MEERKAT  PIPELINE

## INTRO

## FROM THE SHOOT

You will be receiving some data from the shoot that will help you integrate characters. This data is the only thing that links CG to the real world.

You should receive 4 types of data: Live Plate, Visual References, Measurements and HDRs:

### *Plate*

This is what as been shot and what we will animate and integrate the characters on.

Animation BackGround:

The plates need to be exported as an Animation Background (Used as Rotoscop in the XSI camera)

It should be saved as a 1024x576 Jpg in the following location:

P:\ [Project] \Incoming_Video\Animation_BG\[Shot]

It needs to mach the plate edit and be numbered from frame 0125
i.e.: SL_01_00.0125.jpg:

Compositing:

The plate and all the In Camera References will be exported from Hero as a nuke comp here:

P:\[Project]\Comp\[Shot]\Process\[Shot]_Plate_v000.nk

This is the plate that will be use to comp the shot. It's using a DPX sequence that as a greater quality then the Jpg exported for the camera.
NEVER use the animation background for comping.

Rendering BackGround:

We will use a rendering BackGround (Used as Rotoscop in the XSI camera) created using the reference shot in camera. See how to create it here: Rendering BackGround

***References***

References are taken in camera or with a DLSR.

In Camera

Those are the most important references; they will be use through the whole production to make sure that the characters are correctly integrated.

Nuke Setup:

In the nuke comp located here:

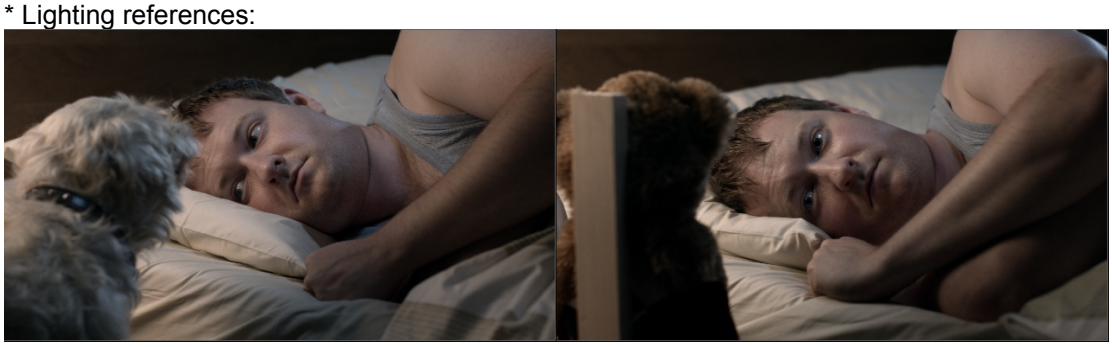
P:\[Project]\Comp\[Shot]\Process\[Shot]_Plate_v00x.nk

You will also find:
* A Grey Ball:

* A Mirror Ball:

* A scale reference:

* Lighting references:



This is a BackUp, do not change this nuke file as it is generated automatically using Hero.
Instead save it as a comp file here:

P:\[Project]\Comp\[Shot]\[Shot]_v00x.nk

Or copy/pass the plates in your current comp.

Make sure to properly label the Reference plates by renaming the read nodes and using sticky notes (alt+n) and BackDrops:

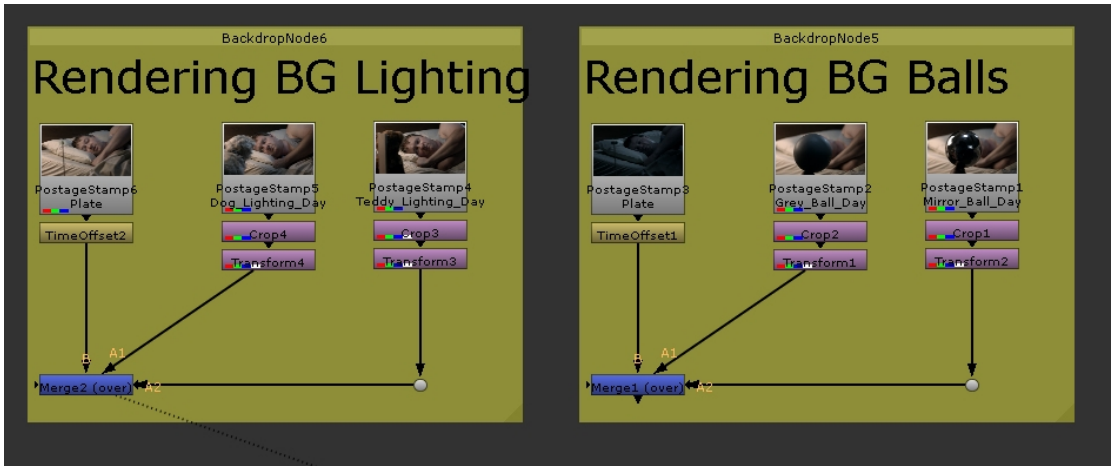

Rendering BackGround:

You then need to create two Rendering Backgrounds. They will be used in XSI and nuke as lighting reference. They should look something like this:

It is better to set those BG in nuke so they can easily be updated if needed.

To set them up you can pass this code in nuke:

“ CODE “




You then need to:
Connect the PostageStamps to the right image sequence (make sure to connect it to the Frame range node)
Adjust Cropping and Translation
Adjust the TimeOffset to show the right frame.

You can then render the Backgrounds as 1920x1080 Jpg to the following location:
P:\[Project]\Shots\[Shot]\Reference
Named: [Shot]_BG_Balls.jpg and [Shot]_BG_Lighting.jpg
i.e:

- SL_01_00_BG_Balls.jpg
- SL_01_00_BG_Lighting.jpg

We will be using those backgrounds in the Lighting section.

DSLR

If needed, pictures of props and texture references will be taken on set in order to recreate them in CG. They should be taken with a flat lighting.

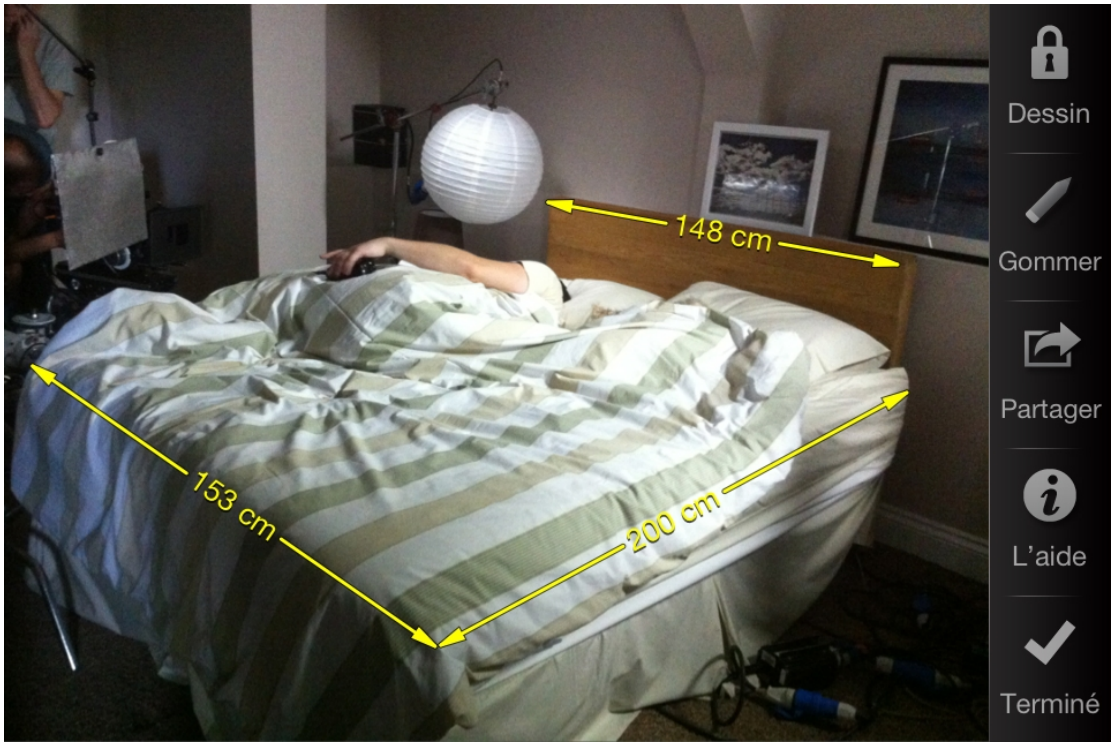
***Measurements***

Measurements are very important as they’ll give you real scales and proportion between the set, the camera and the CG.

You will receive camera measurements:

- Focal length
- Camera height
- Camera tilt
- Distance to object

And also props measurements:




Those in formation will allow you to build the animations and environment scenes. See CAMERA & SET

***Hdr***

You will be receiving Panorama Pictures with different exposure they will need to be sort out and then made into and HDR. The Mirror Ball Reference will also be used at this stage.
See HDR for detailed step by step.

