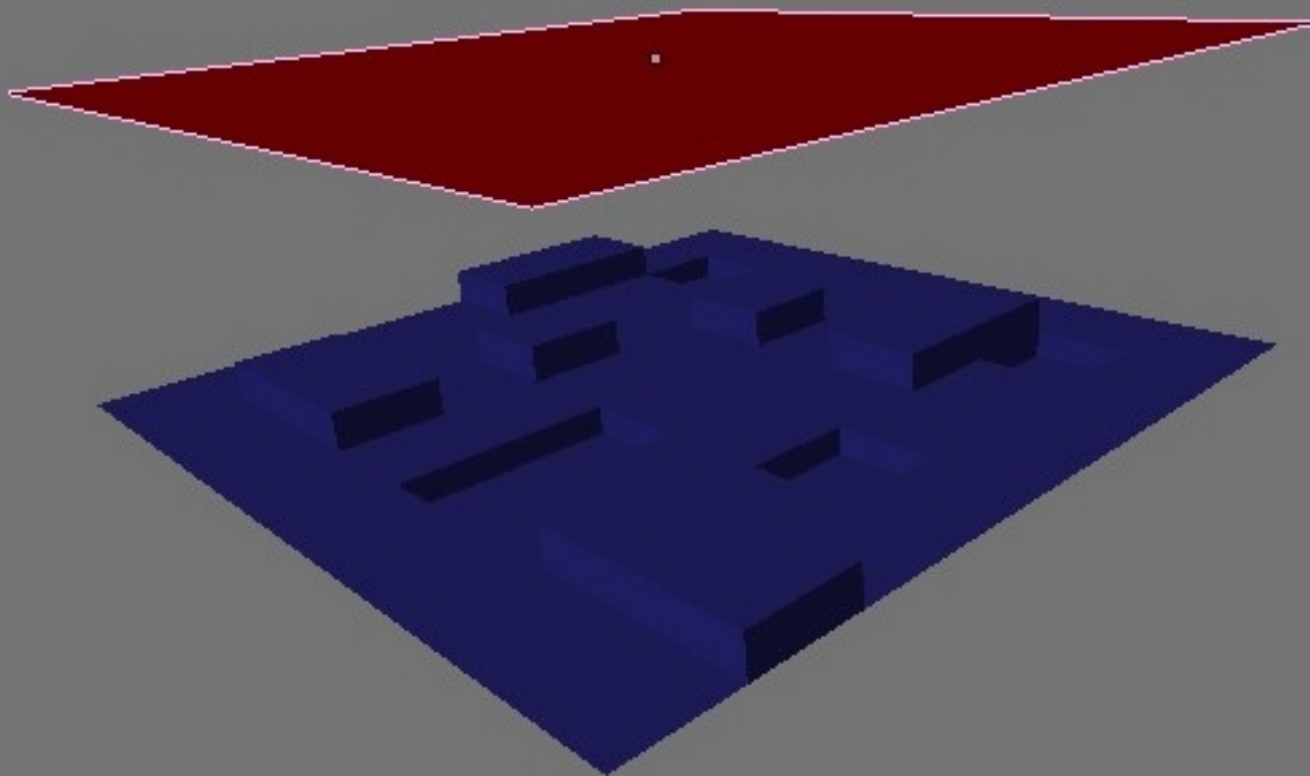


fig.22

che comunque ho mantenuto la classica regola delle tre luci, alla quale ne ho aggiunta una quarta per compensare la poca forza data dall' Ambient Occlusion. Quindi, una luce principale (che a volte era quella superiore, a volte quella sul lato destro) una backlight posteriore ed una fill sul lato sinistro.

Concludendo, mi ritengo abbastanza soddisfatto del lavoro, anche perché è il primo iniziato e terminato con la 2.5x, che sto ancora studiando e che ancora non maneggio alla perfezione. Spero di essere stato abbastanza chiaro. Saluti a tutti, teste di frullatore!





Bump e Normal Map

Facciamo chiarezza su questi due tipi di mappatura

di Lell

Questo articolo vuole fare chiarezza sulle differenze fra bump map e normal map, e spiegarne il funzionamento, i metodi di realizzazione e i campi di impiego.

Cominciamo dalle bump map.

Da wikipedia

“Il Bump Mapping è una tecnica di rendering dei materiali che aumenta la complessità degli oggetti realizzati senza effettivamente aumentare il numero di poligoni che compongono l'oggetto di partenza.

Alla texture che ricopre l'oggetto viene sovrapposta una seconda texture (spesso in bianco e nero) che il motore di rendering utilizza per simulare asperità, solchi, sporgenze e così via. Tali dettagli non fanno parte della geometria dell'oggetto ma vengono aggiunti solo in fase di rendering, spesso basandosi sui valori di luminanza della texture in bianco e nero usata per generare l'effetto.”

[fig1 “A sinistra la bump map di un muro di mattoni e a destra il suo effetto. Un passaggio più

netto dal nero al bianco avrebbe prodotto mattoni più squadrati”]

In pratica la bump map dice al motore di rendering che sulla superficie a cui è applicata ci sono dei rilievi e degli affossamenti. Per dirglielo usa un codice di colore. Di default in Blender il bianco corrisponde ad una zona bassa, mentre il nero ad una zona alta, cosa che rimane puramente relativa in quanto si può invertire l'influenza della mappa quando si vuole. Al contrario di altri

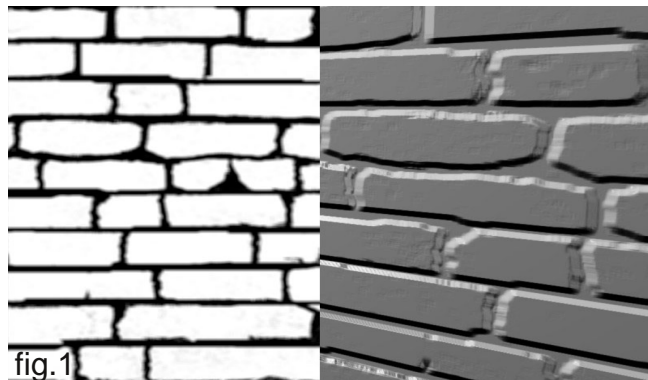


fig.1

programmi, in blender il colore bianco corrisponde ad una zona bassa, mentre il nero ad

una zona alta. In base a questi valori il motore di rendering genera sulla superficie delle ombreggiature e dei punti più luminosi per simulare la diversa inclinazione di quella zona, ingannando l'occhio umano. Tuttavia la bump map non modifica la reale geometria dell'oggetto, per cui, superfici che dovrebbero essere nascoste a causa dell'angolo di visuale, risultano ugualmente visibili, e questo fa capire all'occhio umano che in realtà la superficie è piana quando la visuale è troppo radente alla reale superficie e quando il bordo lineare della stessa si staglia sullo sfondo. Ciò non toglie che in molti casi in cui non si verificano queste situazioni limite le bump map non svolgano un ottimo lavoro di dettagliatura della mesh che altrimenti comporterebbe alti numeri di poligoni.

Le bump map possono essere anche usate per il displacement. La differenza sostanziale sta nel fatto che con il displacement si dice al programma di spostare realmente dei vertici durante il rendering, risolvendo i problemi testati citati per il bump mapping. Questo però ovviamente richiede che i vertici da spostare ci siano, aumentando la densità della mesh tanto più quanto più è dettagliata la mappa di displace.

Come ottenere una bump map

Per ottenere una bump map i metodi sono questi:

- 1 Modellare la mesh e ricavare la bump tramite i nodi di blender
- 2 Disegnare la bump map
- 3 Eseguire il bake del displace.
- 4 Usare la texture applicata al diffuse /color (quando è possibile)

1) In questo caso, dopo aver modellato la mesh (di solito una superficie lavorata come potrebbe essere un bassorilievo o un pavimento di pietra), si esegue un rendering con camera in modalità ortogonale, esattamente frontale alla mesh, e poi nel node compositor di Blender si ricava il canale Z dell'immagine e lo si trasforma in un'immagine di tonalità di grigi. In pratica Blender ti dice quanto distante sta ogni punto della mesh dalla telecamera.

[fig2 “Sequenza di nodi minima per estrapolare la bump map”]

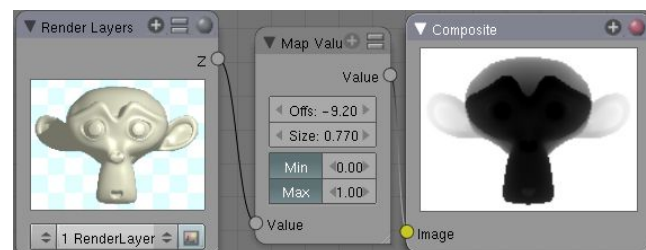


fig.2

2) la bump map può anche essere disegnata fin che si rimane sulle forme semplici. Ad esempio per un muro di mattoni potrebbe essere sufficiente una serie di rettangoli neri su fondo bianco.

Altrimenti si possono usare dei pennelli sfumati (ad esempio dal grigio al nero o dal bianco al grigio ecc.) [fig3 “Pennelli per realizzare bump map e loro relativi risultati”]

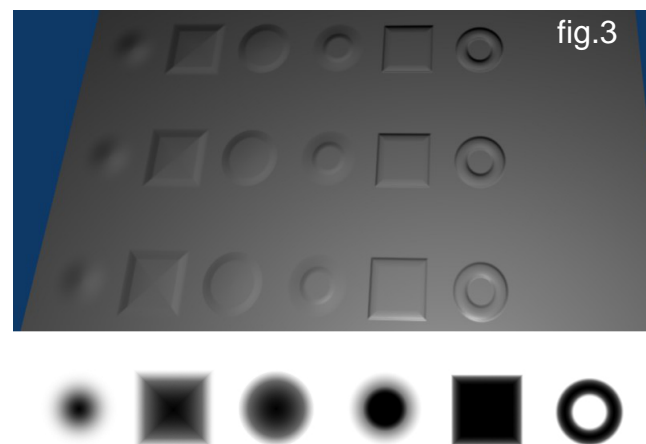


fig.3

Le combinazioni possono essere molte. L'importante è ricordare che le parti chiare sono quelle basse, mentre quelle scure sono alte. E ricordare che per ottenere una superficie inclinata ci vuole una sfumatura lineare da un colore all'altro.

3) Il processo per ottenere la bump map con il bake è simile a quello con i nodi. Bisogna modellare quello che si vuole riprodurre. Poi bisogna posizionare una nuova mesh piana e rettangolare sopra quella da mappare. La mesh piana deve essere unwrappata e nell'UV editor deve esserci caricata un'immagine nuova delle dimensioni che vorremo che la bump map abbia.

La posizione in Z del secondo piano è determinante per il tipo di risultato ottenuto.

A questo punto bisogna selezionare contemporaneamente le mesh da ricalcare e per ultima quella piana. Ora si può passare al bake del bump.

Nel pannello Render aprire la scheda Bake. Dal menù Bake mode selezionare Displacement, e segnare la casella Selected to active, che dice di proiettare la mappa dagli oggetti selezionati su quello selezionato per ultimo.

Ora possiamo premere il pulsante Bake. In questo modo Blender disegnerà sull'immagine che abbiamo caricato nell'UV editor per la mesh piana, la nostra bump map in base agli oggetti che sono sotto. Ora questa mappa può essere caricata come texture su una mesh più semplice per simulare i dettagli.

4) E' credenza comune che sia sufficiente convertire in tonalità di grigio una texture qualunque per ottenerne la relativa bump map. Purtroppo questo fatto non è matematicamente vero. La sua realizzabilità è tutta relativa. E' possibile farlo se abbiamo ad esempio un pavimento di pietre chiare con le fessure scure. Sarà sufficiente applicare questa texture al negativo per ottenere la corretta bump map.

Tuttavia, se vi dovessero essere delle pietre più

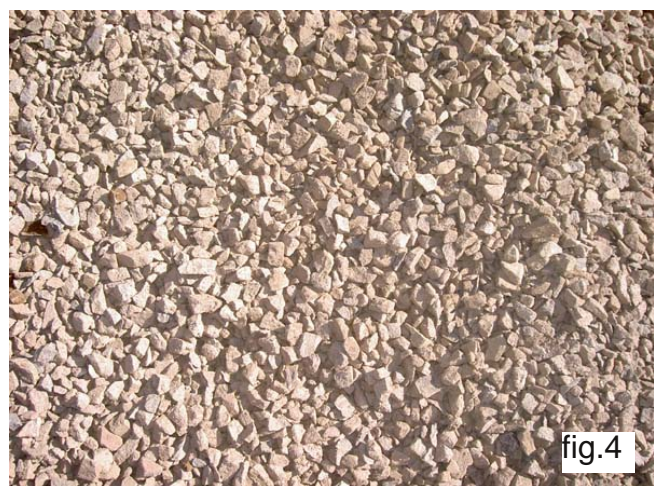


fig.4

scure o effetti di luci e ombre che falsino la luminosità degli elementi allora non sarà più possibile usare la texture come bump.