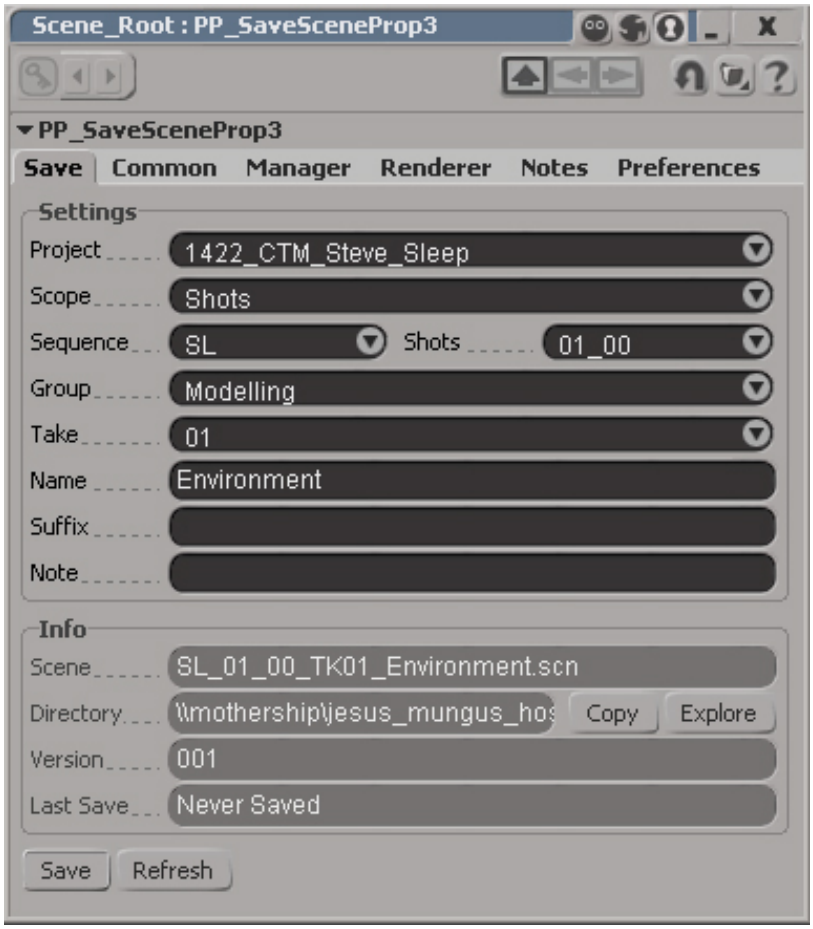
**CAMERA & SET**

This section will detail how to setup the camera and the set so it can be scaled and position properly.
The relationship between the camera, the set and the CG has to be right to make the integration work.

***Environment Setup***

You will first need to separate the shots into deferent's setup depending on what changed per shot. If anything moved in between two shots, you will have to create different setups. So you would end up with something like:
Setup1: SL_01_00, SL_02_00
Setup2: SL_03_00
Etc
Once an element is setup as follow, you don't need to redo the whole process. To create setup 2 you'll be able to save SL_01_00 as SL_03_00 and just move the objects that moved.

Start by saving a scene in Shot/Modelling. Call it environment:

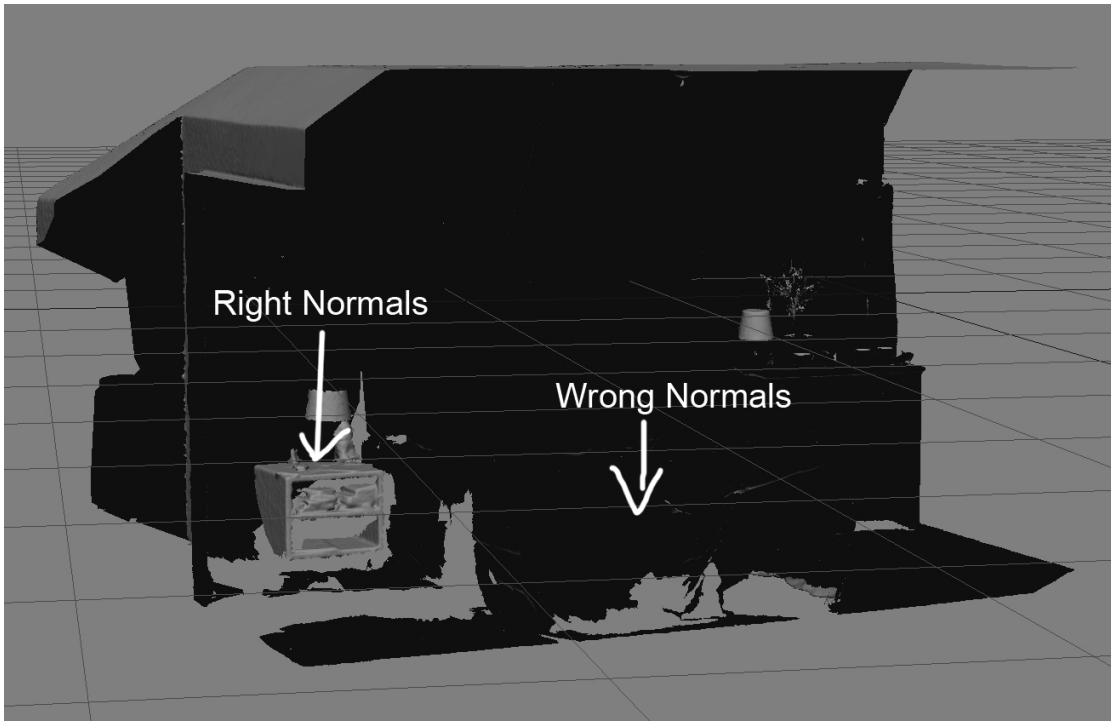



At this point you're ready to build the scene.

Import Modelling from:

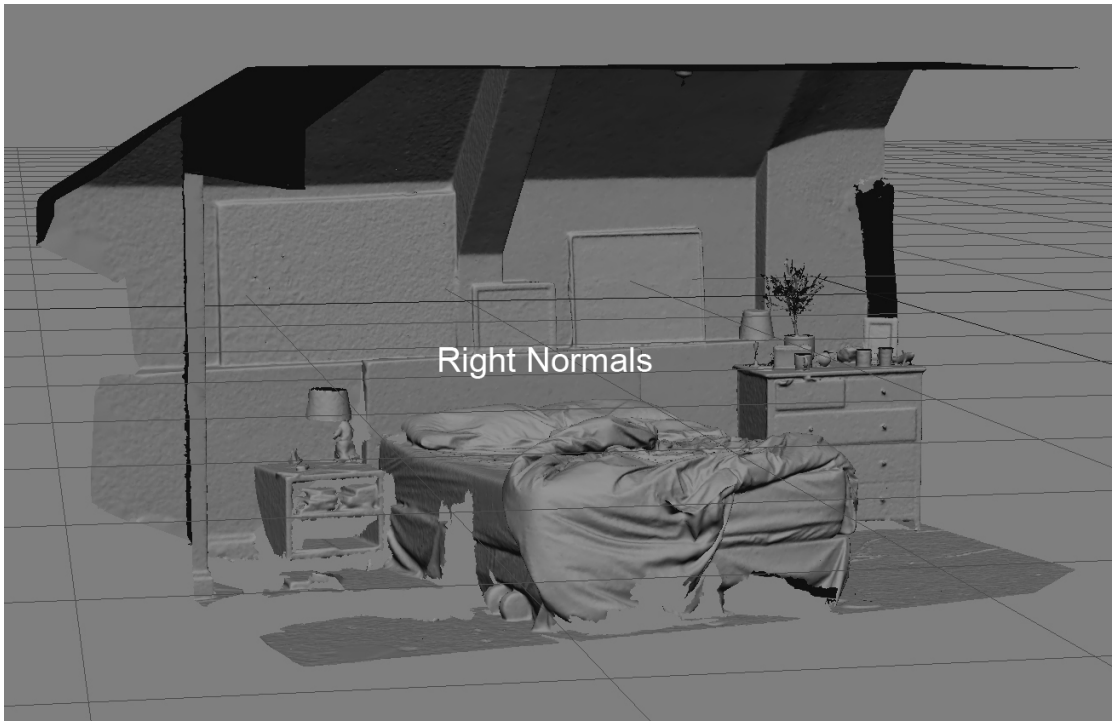
3D Scan

The 3D scan will come as a .obj located here:
P:\[Project]\Incoming_Assets\Scans
Usually the lower rez will be enough. You can check the size of the .obj to know witch rez is the smallest.
Import the .obj in xsi. The mesh will probably be out of scale and not in the right position.
Normals will probably be wrong too.




To correct the normals, select the mesh, go to select polygons (u) and then switch to Polygon Island and select the polygons that have wrong normals.


Then go to model> Modify poly.Mesh> Invert Polygones



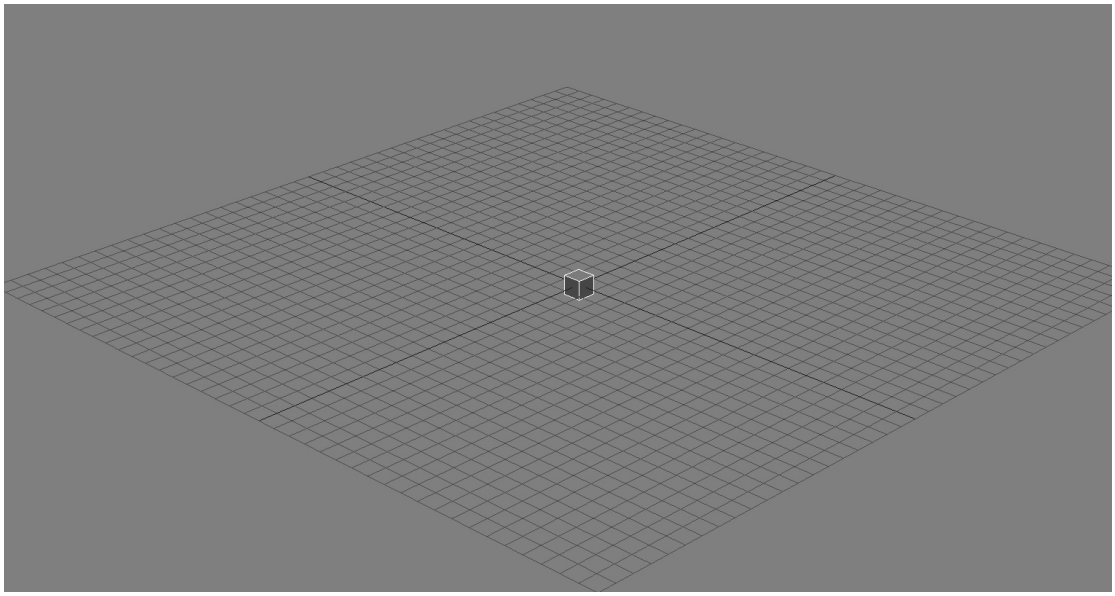

Once you're happy with all the normals, freeze the mesh.
The scale needs to be fixed now. See Scene Scale

123D Catch

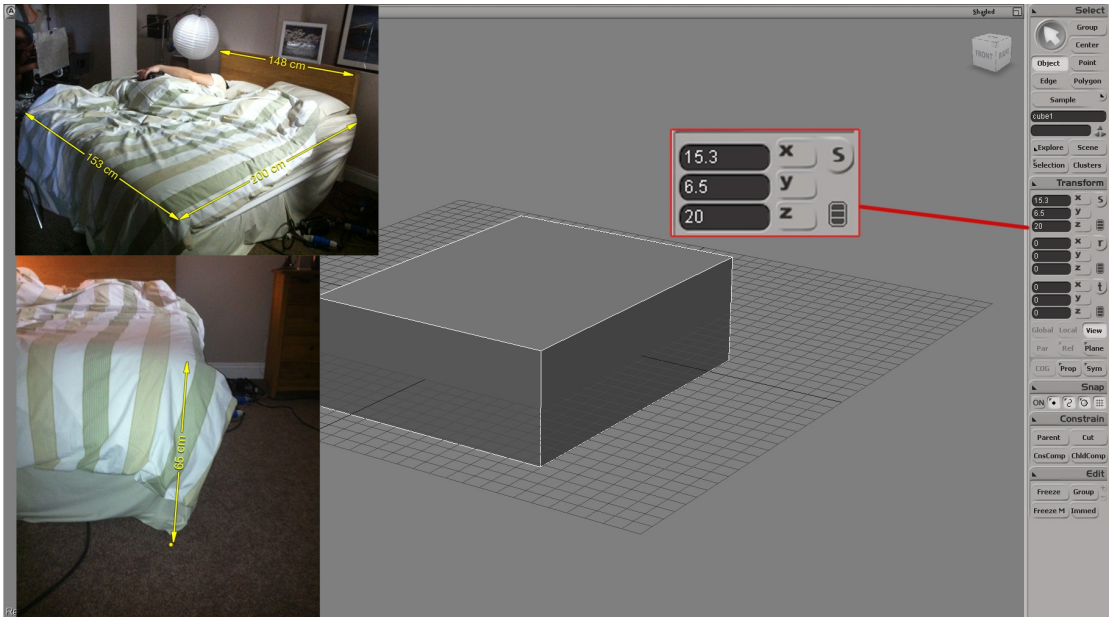
You will need photos of the set.
You can then sownload 123d cath from here: http://www.123dapp.com/catch#downloadCatch
(You will need a login, talk to your supervisor to see how to get it installed on your machine)
The quality of the mesh will depend greatly of the quality of the pictures. You should go through the photos and select only the ones that are sharp and shat with the same exposure.
Copie them into here:
P:\[Project]\Incoming_Assets\123Catch\[Asset]\Photos
Laugh 123D Catch and sign in.
Click Create a New Capture and load all the pictures.
Click Create Project, enter the information asked.
Then click wait or email me.
If the result is not good enough you can manually add traking points.

On set Measurements

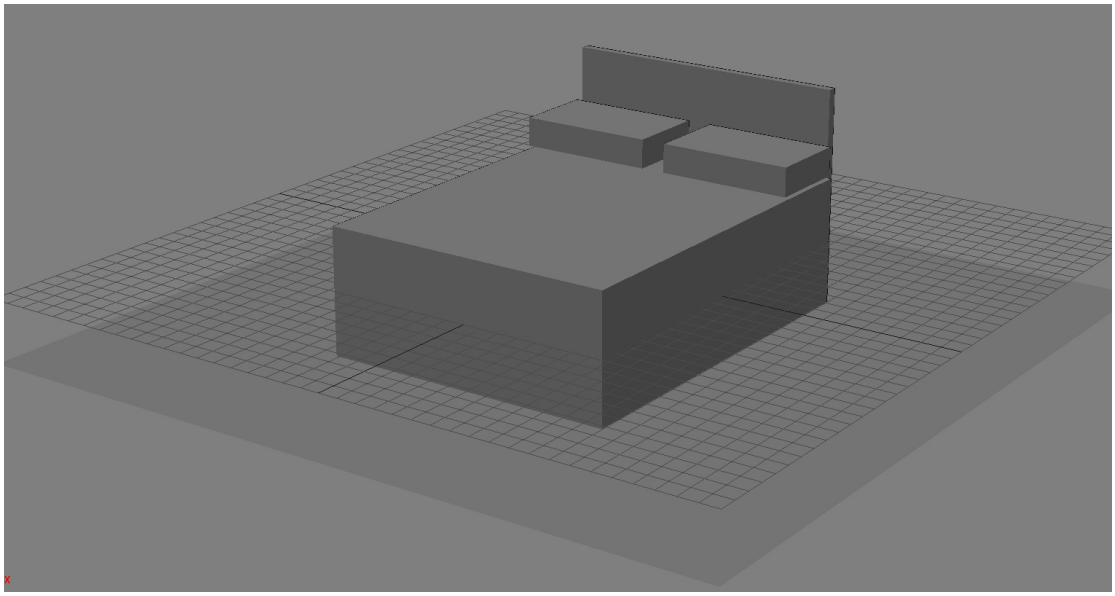
Using the Measurement pictures taken on set create simple geometry that matches the scale
1 xsi unit = 10 cm:
Create a cube with 1 of length (10 cm):



Scale the cube to match the on set measurements:



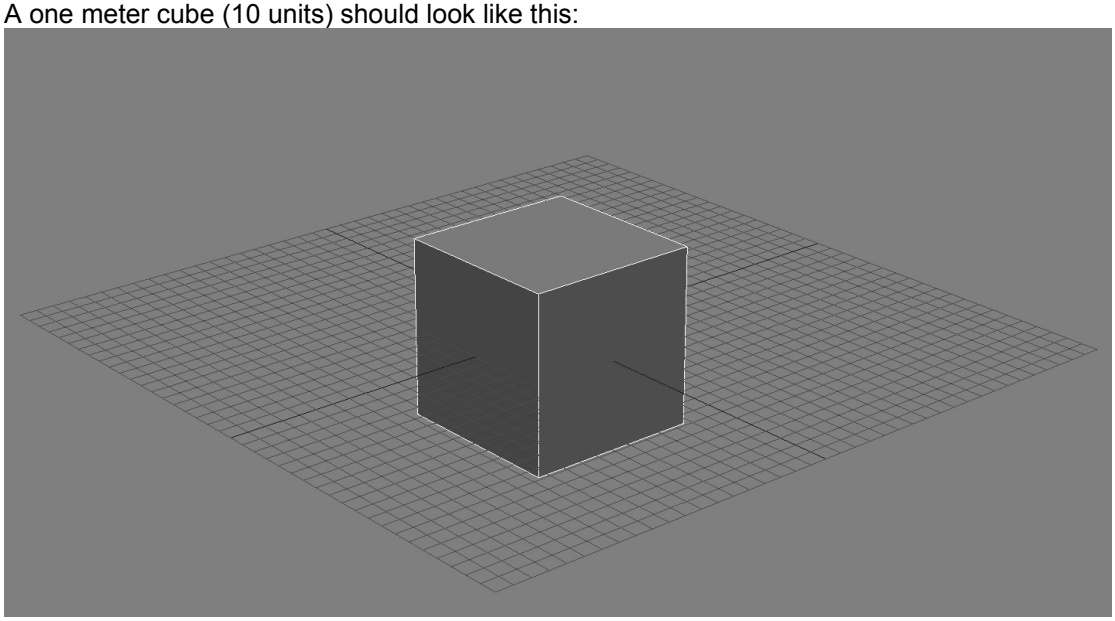

Do this for every measured elements (if no measure is provided, use websites like Wikipedia to find out common sizes of object.):



Name properly each modelling and then Freeze there scales.
You can now move on to positioning the environment.