Prendiamo questa texture di esempio. [fig4] Se volessimo applicarla al canale Nor (che corrisponde anche al bump) ci ritroveremmo con

un risultato non corretto. Infatti le zone in ombra dei sassolini hanno una luminosità uniforme. Però non hanno una quota rispetto al terreno altrettanto uniforme. Lo stesso vale per le zone in luce. Infatti ogni faccia dei sassolini, è un piano inclinato, e per essere rappresentato con una bump map avrebbe bisogno di una sfumatura che va dal nero nella parte alta al bianco della parte bassa, indipendentemente dal fatto che sia in ombra o in luce.

[fig5 "La texture applicata al canale Nor del materiale non crea l'effetto sperato, ma solo un susseguirsi di piano alti e bassi. C.V.D."]

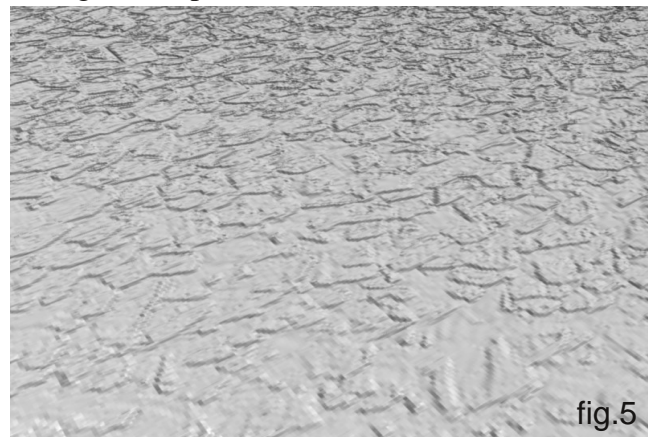


fig.5

Questo non significa che per lavori di tipo hobbistico o dove la perfezione assoluta non è richiesta, usare la texture del diffuse come bump map non sia una buona soluzione. Infatti a volte è sufficiente dare la sensazione di rugosità di una superficie per dare valore all'immagine.

Tuttavia esistono dei filtri di Gimp (tipo Low Frequency Even) o dei metodi particolari per eliminare questo tipo di imperfezioni. Potete trovarne un esempio in BMI9, nell'articolo di Seregost sulla creazioni di texture.

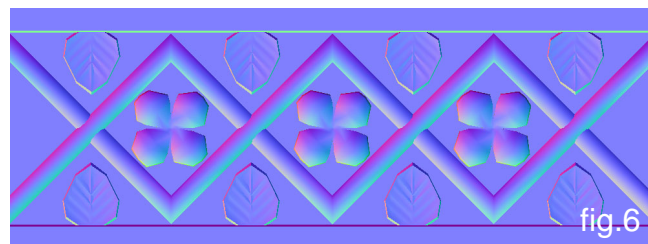


fig.6

[fig6: Normal map]

Passiamo alle normal map

Come le bump map, anche le normal map sono utilizzate per simulare dettagli che non sono

realmente presenti sul modello 3D, senza però modificarne realmente la geometria, nemmeno durante il rendering. Per fare questo la normal map non ragiona per altitudine rispetto alla faccia su cui è applicata come la bump map, ma ragiona sulla direzione delle normali di quella faccia. Mi spiego meglio: ogni faccia di un modello 3D (e anche ogni vertice, ma in questo caso non ci interessa) possiede una normale, che non è altro che una retta che dice al motore di rendering in che direzione è rivolta questa faccia. Questo serve al motore di rendering per decidere quanto deve essere illuminata tale faccia. Con la normal map applicata ad una faccia possiamo dire al motore di rendering che quella faccia è suddivisa in tante facce quanti sono i pixel della mappa, e che ognuna di queste facce è inclinata in un certo modo. Per dire al motore di rendering in che modo ognuna di queste facce è inclinata si usa un codice di colori basato sui tre colori fondamentali (RGB) dei sistemi digitali, ovvero rosso(Red), verde(Green) e Blu(Blue).

A questo punto, per capire come funziona il codice colore, è meglio spiegare praticamente come viene creata una normal map (nella maggior parte dei casi).

Per creare una normal map solitamente è necessario modellare la mesh che dovrà essere rappresentata dalla mappa (ad esempio le rughe e le imperfezioni di un volto o i dettagli più minuti di qualche parte meccanica). Per creare la normal map di questa mesh ad alta risoluzione bisogna seguire il punto 3) spiegato nella sezione delle bump map, con la sola differenza che al posto di selezionare Displacement come metodo di bake, bisogna selezionare Normal. Selezionando Normal compaiono delle possibili scelte sul sistema di assi da adottare. Di solito usando Tangent non si sbaglia e va bene per gli usi più comuni della mappa. Terminato il processo di baking della mappa ci troveremo con la classica immagine multicolore che assomiglierà alla rappresentazione della nostra mesh illuminata da luci colorate. Vediamo di capire come ha fatto Blender a decidere il colore di ogni pixel.

Blender proietta ogni pixel dell'immagine di partenza sul corrispondente punto della mesh,

andando ad incontrare una certa faccia.

Quando trova la faccia va a vedere in che modo questa è inclinata, ovvero di che angoli è inclinata la sua normale rispetto ad ogni asse. Ad ogni asse è assegnato uno dei tre colori fondamentali RGB: asse X -> rosso (R), asse Y -> verde (G), asse Z -> blu (B). A quel punto Blender genera un colore in base al contributo di ogni asse nell'inclinazione della normale di quella faccia. Detto praticamente, se una faccia è inclinata esclusivamente verso l'alto sarà solamente blu, se è inclinata a 45 gradi verso l'alto e verso destra, avrà un colore con uguali quantità di rosso e blu, e così via. In realtà non è proprio così, perché Blender deve tenere conto anche delle direzioni negative, per cui in realtà una faccia rivolta completamente verso Z avrà un colore RGB di valore 128,128,255.

Quando la mappa viene applicata alla faccia come normal map (per farlo bisogna anche attivare il tasto "Normal map" nel pannello texture di Blender) viene eseguito il procedimento inverso, ovvero blender legge le quantità di rosso, verde e blu di ogni pixel, e di conseguenza calcola l'inclinazione della normale in quel punto, e la sua conseguente illuminazione. Infatti l'illusione di tridimensionalità di una faccia mappata con normal map deriva soltanto da effetti di luci e ombre creati sulla faccia.

A causa di questo modo di ragionare della normal map si possono presentare dei casi critici in cui tali mappe non possono essere usate. E' il caso in cui vi siano delle facce perpendicolari al piano sul quale la mappa viene proiettata. Infatti, se una faccia ha la normale completamente rivolta verso Y, guardandola dall'alto avrà uno spessore pari a zero, per cui non potrà essere rappresentata da nessun pixel della mappa. Ne risulta che Blender noterà solamente le facce che siano almeno leggermente rivolte verso Z (riferito al sistema scelto per la mappatura). Se prendiamo il classico muro di mattoni belli squadriati e proviamo a farne il bake, ci troveremo con un'immagine uniformemente blu, perché le uniche facce visibili dal piano di baking saranno quelle rivolte verso di lui, ovvero le facce dei mattoni e il fondo delle fughe.

Di conseguenza, se proveremo ad applicare questa mappa ad una superficie piana, non noteremo nessun effetto di rilievo. Un trucco per risolvere la questione potrebbe essere di realizzare i mattoni con un leggero angolo di spoglia, in modo che le facce del contorno siano leggermente visibili, e quindi rappresentabili con la normal map.

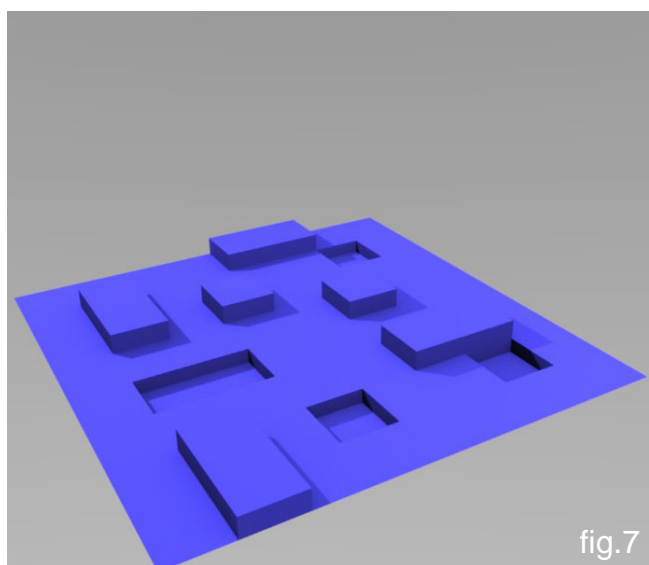


fig.7

[fig7 e 8 “Ho provato a ricavare la normal da questa mesh con tutti rilievi perpendicolari alla superficie, e quello che ho ottenuto è stato quel quadrato bluastro. C.V.D.”]

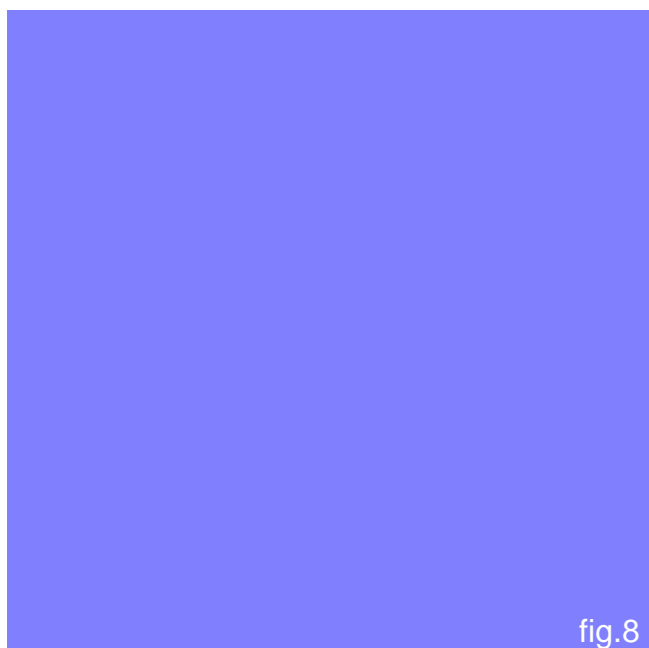


fig.8

[fig9 “Una decorazione modellata per ricavare una normal map per un videogioco. Notare le pareti del ribasso leggermente inclinate per non essere invisibili al bake della normal”]

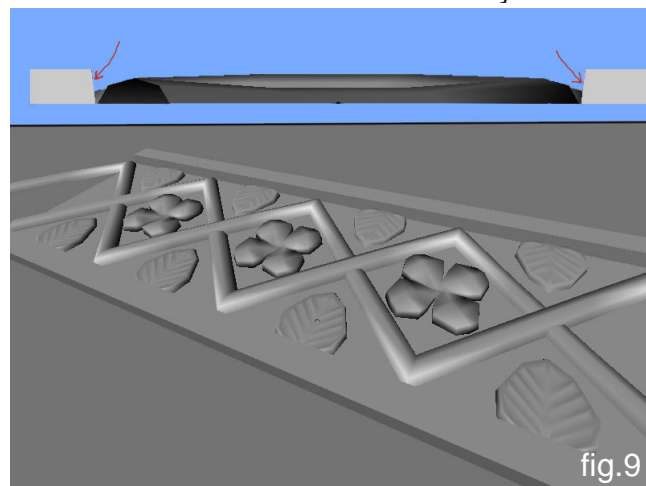


fig.9

[fig10 “La normal map ricavata dalla mesh di figura 9.”]

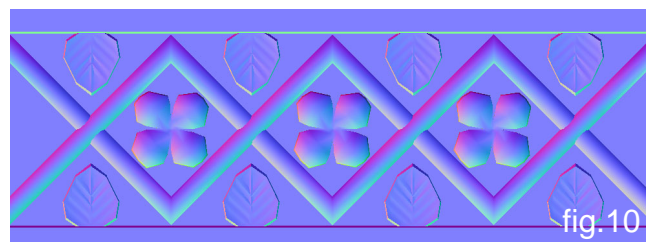


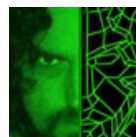
fig.10



Color Correction

Preview e ispirazione: set di nodi per esplorare le possibilità del compositor di Blender

di Fernando (Gillan) Luceri
www.luceri.it



Questo articolo è la versione ampliata aggiornata e tradotta di una serie di articoli pubblicati sul mio blog e nei quali venivano illustrati alcuni node-rig (gruppi di nodi) per il compositor di blender. Abituamente utilizzavo questi nodi per avere una veloce anteprima di alcuni preset per la correzione colore, confrontare fra loro diverse impostazioni e magari farmi ispirare dal risultato di regolazioni che probabilmente non avrei mai utilizzato.