Scene Scale

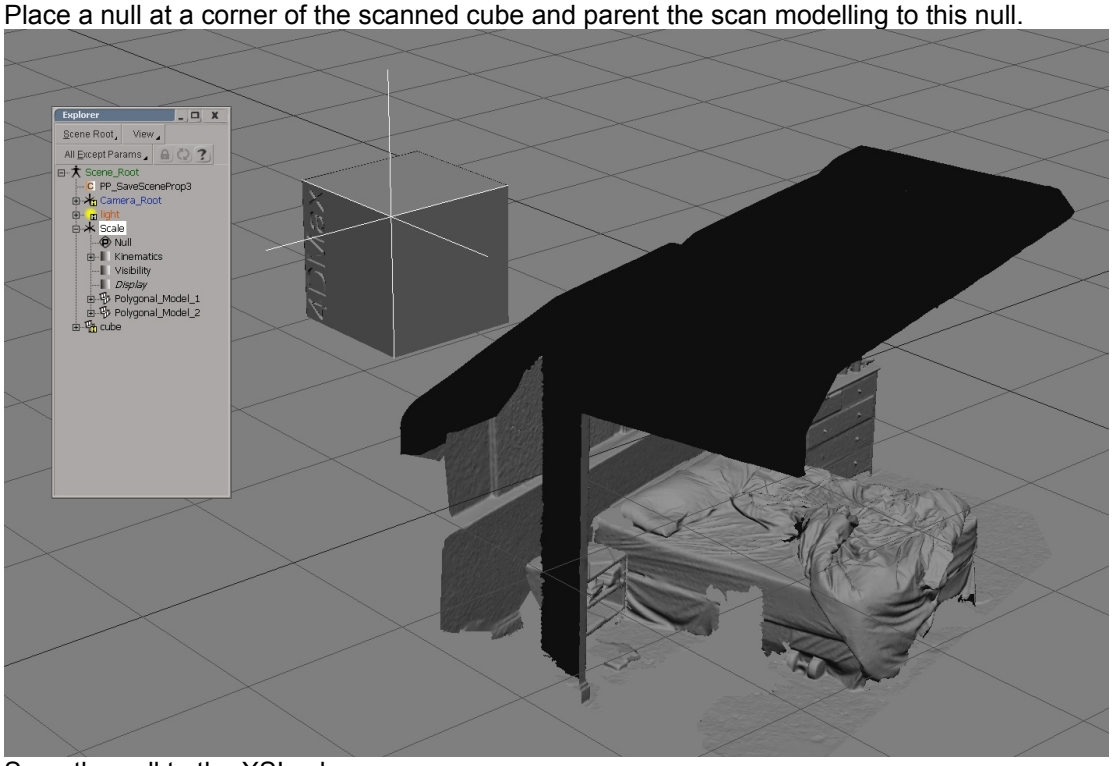
As a starting point the modelling scene should be scaled to the real world units. We use 1 XSI unit = 10 cm
A one meter cube (10 units) should look like this:



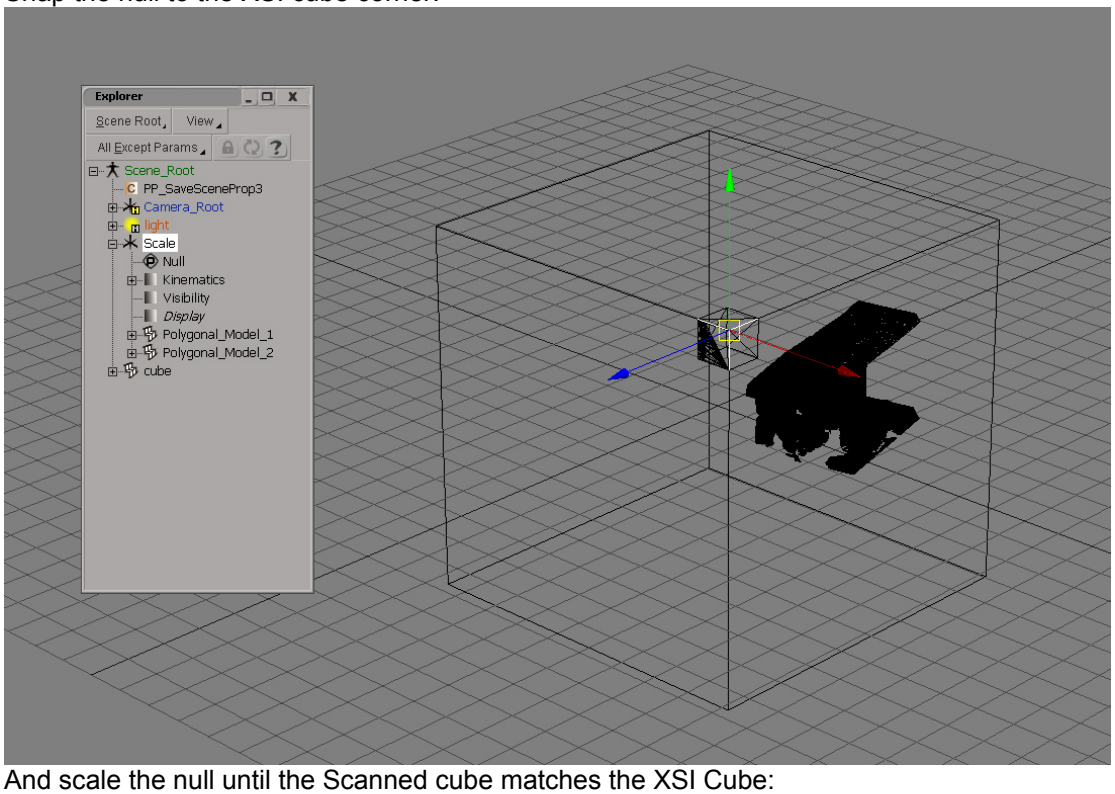
Scan

The Scan comes with a 1meter cube reference; match it to your xsi 10 unit cube:

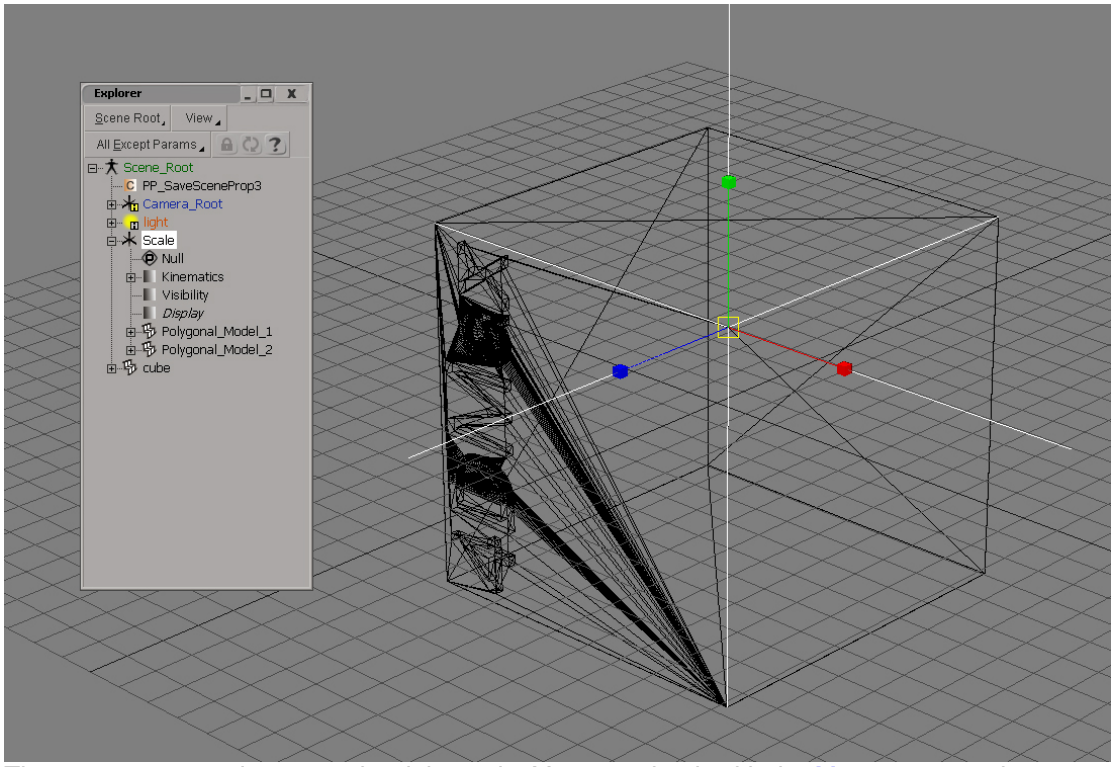
Place a null at a corner of the scanned cube and parent the scan modelling to this null.




Snap the null to the XSI cube corner:




And scale the null until the Scanned cube matches the XSI Cube:




The scan geometry is now at the right scale. You can check with the Measurement pictures taken on set.
Since object don't change scale per setup, it is better to remove the Model from any parent and Freeze the scale.
You can now move on to positioning the environment

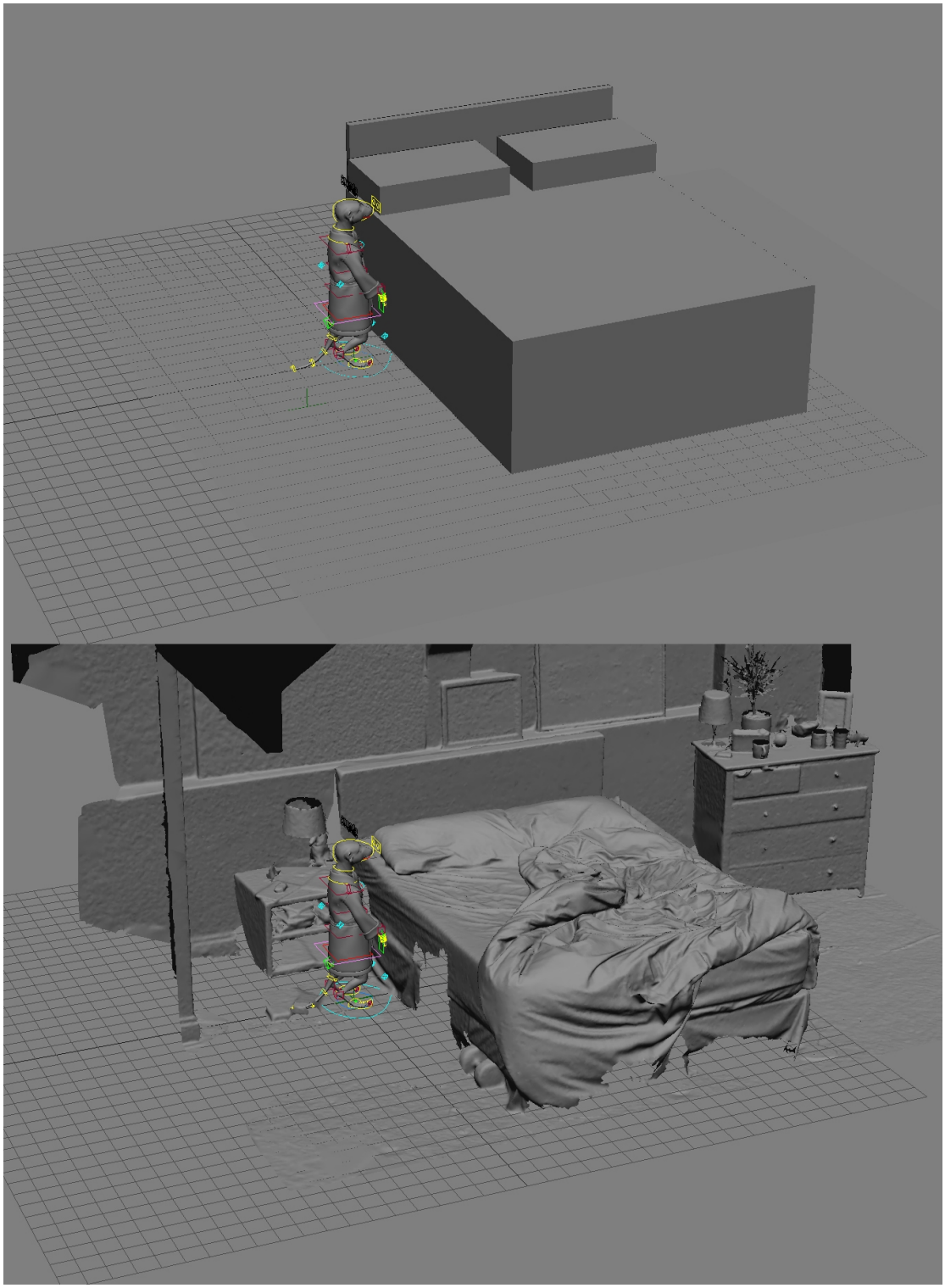
123D Catch

On set Measurements

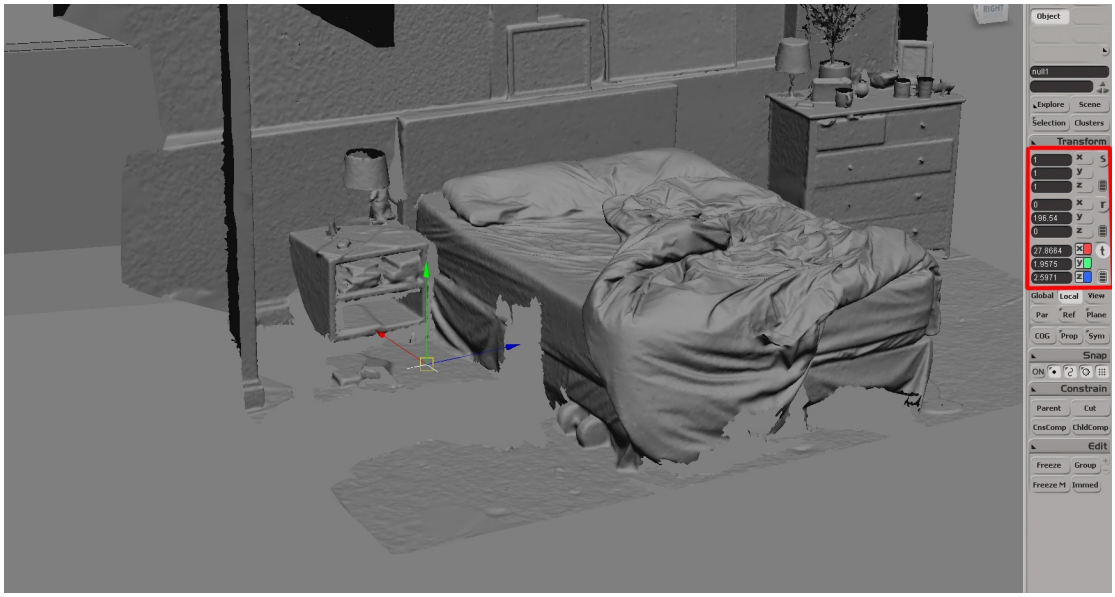
On set measurements are modelled directly at the right scale, you can find a step by step here: Scene Scale>On set Measurements

Positioning Scene

*At this point your modelling should be at the right scale, if not follow those steps Scale Scene*
The goal of positioning the scene properly is to be able to easily place the characters, and scale the set depending on the Characters.
First import your main character as a reference. It should be in the centre of the world with its feet on the grid. The model should NEVER be scaled. This would break the render scene.
*If the animation model comes in with a scale different from one, contact a rigger in order to fixe it. If a rendering model comes in with a scale different from one, see* SHADING>Characters
Parent all the modelling to a null. Then roughly place the set relatively to the character using the parent null. The ground level should be accurate:



TIP: When using the Scan modelling, this process can get quite slow. You can place the null where the character should be, using the snap tool. Only then, parent the modelling to the null and then set all the null transforms to 0. This will give you the same result as moving the set around, but only in a few clicks, avoiding the Open GL lag.



You can now remove the modelling from its parent and freeze all transforms.
And go to Camera Setup

***Camera Setup***

*At this point all modellings should have no transform, the ground plane should match the XSI grid and the scale should match the real life scale. If not see Scale Scene and Position the scene.*
For setting up the camera you will need the Animation Background and the camera info from the shoot (Focal length, Height, and Tilt, camera film back)

Static Camera:

Create a new Camera.
Name It:[Shot]_Camera.
Delete the Direction Cst (In kinematics) and the Camera_Interest null.
Set it's transforms to 0
Rename the Camera_Root to Camera_Tilt. Set it's transforms to 0
Creat a new null. Name it Camera_Height. Parent the Camera_tilt to this null.
Creat a new Null. Name It Camera_SRT.

You now need apply the camera information:

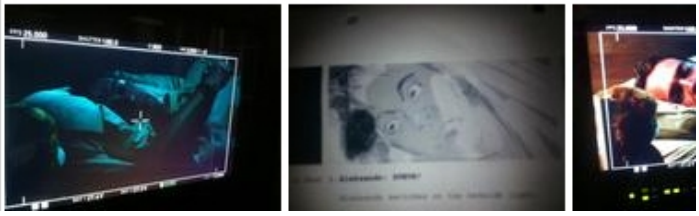
# Sleep · day 1 (Aug. 23, 2012)