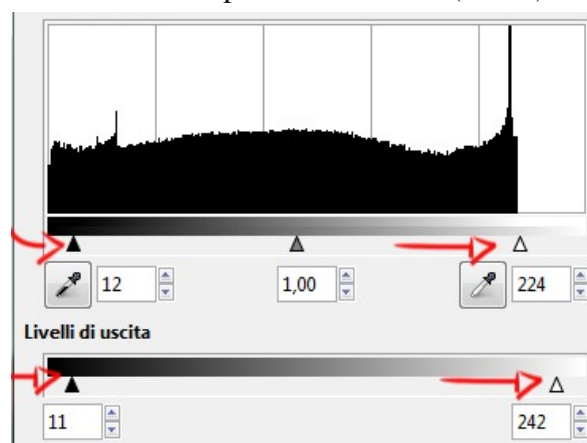
Questi settaggi sono liberamente scaricabili dal seguente link <http://www.luceri.it/2011/01/extra/>. Il primo set di cui parleremo riguarda il bilanciamento della luminosità.

Balance of Brightness

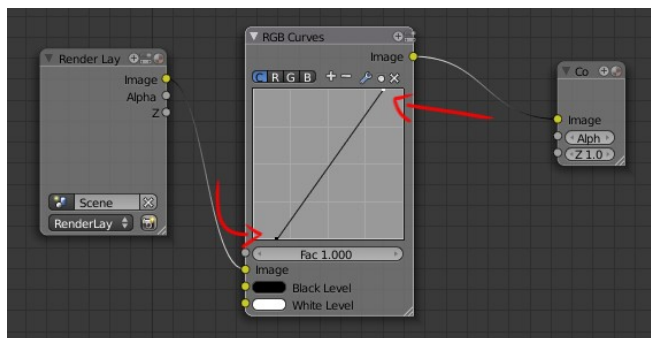
Nel compositor di blender esiste un nodo "Brightness and Contrast" che permette di scurire o schiarire una immagine in input. Spesso però alcune immagini ci soddisfano nelle zone luminose ma ci sembrano troppo scure nelle parti in ombra. Utilizzando il nodo "brightness and

Contrast" per schiarire l'immagine rovineremmo inevitabilmente la parte luminosa che invece andava bene così com'era. In questo caso converrà usare il nodo "Curve".

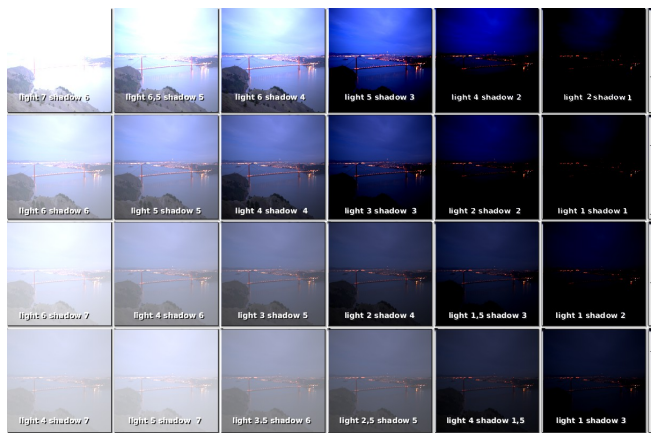
La prima cosa da fare è decidere la profondità del bianco e del nero. Dovremo in pratica decidere quanta parte ed in che misura la parte luminosa dell'immagine dovrà essere vicina al bianco e quanta ed in che misura quella scura sarà vicina al nero. In Photoshop o in Gimp questo può essere fatto con il pannello "livelli" (levels):



In Blender per ottenere un risultato simile bisogna spostare i punti di inizio e fine della curva del nodo "Curve"



Definiamo "1" come la posizione del punto di inizio della curva che dà il risultato più scuro (in basso a destra) e "7" la posizione che dà il risultato più luminoso (in alto a sinistra), stessa cosa per le posizioni del punto finale della curva. A questo punto avremo una serie di possibilità che vanno dalla combinazione più scura light1-shadow1 a quella più luminosa light7-shadow7 passando da combinazioni in cui le parti luminose vengono scurite e quelle in ombra schiarite o viceversa. Il node-rig di cui stiamo parlando realizza una griglia in cui queste possibilità vengono sistemate fianco a fianco per poterle confrontare e decidere la combinazione più appropriata. L'utilizzo è molto semplice: basta caricare nel nodo input l'immagine che si vuole testare, premere F12 ed otterremo una immagine come questa.



Bisogna tener presente che i settaggi più estremi danno dei risultati accettabili solo se in input vi è una immagine hdr o i dati raw di un rendering.

Oltre ai nodi "Brightness and Contrast" e "Curve" esistono altri metodi per aumentare o diminuire la luminosità, il contrasto o la saturazione. Di questi metodi alternativi si occupa il prossimo set.

Alternative

Non sempre i metodi classici danno il risultato sperato, così è interessante valutare anche alcuni metodi meno ovvi. Eseguite un rendering dopo aver caricato la vostra immagine nel nodo input del set che avete scaricato, otterrete qualcosa di simile:



la schermata è divisa in tre aree, in alto il riquadro contenente: sopra la riga con i settaggi per aumentare la luminosità e sotto quelli per diminuirla. Oltre ai classici "Brightness and Contrast" e "Curve" troviamo un set che varia la luminosità solo della componente "value" senza alterare colore e saturazione, poi un nodo che somma i valori dell'immagine a se stessa ed uno che invece li sottrae, ancora un nodo che scherma l'immagine con se stessa opposto ad uno che invece la moltiplica. Infine le opzioni di blending dodge e Burn. Più in basso a sinistra troviamo i metodi per aumentare o diminuire la saturazione. In ordine: il classico nodo "Hue and saturation", Il "Bw mix" che desatura l'immagine miscelandola con se stessa però in bianco e nero, l "All Component" che sovrappone i vari canali di colori come se fossero in toni di grigio ed il "washed" che desatura i colori saturi a favore di quelli poco saturi. Per saturare l'immagine invece "compress" somma i valori di luminosità a quelli di saturazione, "all channel" che moltiplica i valori dei canali rgb, e "Oversaturated" che pone

ogni colore con saturazione al 100%.

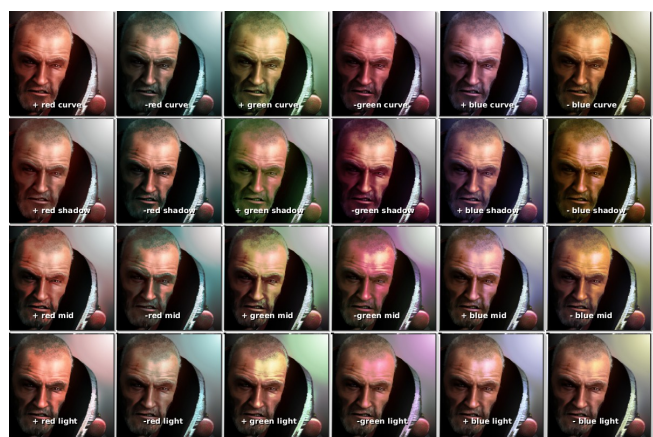
Per il contrasto oltre al classico nodo "brightness and Contrast" troviamo il nodo "curve", il nodo "only value" che non agisce su colori e saturazione ed "extract" che aumenta la differenza fra le parti di una immagine.

E' importante notare che tutti i nodi di cui parliamo in questo articolo sono "resolution independent" ecco perchè non vengono usati filtri tipo "blur" o "sharpen" che invece hanno un effetto molto influenzato dalla grandezza dell'immagine.

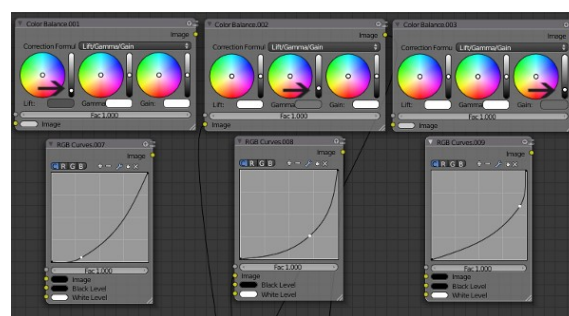
Il prossimo nodo si occupa di bilanciare i colori fra le zone scure e quelle chiare.

Shadow-mid-light.

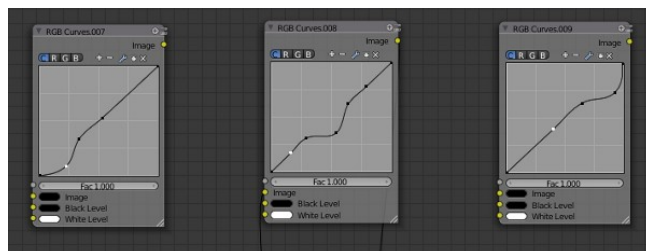
Se è difficile capire se in una immagine c'è troppo rosso (o troppo poco) o troppo verde, è ancora più difficile capire se c'è troppo rosso (o verde o blu) solo nelle aree scure di una immagine, o solo in quelle chiare o in quelle mediamente illuminate. A volte solo provando si riesce a percepire il bisogno di aumentare o diminuire un colore particolare nelle ombre (o nella luce o in mezzo). Per questo motivo, il node-rig di cui stiamo parlando, è particolarmente conveniente, perché permette di visualizzare e confrontare in una sola schermata molte opzioni.



In questo set ho usato dei semplici node "Curve", in realtà per fare questo tipo di regolazioni è molto più efficace il nodo "Color Balance" che agisce in modo molto sfumato e naturale un po come se fosse una curva morbida del nodo "curve".



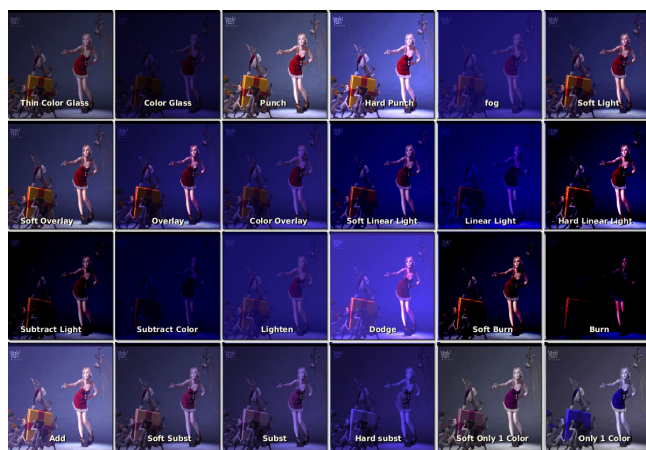
Il motivo per cui non ho usato il color balance è che nel preview mi serve capire immediatamente ed in modo inequivocabile in quale range di luminosità sta agendo il filtro.



Come abbiamo visto, questo set permette di variare il colore di una immagine per correggerne eventuali mancanze ma non è adatto se vogliamo dei cambiamenti radicali nella mappatura del colore. Per questo c'è il prossimo set:

Color Mapping

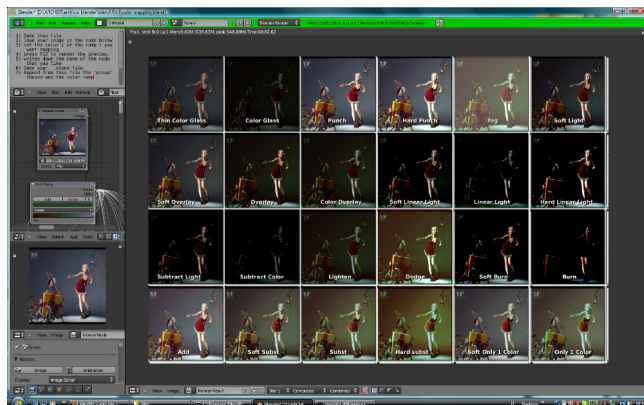
Il concetto è molto semplice, prendere un colore e miscelarlo in varie modalità con l'immagine originale. Caricate nel nodo "input image" la vostra immagine, successivamente decidete il colore da mappare cliccando sul nodo "ramp shader". In questo caso, essendoci un solo colore, il ramp non restituirà una sfumatura ma funzionerà come un semplice nodo "rgbcOLOR". premete F12 e guardate il risultato.



Come potete vedere, i risultati sono molto interessanti. A me piacciono molto "hard punch", e "subst", e spesso "subtract the light". Alcuni effetti non sembrano dare dei risultati interessanti ma bisogna ricordarsi che molto dipende dal colore di input. Alcuni nodi che in questo caso sembrano inutili avrebbero dato un risultato completamente diverso se il colore da mappare fosse stato più chiaro o meno saturo.



quello che rende questo set veramente divertente è la possibilità di usare un ramp (una sfumatura fra più colori) con risultati davvero sorprendenti.



dulcis in fundus...