**Director of photography:** Ed

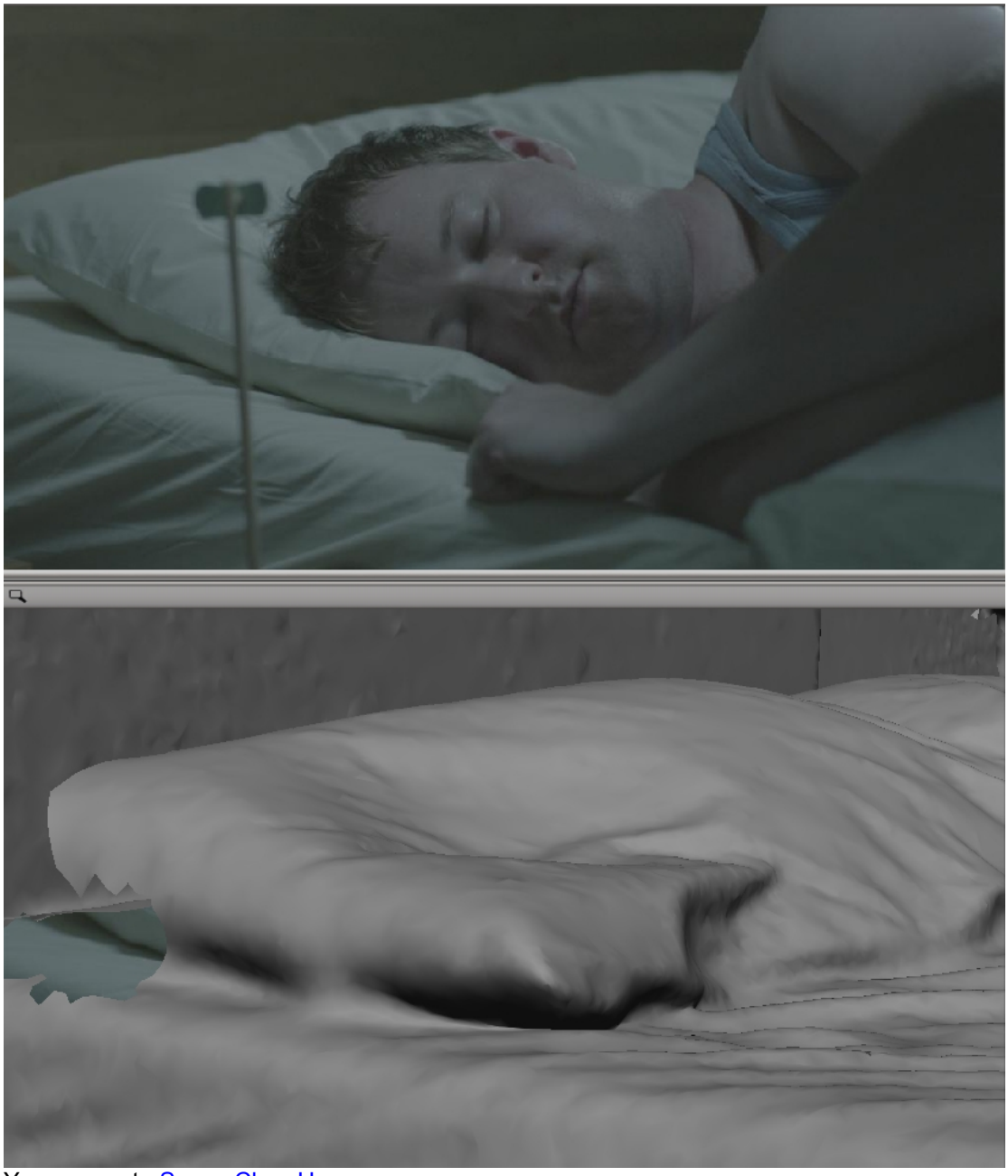
## Shot #1

| Sc# | Lens | Dist. | Orient. |
|---|---|---|---|
| 1 | Canon 50mm | O: 49<br>G: 28 | P: 0<br>T: -6<br>D: 5 |

| Ta# | Notes |
|---|---|
| 1 | |



Here O is distance to object, G is distance to ground, T is tilt.
Make sure the distances are in the right unit (i.e cm)
Apply the Ground distance to the Y axis of Camera_Height and the Tilt to the X rotation axis.
You need to set up the focal length. Make sure the Film Back is properly setup in Camera>Projection Plane>Film Aperture. Film Back can easily be found by googleling the camera used on the shoot.
You can now set the focal length in the same tab.
In the Rotoscopy Option, load the animation BackGround. And activate the Rotoscop.
Using the Camera_SRT you should be able to easily find the camera position. You should move it ONLY on X and Z axis. And rotate it ONLY on Y axis. But it can happen that the height nee to be slightly adjusted, do this using the Camera_SRT
The object distance to camera can give you a rough idea of where the camera should be. You can also use the Photos taken on the shoot to see where the camera was.

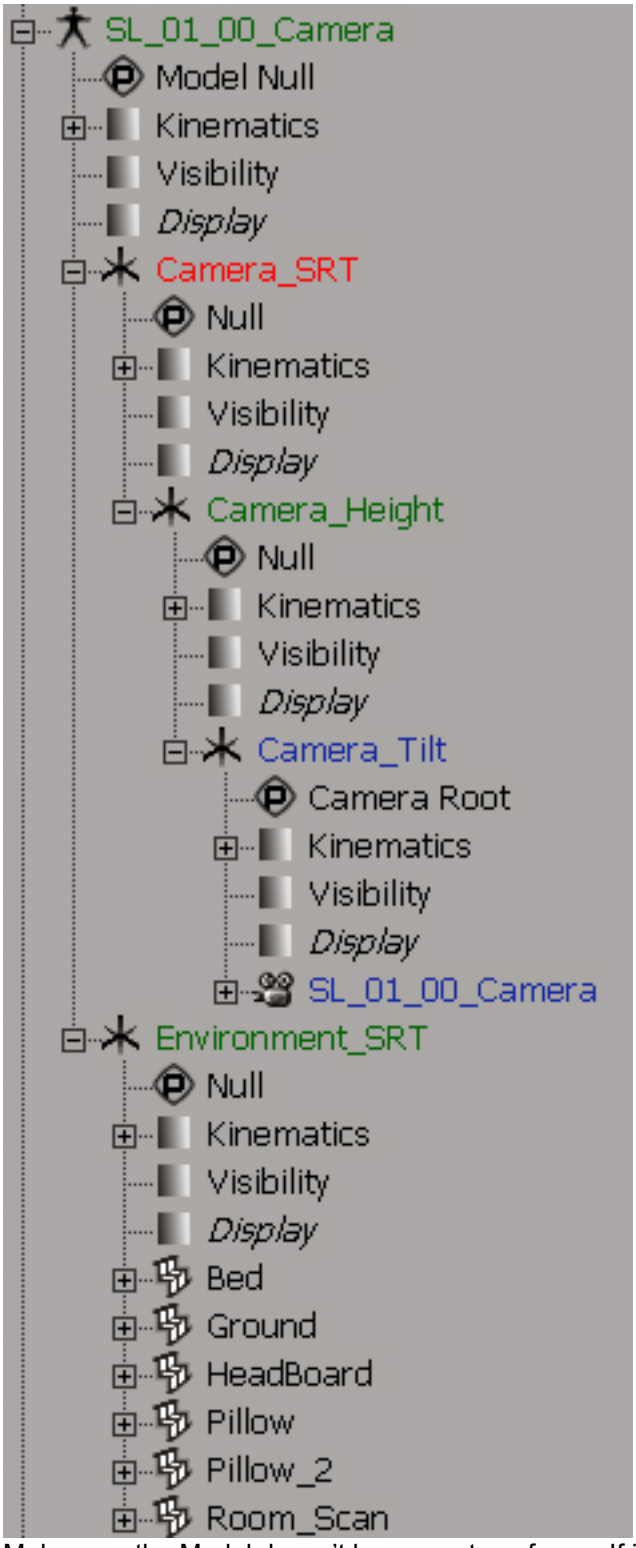


You can go to Scene CleanUp.

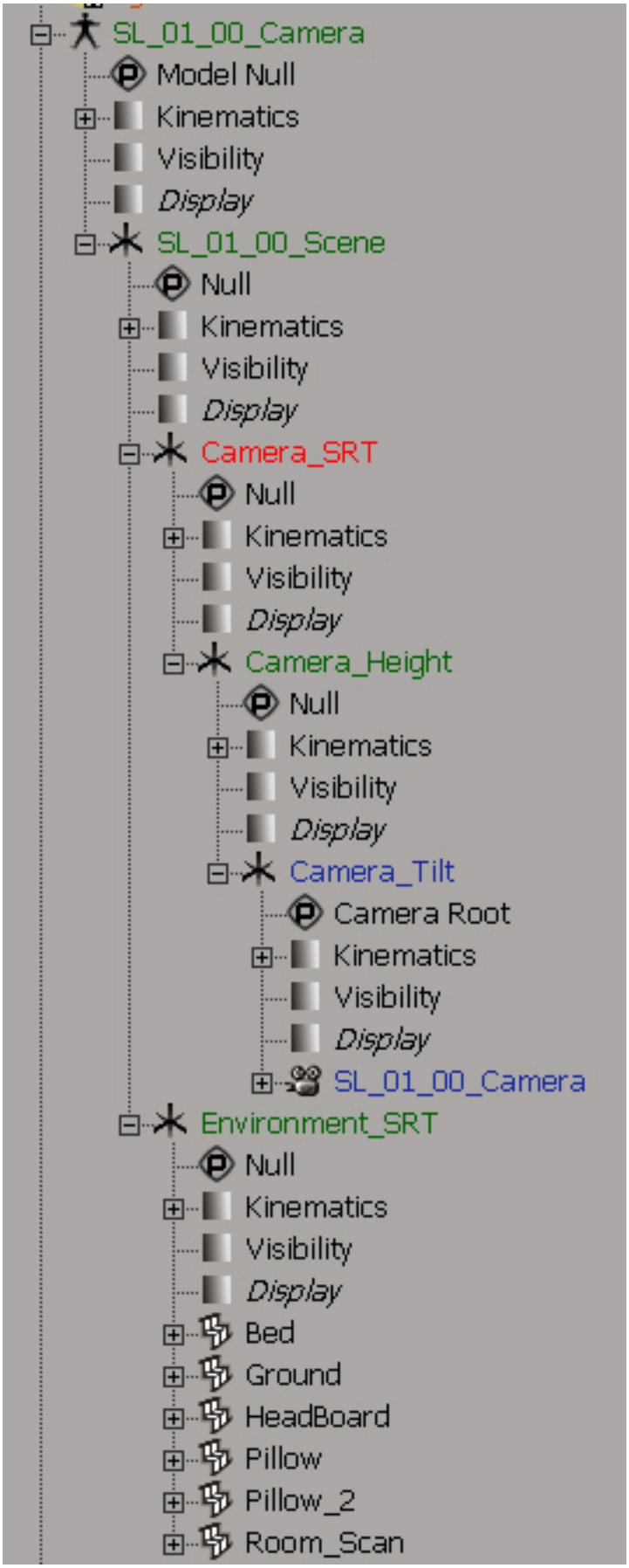
Scene CleanUp:

*At this stage your camera, environment modeling and you Animation BackGround should be matching. If this is not the case see Camera Setup*

Create a model and name it [Shot]_Camera
Make sure all the environment transforms have been freezed and place it in the model. (If there is various mesh, place them first under a Null named Environment SRT. Make sure nothing has transforms)
Place the Camera_SRT in the Model too.