Color Grading

"I termini 'color correction' e 'color grading' sono spesso usati come sinonimi, anche se 'color grading' è usato soprattutto per indicare le regolazioni del colore fatte per scopi creativi e non per correggere i difetti del colore" Adobe.

Come accennavo prima, in questi set ho evitato tutti i filtri dipendenti dalla risoluzione (quelli per capirci in cui si decide per quanti pixel si propaga l'effetto) quelli non legati alla natura del colore (tipo flare, noise etc.) e non ho usato

maschere e sfumature. Per il color grading, tuttavia, l'uso di questi nodi è fondamentale per ottenere un buon risultato. Pertanto, spiegando il nodo, cercherò di indicare quale sia la loro utilizzazione finale facendo un esempio di quali filtri maschere ed effetti è possibile aggiungere. Come al solito, caricate la vostra immagine nel nodo input e premete F12.



Prima di tutto gli ospiti. In questa raccolta non ho potuto fare a meno di includere alcuni preset di Francois Tarlier, (Grazie per l'ottimo lavoro sul nodo Color Balance).



F.Tarlier Bleach

La versione di F. Tarlier del Bleach Bypass. Come vedremo con Il "Bleach" si possono ottenere molti effetti, ma mi piaceva l'atmosfera e la qualità di questo.

F. Tarlier Cool and Warm



Tutti i film di azione oggi hanno questi colori (transformers, Iron Man, ecc), funziona bene (come in questo caso) in combinazione con filtri

glow e glare e edge enhanced.

F. Tarlier Punchy



Un filtro che esalta i colori e il contrasto (attenzione è facile *bruciare* le zone luminose)

F. Tarlier Day to nit (day to Night)



Il filtro di Tarlier fa bene il suo lavoro. Ma un convincente effetto "da giorno a notte" richiede una buona mascheratura. In questo caso ho abbassato la luminosità del cielo e delle cose lontane prima di applicare il filtro poi ho aumentato il giallo nelle zone luminose. Infine, un po' di **glow** sulle cose più brillanti e una leggera desaturazione delle zone più buie.

Technicolor3strip



Direttamente dal DVD "Elephant Dream", technicolor3strip simula una pellicola Technicolor per dare l'aspetto molto colorato dei film degli anni dal 1935 al 1955 (nuove produzioni The Aviator, hercules ecc.)

Bleach



"Bleach Bypass, Skip Bleach, Bleach Reduction, ENR, ACE e Silver Tint sono tutti sostanzialmente lo stesso processo che dà come risultato un caratteristico elevato contrasto, con colori duri e sbiaditi. In tutti i casi una immagine in bianco e nero viene sovrapposta all'immagine a colori. La tecnica è stata utilizzata per un lungo periodo di tempo, ed è continuamente in voga. Buoni esempi comprendono Seven, Evita, 1984 e Salvate il soldato Ryan, ma naturalmente ci sono numerosi altri esempi che imitano l'effetto usando il miglioramento del colore in post-produzione."
© Kevin Shaw 2005.

Ho cercato di riprodurre in blender alcune tecniche tipiche della analogico "Bleach bypass", ho sovrapposto (in quattro modi differenti) su una immagine a colori, la stessa immagine in bianco e nero.

Cold Enhanced – Coldest



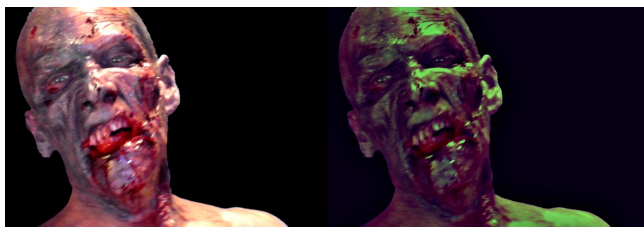
"Coldest", ricrea le luci e le ombre blu che imperversavano nei film di fantascienza degli anni '80, Ovviamente, funziona meglio con un sacco di **Glow** e **Glare**. "Cold Enhanced" desatura i colori caldi e esalta quelli freddi.

Warm Enhanced – Warmest



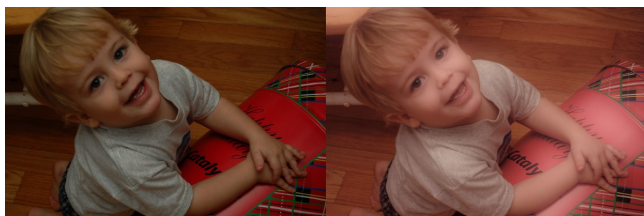
Al contrario, "Warm enhanced" desatura i colori freddi ed esalta le tonalità. "Warmest" dà fuoco alla tua immagine.

Terror



Filtro che trasforma una foto tranquilla in un incubo, è meglio se si utilizza un **control brightness** e un **selective contrast**.

Romantic



Con questo filtro, dal nome esplicativo, è necessario utilizzare un **glow** morbido e chiaro

Medieval



Questa atmosfera è tipica dei film storici. In questo caso ho sovrapposto un'immagine con crepe e fessure

Frozen



Molto efficace per congelare le immagini, soprattutto se utilizzato con un filtro **Sharpen**.

Matrix



Un semplice grading per ricreare l'atmosfera del mondo virtuale di Matrix, ora manca solo il Bullet-time.

Punch Color



Un filtro che massimizza la saturazione del colore. I bianchi puri ed i neri vengono sostituiti con del colore molto chiaro o molto scuro .

Clean



Un filtro ampiamente usato nel mondo della moda e del fashion. Brucia le parti chiare mentre desatura quelle scure, è usato spesso con molto glare. Dopo questo filtro il **Color Balance** funziona molto bene.

Old Photo



Questo filtro riduce notevolmente la gamma e scurisce i bianchi. In questo caso ho usato un ombra e un nodo **displacement** per simulare le incisioni all'acquaforte.

Polaroid



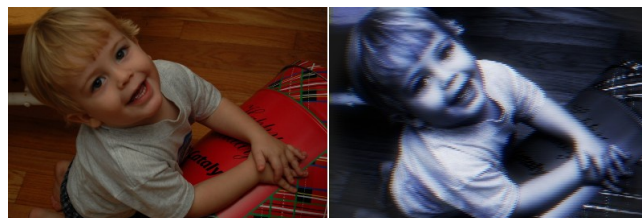
Un modo alternativo di usare il nodo **tonemapping** (utilizzato normalmente per selezionare l'intervallo della HDR) per riprodurre le vecchie foto degli anni '70. Ho aggiunto una leggero **blur** e ho sovrapposto una immagine di carta stropicciata aggiungendo infine il classico bordo bianco.

Seppia



Ecco il classico effetto seppia, qui arricchito da un **vignetting** screpolato e da un effetto lavato ottenuto con il nodo erode.

Old tv



Al grading ottenuto con il filtro ho aggiunto un **lens distortion** con dispersione e un lieve **glow**. Poi ho creato un effetto interlacciato ottenuto sovrapponendo la stessa immagine leggermente spostata attraverso una maschera di strisce sottili.

Conclusioni

Spero che questo articolo possa esserti utile, credo che possa farlo in diversi modi. Il più ovvio è quello, naturalmente, di scaricare i file dei set e di utilizzare l'anteprima per vedere come il tuo rendering potrebbe apparire, ma può essere interessante anche curiosare all'interno dei gruppi per carpire i loro segreti e capire di più di come funzionano i nodi nel compositor di Blender. In ogni caso spero che ti sia piaciuto leggerlo almeno quanto io mi sono divertito a scriverlo. Buon divertimento.

Ultima Ora!

Per utilizzare questi preset nelle proprie creazioni bisognava annotarsi il nome e poi "appenderlo" nel proprio file. Per fortuna, adesso, tutto questo non è più necessario grazie ad un add-on entusiasmante. Fabio Russo ha programmato uno script davvero utilissimo, Basta scaricarlo da questo indirizzo http://wiki.blender.org/index.php/Extensions:2.5/Py/Scripts/Compositing/Compositing_Presets copiarlo nella directory add-on di blender e poi attivarlo dalla finestra User Preferences. Una volta nel compositor sarà disponibile un menù attraverso il quale sarà possibile scegliere di utilizzare molti dei preset che abbiamo visto in questo articolo.



Un mare da favola

Seconda parte di un viaggio in fondo al mare

di Riccardo (jazzroy) Covino

Prendendo idealmente il testimone da Deskplus, che ha scritto l'articolo pubblicato precedentemente, proverò a fare un'analisi complessiva delle difficoltà tecniche e delle soluzioni sperimentate nella produzione digitale del progetto Un Mare da Favola.

Dover fondere assieme del materiale girato subacqueo con personaggi animati in 3D era una situazione un pò nuova per tutti e perciò ci siamo trovati spesso a complicarci la vita da soli, esplorando possibilità più o meno realizzabili su problemi che esistevano solo nella teoria!

Un esempio per tutti è quello relativo al camera tracking.

CAMERA TRACKING

Uno dei maggiori timori iniziali era proprio quello di non riuscire ad ottenere un buon camera tracking, il che a buon senso risultava

indispensabile vista la grande quantità di scene in cui i personaggi animati appaiono.

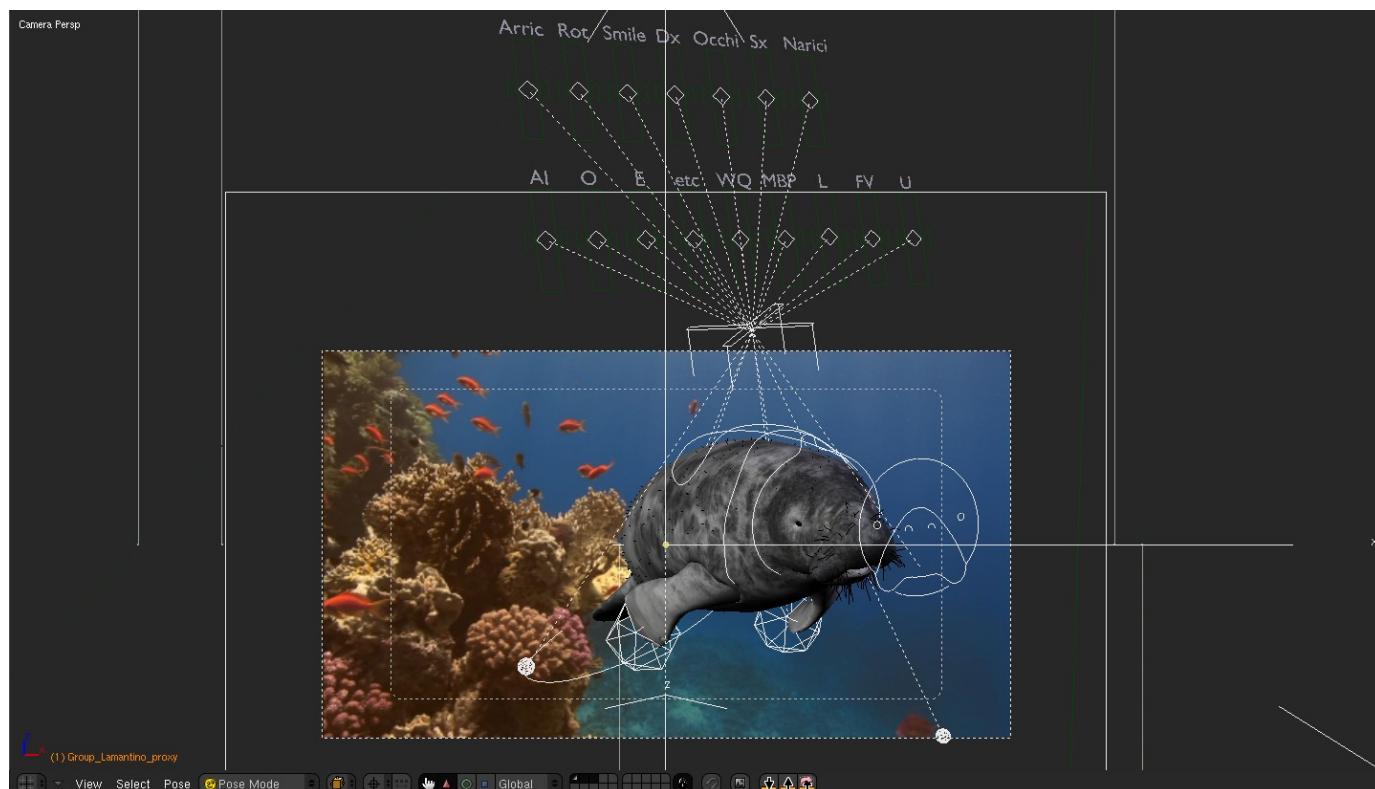
Come fare ad avere un buon tracking? Sicuramente il mare non presenta linee rette e superfici regolari che aiutino a calcolare la prospettiva.

Anche il posizionamento di marker non è facile: si possono avere oggetti fermi solo se posati sul fondo, senza contare che a seconda della distanza virano sul blu e cambiano di luminosità.