Make sure the Model doesn't have any transforms. If it does use child compensation to remove them.
Create a new Null named [Shot]_SceneScale and parent Camera_SRT and Environment_SRT to it.
This will be the null used to change the scale of the meerkat. See Character Scale.
You should end up with this hierarchy:




By default [Shot]_SceneScale, environment_SRT and all the modelling shouldn't have any transformation.

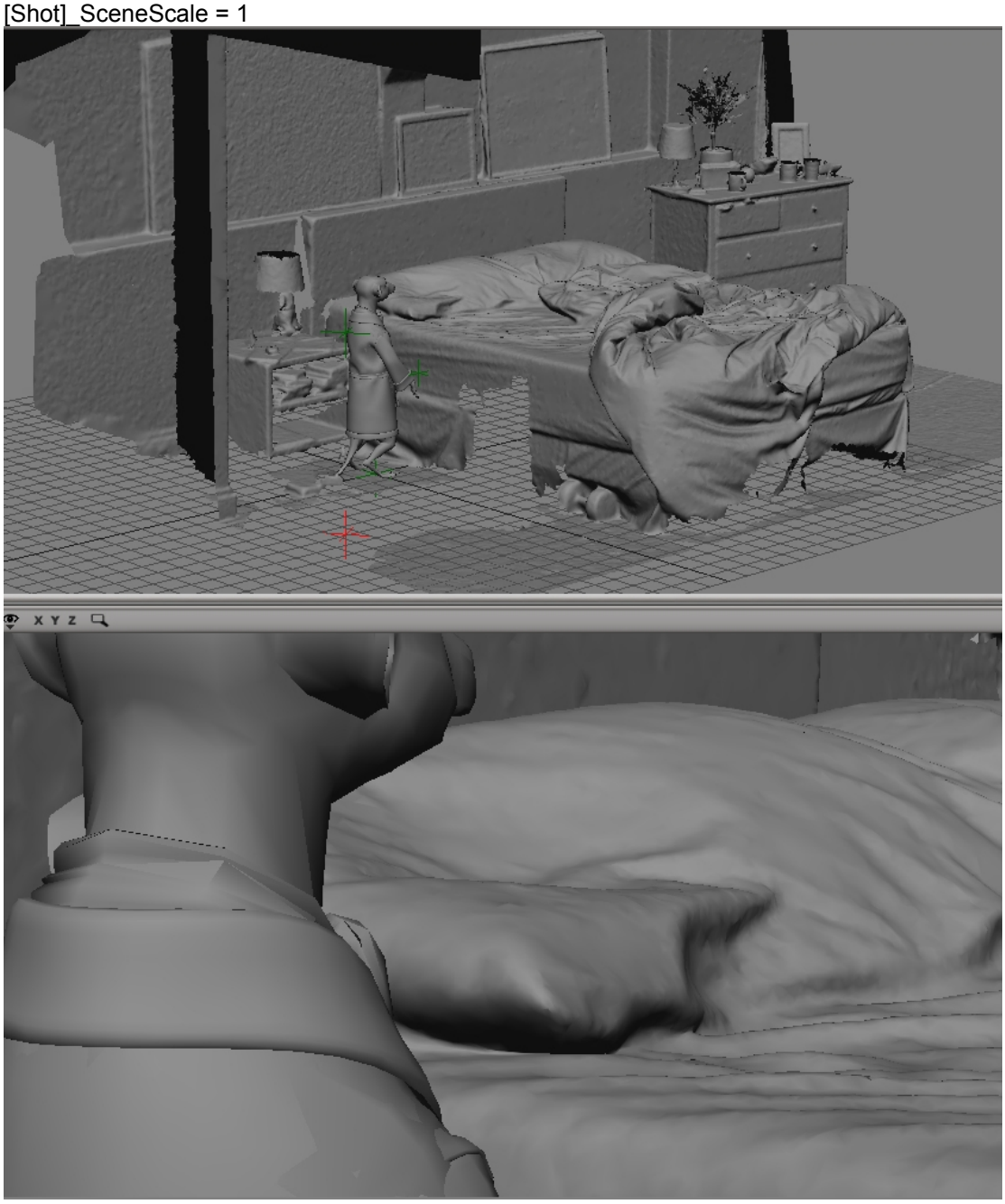
Character Scale:

*At  this point you should have a camera model with the standard hierarchy, if this is not the case, see Scene CleanUp*

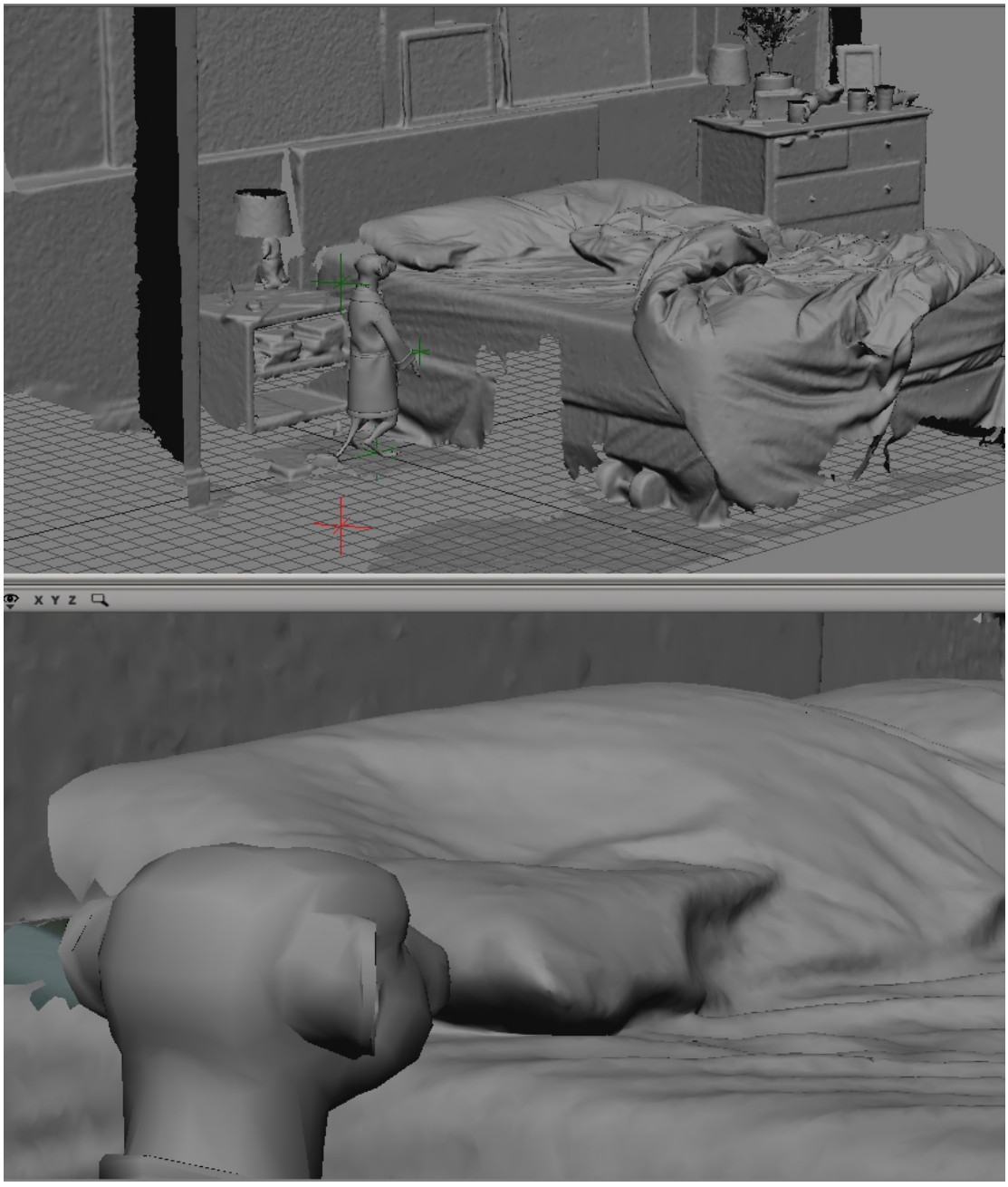
It is extremely import to NOT SCALE the animation characters.
We export point cash, witch transfer point position but does not affect scale, scaling the Character SRT would break the shading and the Fur.