A causa di questi problemi e del fatto che la verifica della possibilità di effettuare il tracking sarebbe stata fatta solo dopo il rientro in Italia le prime riprese per i personaggi animati sono state fatte con camera fissa.

Il che però rappresentava un forte limite: un film in cui ogni volta che entra un personaggio simulato la telecamera è ferma non può certo definirsi all'avanguardia, anzi..

Oltretutto l'intenzione fin dall'inizio è stata di



creare personaggi realistici e scientificamente corretti, perciò anche il tipo di inquadrature non sarebbe dovuto cambiare rispetto al resto del girato.

E così, confrontandosi una sera, arriva l'evidenza della realtà: i pesci nuotano!

Già i pesci nuotano, mica camminano.. mica hanno i piedi "incollati" al pavimento..

E così si è passati da una inutile preoccupazione ad un eccesso di libertà, azzardando le riprese più ardite e movimentate, fiduciosi che tanto un pò di "galleggiamento" del personaggio nella scena non sarebbe affatto apparso come problematico.

Non era facile per Leo immaginarsi gli ingombri dei personaggi ed i loro movimenti, ed è stato bravissimo a fare le riprese nell'ottica di cosa dopo si sarebbe inserito.

E non è stato facile neanche dopo gestire questa "allegria" con cui ci siamo liberati del tracking, perchè se è vero che non c'è bisogno di essere incollati al terreno è altrettanto vero che tutte le vibrazioni ed imperfezioni del movimento della camera vanno poi ritrovate nel percorso dei personaggi.

Così Deskplus ha trovato un'occupazione notturna ai suoi pc, intenti a calcolare decine di

tentativi con Voodoo per far rendere credibile la sovrapposizione.. ed in certe scene è stato necessario fare il tracking manuale frame a frame.

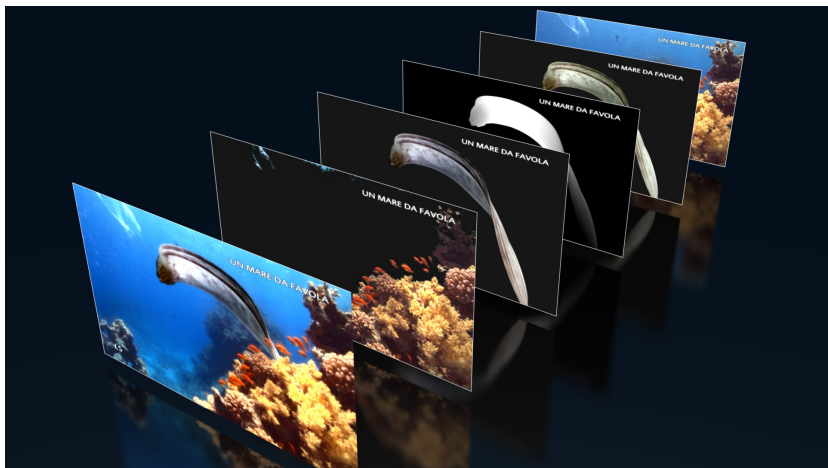
LIP-SYNC

Avete mai visto un pesce parlare? Copiare la nuotata è stato decisamente più semplice avendo delle reference video, ma decidere come avrebbe dovuto muovere la bocca è stato frutto di decine di tentativi e confronti tra i membri del team. Inizialmente l'errore più comune è stato quello di voler replicare ogni singola lettera pronunciata ma risultava orrendo il risultato finale, quasi alla "Paperissima".

Escludendo parecchie lettere e riconducendole a meno di dieci fonemi il risultato è stato decisamente più credibile. Da notare che i movimenti del corpo e della testa non sono stati volutamente "esagerati" per scelta del regista che non ha voluto storpiare troppo i personaggi.

L'altra grossa difficoltà è subentrata quando aprendo le bocche (delle più svariate morfologie!) non sempre era presente una lingua o dei denti!

La tartaruga ad esempio ha un becco rigido e non può pronunciare la "o" o la "u", quindi abbiamo mediato dando un po' di enfasi sul movimento della testa.



CHROMA KEY

Ci serviva un fondale blu.

Non è stato difficile trovarlo..

Il mare è un immenso fondale per chroma key, ma.. ci sono dei ma.

Il primo è che non sempre la ripresa guarda verso il profondo blu, spesso erano in mezzo ai coralli, oppure con la luce che fa capolino tra le increspature, e poi.. il posto è abitato, specialmente nel Mar Rosso!

Talvolta sembrava di essere nel traffico delle 6, tra branchi di pesci, sub della domenica e scafi in lontananza.

Il Chroma è stato provato in tutti i modi e in diversi software, e l'esito è sicuramente da raccontare, ma per farlo occorre una precisazione.

L'acquisizione del girato avviene tramite il software di acquisizione e montaggio Edius, che lavora egregiamente in simbiosi con la scheda di acquisizione e le videocamere ad alta definizione.

I file così ricavati sono avi non compressi sui quali il software riesce a sfruttare l'accelerazione della scheda, permettendo così di manipolare agevolmente diverse tracce una sull'altra. Così le sperimentazioni sul chroma sono iniziate direttamente in Edius, che ha degli strumenti semplici ma molto

efficaci per bucare le immagini.

Quando abbiamo affrontato il problema stavamo lavorando sulla scena della murena che esce da dietro al corallo, perciò avevamo bisogno di avere il fondale sdoppiato, con elementi davanti al persoanggio ed elementi dietro.

Tutto sembrava funzionare abbastanza bene sperimentando in PAL, ma quando si passa in HD si nota immediatamente il classico fastidioso alone nero intorno ai personaggi alphiati.

Il problema è chiaro e antipatico: Edius interpreta non correttamente la componente colore dell'alpha, e dà sempre lo stesso risultato indipendentemente dal fatto che in

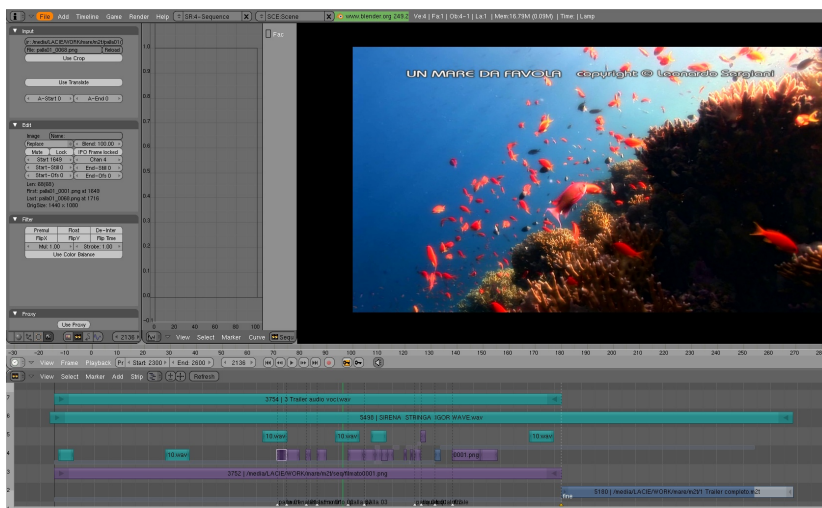
Blender si usi Pre-multiplied o Key.

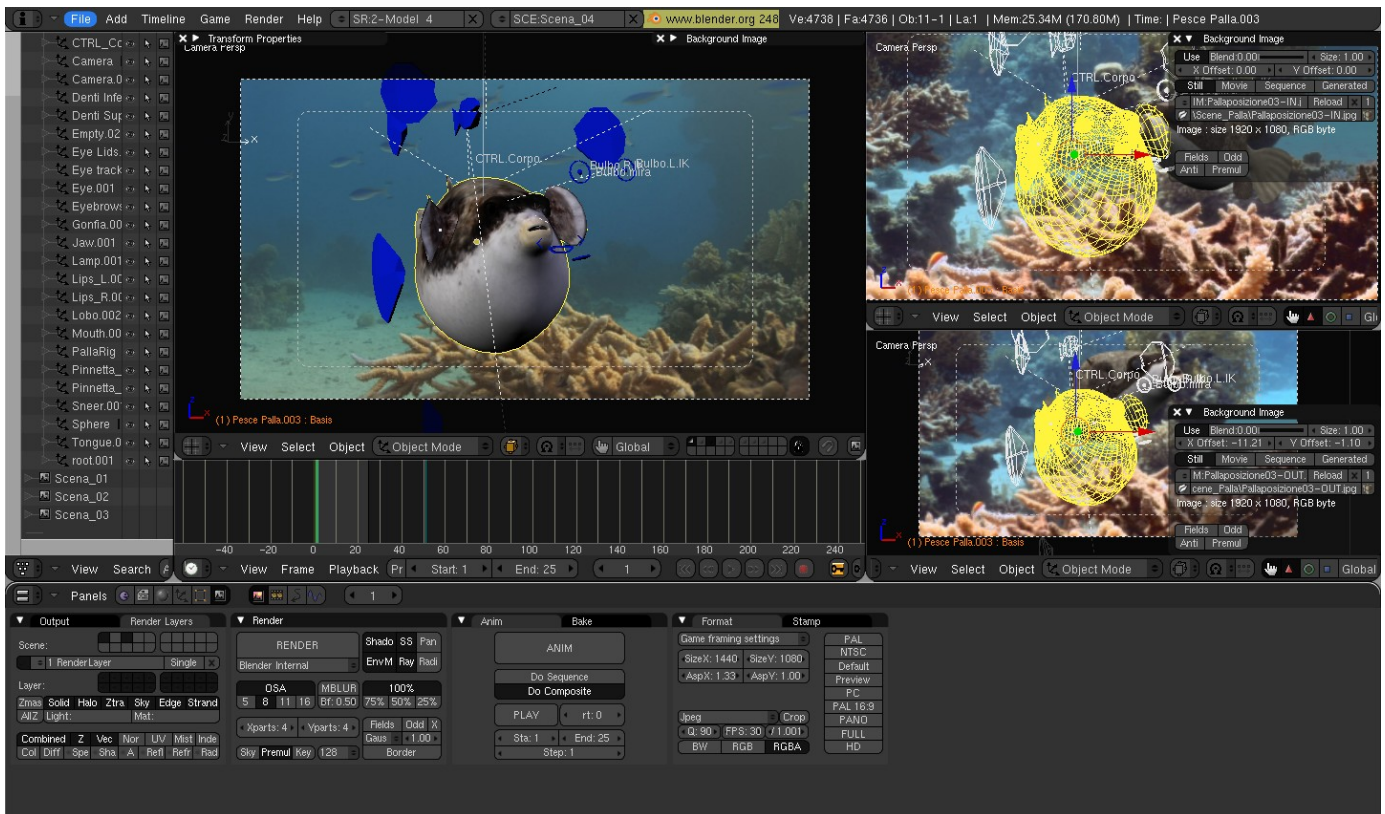
Questo significa che l'integrazione tra fondale e personaggi animati va fatta al di fuori del software di montaggio.

Si fanno diversi tentativi con After Effects, ma il tutto risulta un pò macchinoso: da Edius bisogna esportare per After Effects il filmato, da Blender le sequenze alphi, si composita in AE, si riesporta e si rimette il risultato finale in Edius.

Dovendo fare spesso correzioni sulla luce, sul colore, sui tempi e sul movimento il processo rallenta di molto le operazioni.

A questo si aggiunge il fatto che non eravamo pienamente soddisfatti di come buca, nonostante avessimo usato un approccio a più livelli per poter avere una gamma più ampia di





tonalità incluse.

Le difficoltà ci hanno così spinto a tentare la strada del compositing in Blender, e dopo diversi tentativi e innumerevoli ricerche su internet di set di nodi efficaci alla fine siamo giunti ad un "mostruoso" potentissimo flusso nel compositor che selezionava e buca con un grado di controllo e di morbidezza inaspettato.

Per noi Blender ha battuto Edius e After Effects senza il minimo dubbio!

IL MATCHING PERFETTO TRA HDV e RENDER

Le riprese in mare con videocamere HDV hanno rappresentato una serie veramente lunga di problemi dal punto di vista della qualità e della verosimiglianza dell'immagine.

Il primo, più evidente di tutti, è che i colori in mare cambiano con la distanza.

Le lenti con filtri permettono riprese con colori ottimali solo ad una determinata distanza, ciò che è troppo vicino diventa rosso per via del filtro, ciò che è troppo lontano tende a diventare solamente variazioni di blu.

Per ovviare a questo limite in Blender si è sempre aggiunto l'output dello z-buffer, questo poi viene usato come maschera per regolare l'intensità di effetti di colorazione in postproduzione.

Il risultato è un render a colori pieni, successivamente compositato e colorato in base alla luminosità delle immagini generate dallo z, questo ci ha permesso una regolazione precisa della tonalità di blu da ottenere in lontananza senza dover ripetere il render.

Un altro grosso problema è la compressione del formato HDV, che è in 4-2-0 e se viene analizzato per singoli canali colore è veramente uno scempio del dettaglio.

L'unica strada è stata degradare il render sino al livello dell'HDV, per fare questo abbiamo separato i canali RGB, lavorando su ognuno per portarlo alla qualità del formato, per poi riaccorparli insieme.

Infine il mare è pieno di corpuscoli, elementi in sospensione che tolgono limpidezza e danno la sensazione del movimento. Questi sono stati riprodotti con render particellari in Blender, sovrapposti poi alla scena intera per uniformarne la presenza.

RENDERING

L'obiettivo del realismo ci ha spinto subito a non usare il motore interno, perchè la mancanza di microdisplacement e di Global Illumination ci sembravano ostacoli determinanti.

Così abbiamo tentato un pò di alternative, nell'ordine: Yafray, Aqsis, Pixie, Gelato, Luxrender.

Il tentativo con Yafray non è durato molto, soprattutto a causa del fatto che nessuno di noi lo sapeva usare e i primi risultati non sono stati incoraggianti.

La catena dei motori Renderman invece è stata ovvia, partendo dal più conosciuto, al più completo, finendo sul più veloce.

Aqsis non dispone di Ray Tracing e i tempi di render in HD si sono rivelati biblici, Pixie sicuramente renderizza più velocemente, ma con problemi di corretta interpretazione del displacement, inoltre comunque i tempi si sono dimostrati lunghi con l'aggiunta del motion blur, andando nell'ordine di ore per fotogramma.

Gelato infine, grazie all'exporter di Marioamb, è stato il motore Renderman che per un pò di giorni ci ha fatto sognare, avendo tempi di rendering interessanti e un controllo relativamente semplice.

Luxrender è stato un tentativo tardivo, ma alla fine ciò che veramente ha dato la svolta è stato un giorno di prove con il motore interno.

Qualche luce leggermente colorata e accuratamente posizionata, un alto livello di subdivisions per un pò di displacement, un motionblur in compositing che dura pochi secondi, e le scene rendevano esattamente come avevamo sempre voluto.

La luce sott'acqua si comporta in maniera molto strana, scomponendosi, rimbalzando, arrivando da direzioni inspiegabili: l'osservazione dei coralli nel fermo immagine ha mostrato chiare incoerenze, con ramificazioni illuminate da destra e pochi centimetri a fianco ramificazioni illuminate da sinistra.

Quello che siamo riusciti ad ottenere con il motore interno è stato di avere luci con la giusta temperatura, controllo pieno del motion blur,

dello z-buffer e del color grading il tutto con render molto veloci.

Diverse scene sono state infatti renderizzate direttamente sui portatili mentre si lavorava su quelle successive.

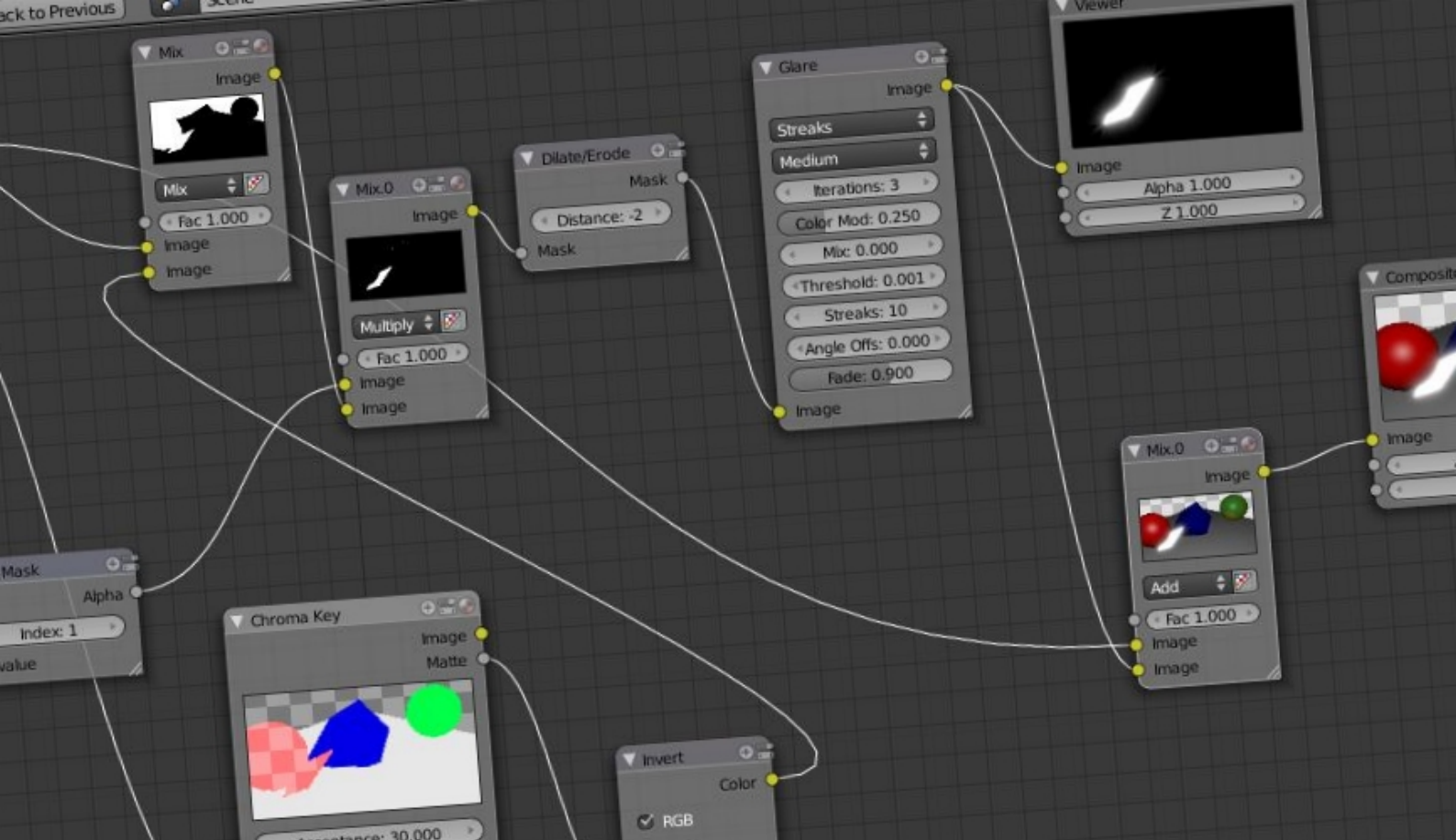
CONCLUDENDO...

L'esperienza che ci ha portato a chiudere il teaser di "Un mare da favola" è stata una serie di passi, spesso faticosi, molte volte frustranti, ma che ci hanno fatto percorrere moltissima strada.

Abbiamo imparato e inventato i modi per risolvere i problemi che sapevamo di dover affrontare ed anche quelli che non ci aspettavamo di incontrare, abbiamo man mano migliorato le scene in ogni aspetto, rifacendole molte volte, ritoccando i personaggi, i rigging, le animazioni, ridisegnando le texture, cambiando le luci, ricompositando il tutto infinite volte.

Il risultato è stato che vediamo ancora tanti difetti, ma se guardiamo ai primi risultati che ci avevano tanto entusiasmato adesso ci sembrano veramente inguardabili, dandoci conferma del progresso ottenuto.

Adesso stiamo facendo pratica in una parte ancora inesplorata: promuovere il teaser, cercare produttori seri ed intenzionati, dare la giusta dimensione a quello che finora è stato un grande sforzo collaborativo di passione e dedizione.



Trasparenze e alpha

Alpha, trasparenze Z-Transp e Raytrace con Blender 2.5

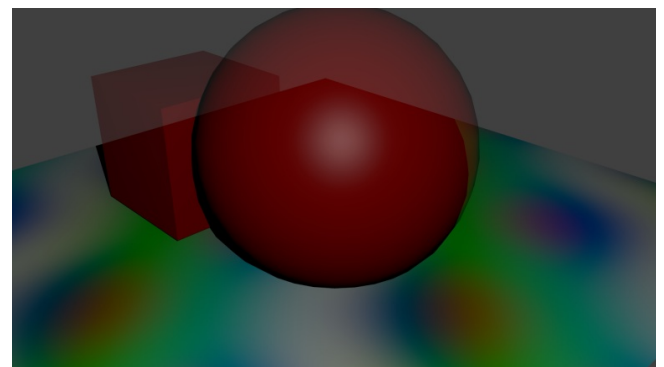
di Francesco "RedBaron85" Milanese

Questo tutorial nasce da una considerazione, espressa nel mio ebook "I Materials in Blender 2.5", riguardante alcune conseguenze derivanti dall'uso di Raytrace al posto di Z-Transp per la trasparenza.

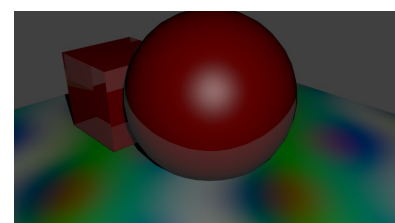
Come è noto, Z-Transp mantiene le informazioni sulla distanza degli oggetti rispetto alla telecamera ma non consente di implementare effetti propri dell'algoritmo di Raytracing (es.: la rifrazione), che per ogni pixel proietta uno o più "raggi" e li lascia viaggiare nella scena, collezionando ovviamente, ad ogni impatto/riflessione, informazioni su quanto "visto" da tali raggi.

Questo si traduce, in prima battuta, in una resa più realistica delle trasparenze e nella possibilità di implementare effetti come rifrazione o Glossiness, ma ci sono anche delle differenze non visibili osservando il rendering finale...

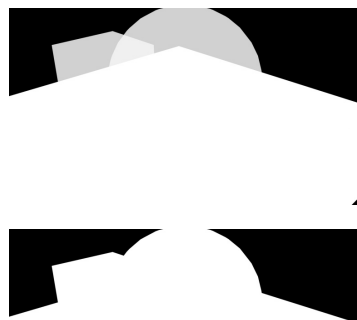
L'immagine 1 mostra il rendering di una scena effettuato applicando al cubo e alla sfera la trasparenza Z-Transp e Alpha a 0.4.



L'immagine 2 mostra il rendering della stessa scena effettuato applicando agli oggetti la trasparenza Raytrace, sempre con Alpha a 0.4 e, questa volta, IOR (indice di rifrazione del Material) 1.5.



Al di là dell'evidente differenza nella resa degli oggetti, per via dell'effetto di rifrazione nell'immagine 2, osserviamo, nelle immagini 3 e 4, il canale Alpha dei rendering effettuati, rispettivamente, con Z-Transp e Raytrace, ricordando che in entrambi i casi il valore di Alpha era 0.4.



Esportando il file immagine del rendering finale in formati con PNG, quindi, Blender assegna Alpha 1 ai pixel corrispondenti agli oggetti dotati di Raytrace.

Ciò può costituire un problema per una successiva rielaborazione dell'immagine mediante programmi di fotoritocco e compositing, perché bisognerebbe ricostruire la maschera di Alpha nelle zone corrispondenti... in realtà, sfruttando il Compositing dei Nodes di Blender, è possibile salvare capra e cavoli ottenendo un'immagine renderizzata con Raytracing e applicandole l'Alpha corretto nei punti desiderati.

In questo articolo, vediamo proprio come configurare l'editor dei Nodes per ottenere questo risultato, mostrando i Nodi da utilizzare e i loro collegamenti (per una trattazione approfondita dell'argomento rimando al mio ebook "Il Compositing in Blender 2.5", già recensito nel nono numero di questa rivista).

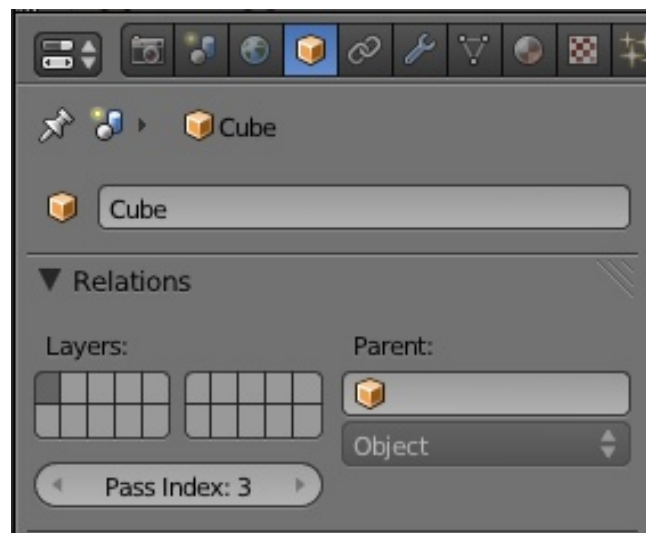
Anticipo subito che, nella soluzione qui proposta, dovremo fare ricorso all'onnipresente nodo ID Mask; tale scelta può rappresentare un problema per scene con molti elementi provvisti di Materials con valori Alpha differenti o con altri effetti da implementare, mentre per "casi base" va benissimo.

Nella seconda parte dell'articolo accenneremo comunque alla possibilità di creare uno schema dei Nodi ancora più articolato per applicare l'effetto a mesh con lo stesso Pass Index ma

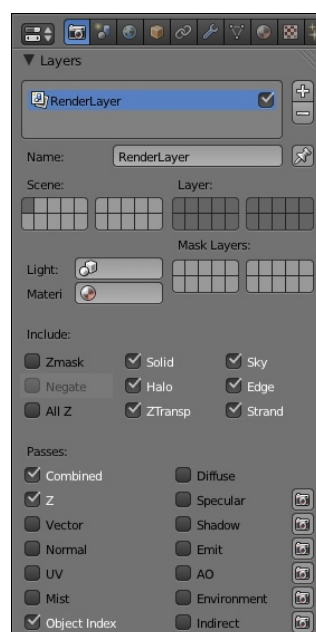
Materials differenti.

Diamo quindi un Pass Index specifico, ad esempio 3, al cubo e alla sfera.

E' possibile impostare il Pass Index per l'oggetto selezionato mediante il relativo pulsante numerico nella scheda Object, all'interno della Properties Window, raffigurato nell'immagine 5.



L'immagine dev'essere renderizzata con trasparenza Raytrace, in modo da implementare gli effetti IOR e Gloss (che non possono essere riprodotti in Post-Produzione), per cui opereremo sul canale Alpha in Compositing sfruttando le zone estrapolate da ID Mask, cambiando il valore di Alpha per quei pixel nell'immagine finale.



Per rendere disponibile il canale di Output "IndexOB" in Render Layers, selezioniamo la casella Object Index nella scheda Render all'interno della Properties Window, come visibile nell'immagine 6.

A questo punto possiamo passare alla configurazione completa dei Nodi: colleghiamo l'uscita

IndexOB di Render Layers ad un Nodo ID MASK con indice 3 per ottenere una maschera con Alpha a 1 solo per gli oggetti da isolare, dopodiché con un Nodo Math in modalità Subtract sottraiamo a tale maschera il valore 0.7 (per realizzare quindi l'Alpha a 0.3, come desiderato, in quei punti).

Abbiamo quindi due canali Alpha: quello in arrivo da Render Layer e quello in arrivo da Subtract, che ha sì Alpha a 0.3 per gli oggetti, ma 0 per tutti gli altri !

Dobbiamo combinare queste due immagini Alpha e passare il risultato al relativo canale di input di Composite (che riceve sul canale di Input Image l'immagine in arrivo direttamente da Render Layer)... questo è, in realtà, un piccolo problema, perché nella scena potrebbero esserci vari oggetti, con valori Alpha variabili, e dobbiamo preservare tali valori, modificando solo quelli di ID Mask.

Per risolvere questo problema, facciamo un passo indietro: nel Nodo Math Subtract, inseriamo come valore quello che vogliamo assegnare alla maschera, cioè nel nostro caso 0.3, ottenendo Alpha 0.7 per la maschera.

Adesso inseriamo un Nodo Mix nell'Editor, impostandolo come segue:

modalità: subtract;

Factor: l'immagine-maschera in arrivo da ID-Mask

Image 1: il canale Alpha del Nodo Render Layers;

Image 2: l'immagine in arrivo dal Nodo Math-Subtract.

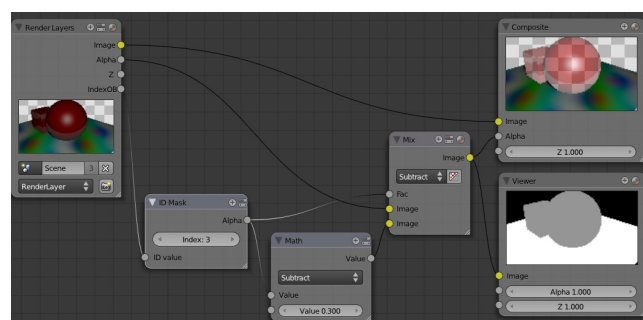
In questo modo, al canale Alpha originale (che, ricordiamo, presentava valore 1 per gli oggetti con trasparenza Raytrace), sottrarremo:

0 nei punti non estrapolati da ID Mask, preservando quindi il valore Alpha originale di quei punti;

0.7 (o, più formalmente, “(1 – valoreAlphaDaOttenere)”) nei punti selezionati da ID Mask, ottenendo per questi proprio il valore 0.3, come desiderato.

Il canale di output Image del Nodo Mix può essere quindi collegato al canale di input Alpha del Nodo Composite.

Come sempre, un'immagine vale più di mille parole: l'immagine 7 mostra lo schema appena descritto (con un Nodo Viewer associato all'output di Mix, per visualizzare separatamente il canale Alpha dell'immagine prodotta), rendendolo decisamente più comprensibile!



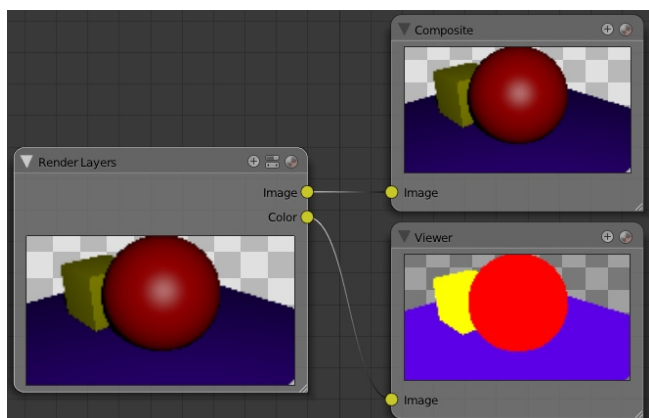
In presenza di oggetti semitrasparenti posti, come in questo caso, su oggetti opachi (il Plane), il risultato potrebbe non essere proprio corretto, per cui potrebbe essere necessario implementare altre maschere di filtraggio; in generale, comunque, le immagini con trasparenza IOR da utilizzare per il compositing in fotoritocco non presentano queste caratteristiche (a volte si ricorre a Blender solo per gli effetti di rifrazione, dopodiché si estrapola l'oggetto con tale effetto per inserirlo, ad esempio, in un depliant o una brochure).

C'è da dire, comunque, che il Nodo ID Mask non è l'unico strumento a disposizione per separare un oggetto dagli altri con una maschera in Compositing; ad esempio, in un tutorial (disponibile sul mio sito web) sulla realizzazione dell'effetto Glare stile film “Tron”, ho fatto uso anche del canale Diffuse di Render Layers (che restituisce appunto il solo colore diffuso per ogni oggetto, senza ombreggiature) e del Nodo Chroma Key.

Nel caso di più Materials differenti con effetti Raytrace Transparency, possiamo dare alle varie mesh lo stesso Pass Index per ricavare, inizialmente, un'unica maschera con Alpha a 1

per tali oggetti, ma sfruttare altri canali, con altri Nodi, per scomporre tale maschera in base ai diversi Materials ed applicare quindi livelli di trasparenza differenti.

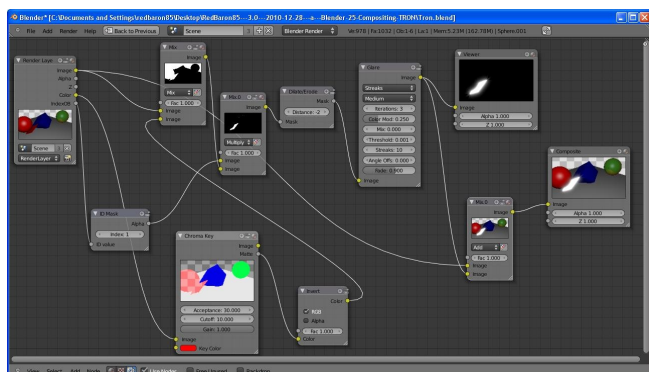
Un esempio di canale sfruttabile per questa operazione è COLOR, che restituisce il colore diffuso degli oggetti in maniera uniforme, senza ombreggiatura, come visibile ad esempio nell'immagine 8.



Il canale può essere utilizzato perché, in genere, se i Materials sono differenti, il loro colore diffuso è diverso; eventualmente, possiamo ripiegare su altri canali, ma il principio è lo stesso.

Applicando un Nodo Chroma Key ad una immagine ottenuta con COLOR possiamo creare una maschera di opacità, dopodiché sarà sufficiente moltiplicare tale maschera con quella in arrivo da ID Mask per ottenere una nuova immagine-maschera Alpha ancora più selettiva.

Questo processo è alla base del mio tutorial sull'effetto Glare stile Tron; la configurazione dei Nodi di quella scena è visibile nell'immagine 9.



Per questo tutorial è tutto !

Bye ! ;-)

--- Francesco "RedBaron85" Milanese, Blender Foundation Certified Trainer.

Ha realizzato e gestisce il sito web "RedBaron85.com" (www.redbaron85.com), dove pubblica periodicamente videotutorials e guide sull'utilizzo di Blender ed altri programmi. Ha inoltre pubblicato articoli su Blender Magazine Italia e tenuto corsi, seminari e consulenze su vari software e linguaggi di programmazione riguardanti la CG 3D.



Fermati Tempo!

Il Making Of della copertina di questo numero di BMI

di Guida Pasquale

Salve lettori, di seguito parlerò brevemente della realizzazione del lavoro "Fermati tempo!".

PASSO 1: L'IDEA.

L'idea è nata quasi per caso mentre vedevo il film "Alice in the wonderland". Vedendo il Bianconiglio, non ho potuto fare a meno di pensare alla sua celebre frase "è tardi! è tardi!", e per quanto possa sembrare buffo, noi e il Bianconiglio non siamo tanto diversi (apparte per la peluria bianca e le orecchie lunghe). Anche noi, ormai, siamo schiavi del tempo.

E come possiamo liberarci dalle catene che ci legano al tempo? Bhè, è semplice: "fermandolo!". E chi poteva rappresentarci se non lui? Questa è stata su grandi linee, l'idea di fondo.

Dopo aver abbozzato su carta l'idea, ho iniziato la modellazione.

PASSO 2: LA MODELLAZIONE.

Per la modellazione non ho usato alcun tipo di reference impostato come immagine di background. Come primo oggetto della scena, ho modellato il coniglio partendo dalla testa.

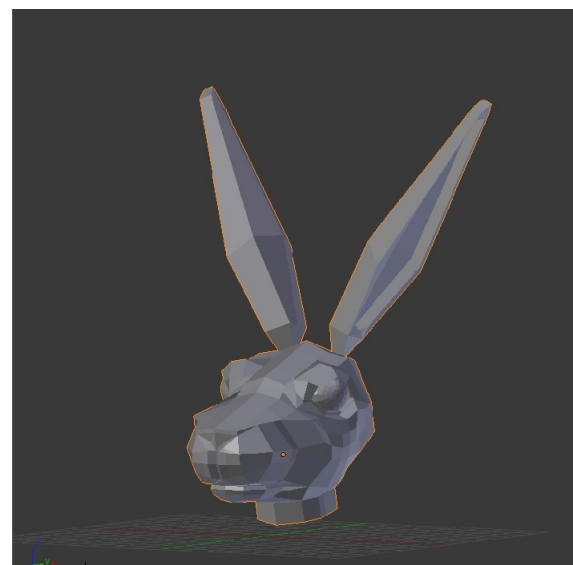


fig1

Successivamente ho modellato il corpo fig1 e fig2.

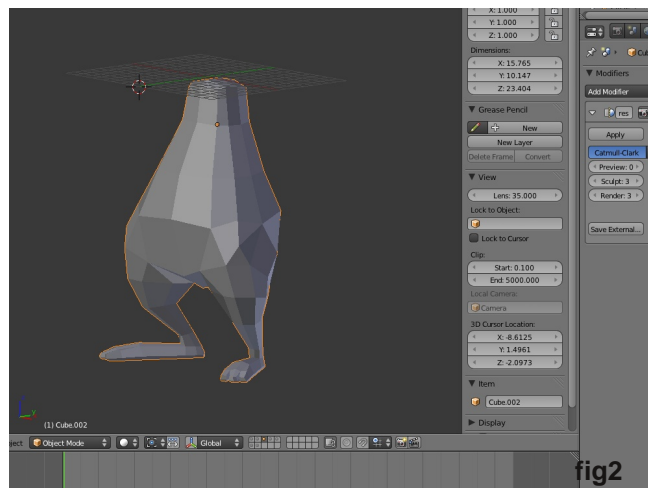


fig2

Dato che il coniglio indossa una giacca, collo e braccia vengono nascoste, e pertanto non le ho create.



fig.g3



fig.4



fig.6

fig.5

Dopo aver modellato anche gli indumenti, e dopo aver applicato ad ogni oggetto un modificatore Subsurf il coniglio si presenta così: fig.3.

Per metterlo in posa inoltre, ho creato una armatura molto semplice che controlla giusto i vertici necessari per fargli assumere la posizione abbozzata su carta in precedenza fig.4.

Ho messo il coniglio in posa e posizionato anche la camera per avere l'inquadratura desiderata.

Dopodichè ho iniziato a modellare il grande orologio con il vetro rotto. La parte più lunga della modellazione di questo oggetto è stata quella degli ingranaggi interni fig.5 e fig.6.

Per gli ingranaggi dentati ho modellato prima un quarto della mesh e poi tramite il modificatore mirror l'ho specchiato prima su un asse in modo da ottenere un semi ingranaggio e poi sull'asse opposto ottenendo un ingranaggio completo fig.7.

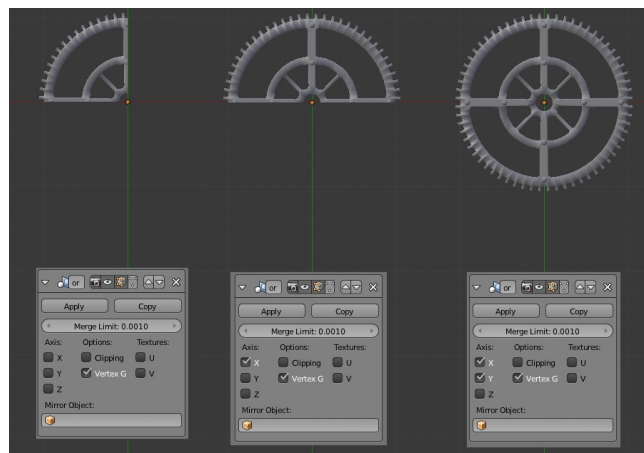


fig.7

Ho modellato successivamente altri due piccolli

oggetti che il coniglio impugna: la mazza da baseball e il suo orologio da tasca, dove la catenina l'ho ottenuta applicando al primo anello un modificatore array e successivamente un oggetto curve fig.8 e fig.9.

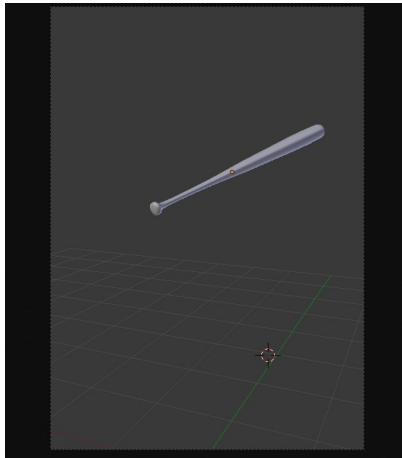


fig.8

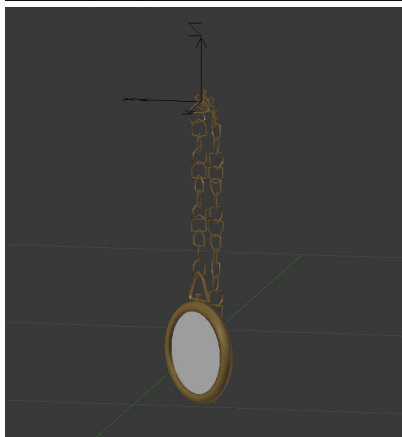


fig.9

Passando allo sfondo, le rocce le ho modellate seguendo il tutorial sul numero precedente di questo favoloso magazine, BMI9, grazie al quale nel render ho ottenuto risultati più che soddisfacenti fig.10.

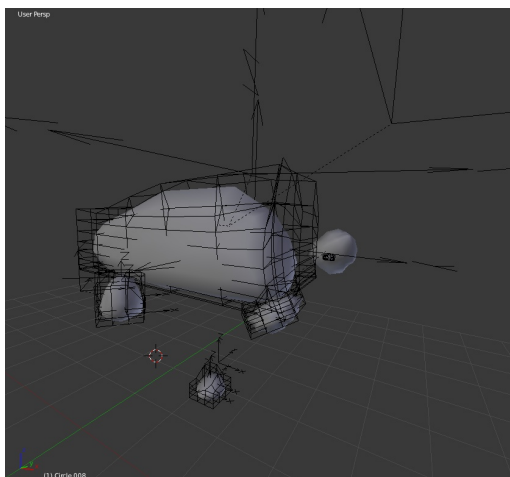


fig.10

Infine ho modellato gli ultimi elementi di scena come l'erba, dove non ho usato particelle ma tanti gruppi di mesh a forma di filo d'erba posizionate nel modo adeguato, e i funghi giganti che fanno da sfondo fig.11e fig.12.

La scena finale è stata questa fig.13.



fig.11

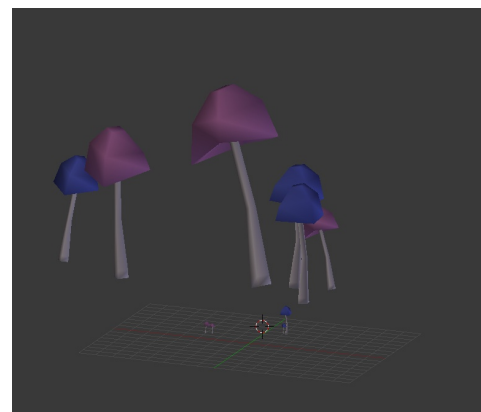


fig.12

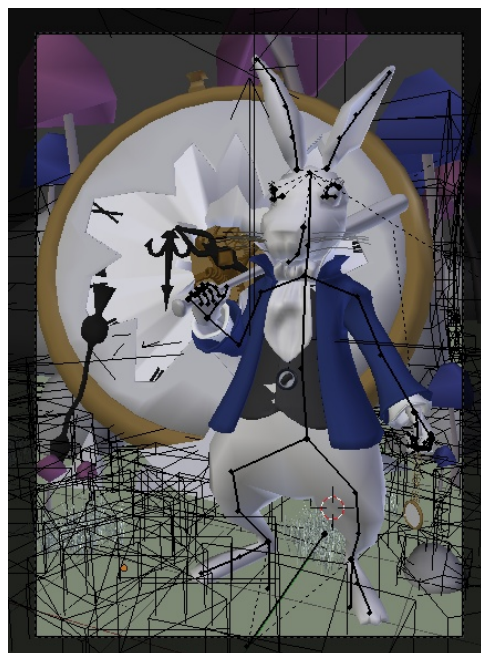


fig.13

PASSO 3: I MATERIALI

Dopo aver creato una mappa UV del solo corpo del Bianconiglio, gli ho applicato una texture nera, colorata in seguito grazie al mode Texture Paint (fig.14).



fig.14

Per il gilet e la giacca ho usato rispettivamente una texture di un tessuto in stile scozzese e una texture di un tessuto in jeans. Per i funghi ho assegnato due materiali distinti: uno per lo stelo ed uno per la testuggine. Il secondo materiale l'ho ottenuto applicando 2 texture: una clouds e una distorted noise, entrambe con il normal map attivo per dargli un effetto più realistico (fig.15).

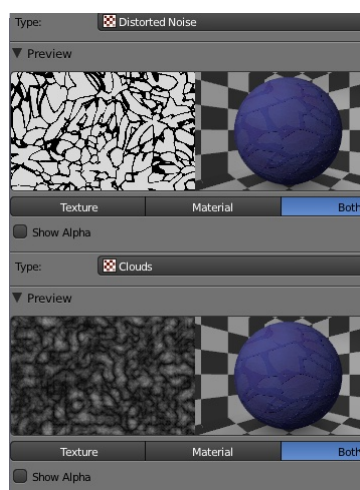


fig.15

Per gli altri oggetti di scena sono stati applicati semplici materiali. Per l'illuminazione ho usato una semplice lamp point per creare una sorgente luminosa e mi sono affidato all'Ambient Occlusion e all'Environment Lighting.

PASSO 4: IL RENDERING

Il rendering non è stata cosa semplice. Ho usato l'internal di Blender e molto spesso l'applicazione crashava, maggiormente quando aggiungevo, tramite i particles, i peli al coniglio. Per questo motivo, ho "depilato" il coniglio facendogli perdere un effetto più realistico. Ma vabbè ...

Finalmente dopo quasi un'ora di rendering il lavoro è quasi pronto e come ultimo passo, tramite il compositing in un file a parte, ho schiarito la scena ed ho rinforzato un po' i contorni per ottenere un risultato meno sfocato e più illuminato.



Questo è stato il mio primo vero lavoro e devo dire che sono rimasto molto soddisfatto dai risultati.

Un ringraziamento a BMI e a tutti voi che avete letto questo articolo.

Saluti, Guida Pasquale.



Tecnologia

Contest per scegliere la prossima copertina di B.M.I.

Il tema del contest per la copertina del prossimo numero di B.M.I. è "Tecnologia".

Ho sempre creduto che qualsiasi tecnologia abbastanza avanzata sia indistinguibile dalla magia. Forse è per questo mi affascina il tema del prossimo contest... e a voi?

Il regolamento è sempre lo stesso:

- Le immagini devono essere eseguite principalmente con Blender.
- Si possono eseguire anche più lavori, ma ricordate che conta la qualità non la quantità.
- Sono permessi motori di rendering esterni.
- Sono permessi altri programmi esterni ma solo per ritocchi di postproduzione ed a patto che siano Open.
- Le immagini devono avere queste dimensioni: 1024*1448 px o superiore, basta rispettare la proporzione A4.
- Le immagini possono essere postate come Wip su Blender.it (sezione immagini), Kino3d (sezione open lab) o Blender-Tutorial, con la dicitura [BMI11] prima del titolo.
- I lavori finiti devono essere mandati

all'email anfeo@libero.it **entro il 22 maggio 2011.**

- Ricordate di specificare il titolo dell'immagine e il nome che volete usare come firma (nick e/o nome vero).
- Il vincitore avrà l'onore di vedere pubblicata la sua immagine sulla copertina di Blender Magazine Italia.
- Il vincitore potrà scrivere un piccolo Making Of del suo lavoro che sarà pubblicato sulla stessa rivista elettronica. L'articolo non è obbligatorio.

Al termine della data di consegna, verrà aperto un sondaggio affinché, chi vuole, possa votare, tra le immagini partecipanti, quella che vorrebbe vedere sulla copertina del prossimo numero di Blender Magazine Italia.

Gallery dell'ultimo Contest

Si è concluso il contest "Tempo" che ha decretato la scelta della copertina di questo numero di BMI.

Vediamo tutte le immagini che hanno partecipato a questo concorso.



E' tardi
- Piter85



The Time Machine
- Fabio Russo



I segni del tempo
- Fabrizio Sirtori



Cell's time machine
- Gorex87



Fermati tempo!
- Pasquale Guida



Padre Tempo
- SandroP



What is time
- Wolverine



▲ RED KITCHEN

NOME: Kister

EMAIL: kister1@hotmail.it

SOFTWARE: Blender 2.49, yafaray

▼ TEMPIO ROMANO

NOME: Alessandro Battisti

EMAIL: alessandro.battisti@gmail.com





Filippo Veniero (2011)
www.ifilgood.net

▲ BOTTLES

NOME: Filippo Veniero (ifilgood)

WEBSITE: www.ifilgood.it

SOFTWARE: Blender 2.49, Yafaray, Gimp

▼ SALA RIUNIONI

NOME: Alessandro Battisti

EMAIL: alessandro.battisti@gmail.com





▲ BAGNO

NOME: Bernardo Iraci

EMAIL: bernardo@mccinfo.it

SOFTWARE: Blender 2.49b, yafaray

▼ BAD FELLA

NOME: Andrea Fiori

EMAIL: waafcg@gmail.com

BLOG: waafcg.blogspot.com





Blender Magazine Italia
numero 10
anno 2011

Responsabili:
Alfonso Annarumma (Anfeo)
Luca Pinciani (Sinistar)

Collaboratori:
Gikkio

Grafica:
Davide_G

Siti di riferimento:
www.blender.it
www.kino3d.com
www.blender-tutorial.com
www.blender.org

Software utilizzati:
Blender
Scribus
Pdftk
The Gimp
Openoffice

Vuoi collaborare con noi?

Scrivi un articolo per Blender Magazine Italia

Ti stiamo aspettando!

Le indicazioni per scrivere un buon articolo sono semplici:

- Scrivere un documento di testo apribile da OpenOffice senza immagini ma con i riferimenti per inserire l'immagine corretta nella posizione esatta.
- Creare una cartella con le immagini e nominarle come scritto nel testo dell'articolo (ad es. fig1.jpg, fig2.png).
- Creare una immagine per l'header dell'articolo.
- Fornire (in modo facoltativo) descrizione personale, contatti e sito internet dell'autore.

Invia l'articolo all'indirizzo
anfeo@libero.it

SITO WEB

Tutti i numeri di
Blender Magazine Italia
sono disponibili
per il download gratuito
all'indirizzo

www.BlenderMagazinItalia.it