To visually change the character scale, we will scale the set and the camera at the same time by scaling the [Shot]_SceneScale Null:

[Shot]_SceneScale = 1




[Shot]_SceneScale = 1.2 Notice how the meerkat is the same size on the grid, but smaller in the camera. The bed on the other hand is still in the exact same place in camera view.




The Scene Scale can be defined in the Environment scene (Using the data from the set  i.e. if Meerkat on shoot was 65 cm  CG meerkat is 85 cm so the scene need to be scaled by 1.31)

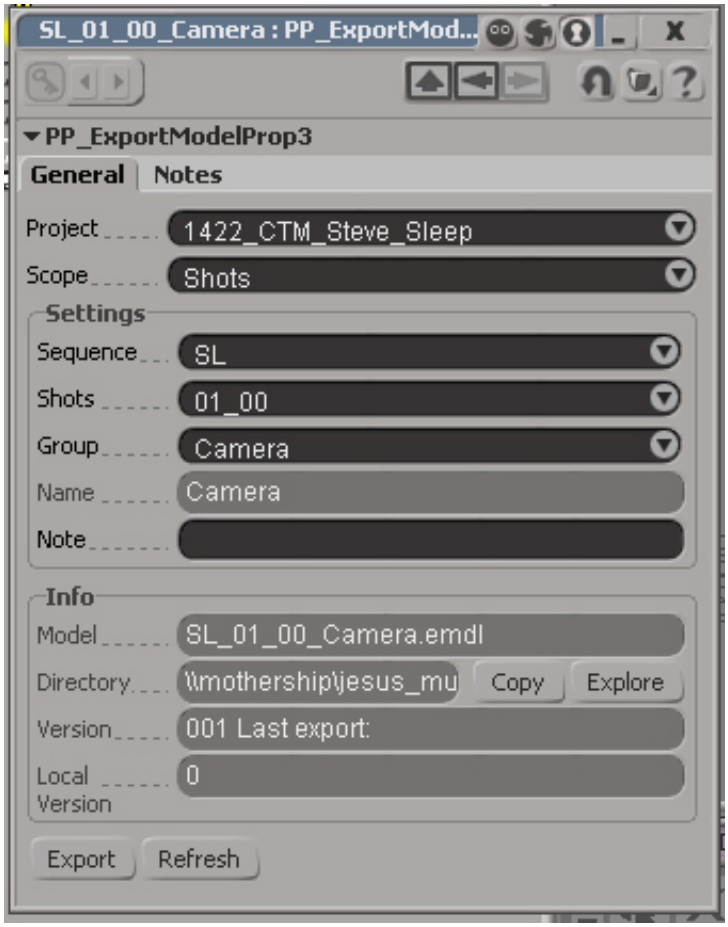
It can also be changed in the animation scene.

Unfortunatly some limitation in XSI creates wrong renders when Cameras are scaled.

IT IS VERRY IMPORTANT TO REMOVE ANY SCALE ON CAMERAS AND CAMERAS PARENTS.

This will be done in the render scene. Please see Camera & Environment

You can now export the Camera model using the PP_Export_Model.
Make sure that the latest camera exported always matches the scale set in the Animation scene. A quick look at the animatic or animation preview will make it obvious.



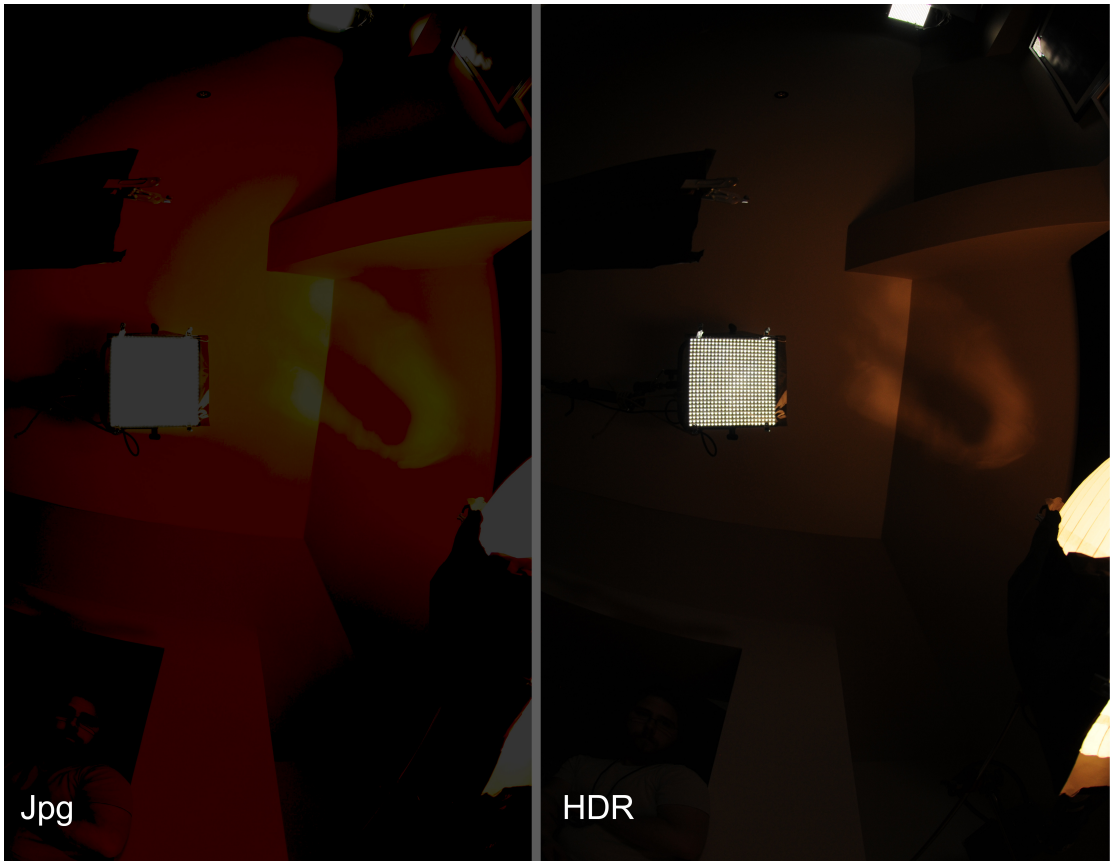

**HDR**
When referring to HDR we actually mean HDR Environment Map.

We usually use spherical maps:

HDR will be used to be reflected the XSI scene, setup the lighting and secondary bounces (Final Gather)
It is important that the HDR matches the Shot setup. For every new lighting setup you will need a new HDR.

Gather all the panorama pictures taken on the shoot (They should be in p:\[Project]\Pictures\ [Project]_shoot\ ) and locate the Mirror Ball shot in camera.

Hdr means Hight Dinamic Range, it is a picture that contains more info then a regular picture, notice the difference on the highlight when you change the exposure on a regular jpg and an HDR:



The JPG doesn’t have any bright light anymore, whereas the HDR is still full of details.

***PT Gui***

You can follow this easy tutorial:
http://www.ptgui.com/hdrtutorial.html

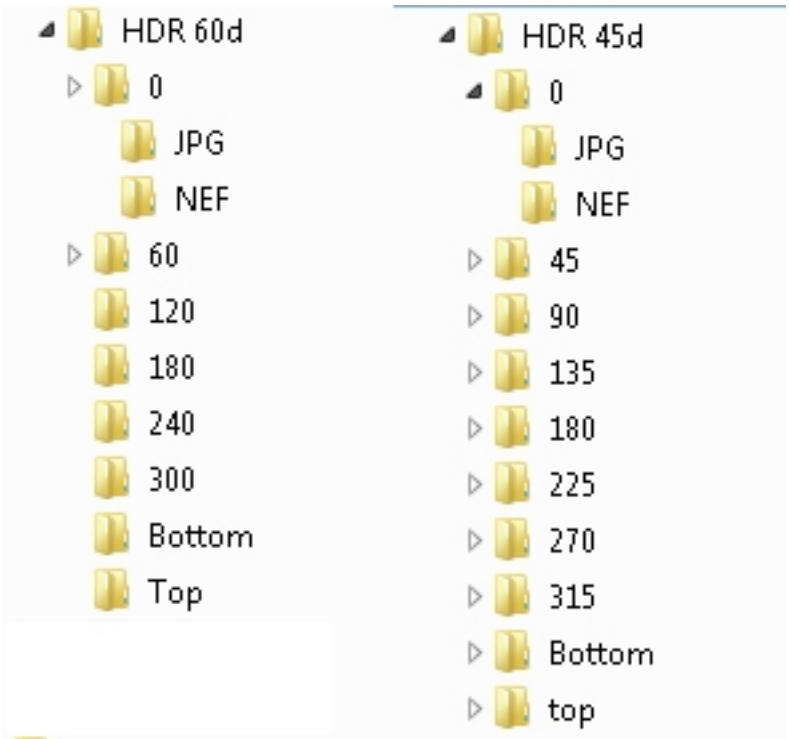
Export your HDR to the shot folder:
P:\Shots\[Shot]\EnvMap
as [Shot]_RAW_HDR.exr
You can open the HDR in photoshop to clean it and remove the tripod. Be careful to keep it in 32Bit!

***Trouble Shoot:***

If the PT Gui doesn't manage to automatically create an HDR you can follow those steps:

Merge to HDR:

Create a folder per photo angle containing all the different exposures. (If you have the raw file and the Jpg separate them into different folders):

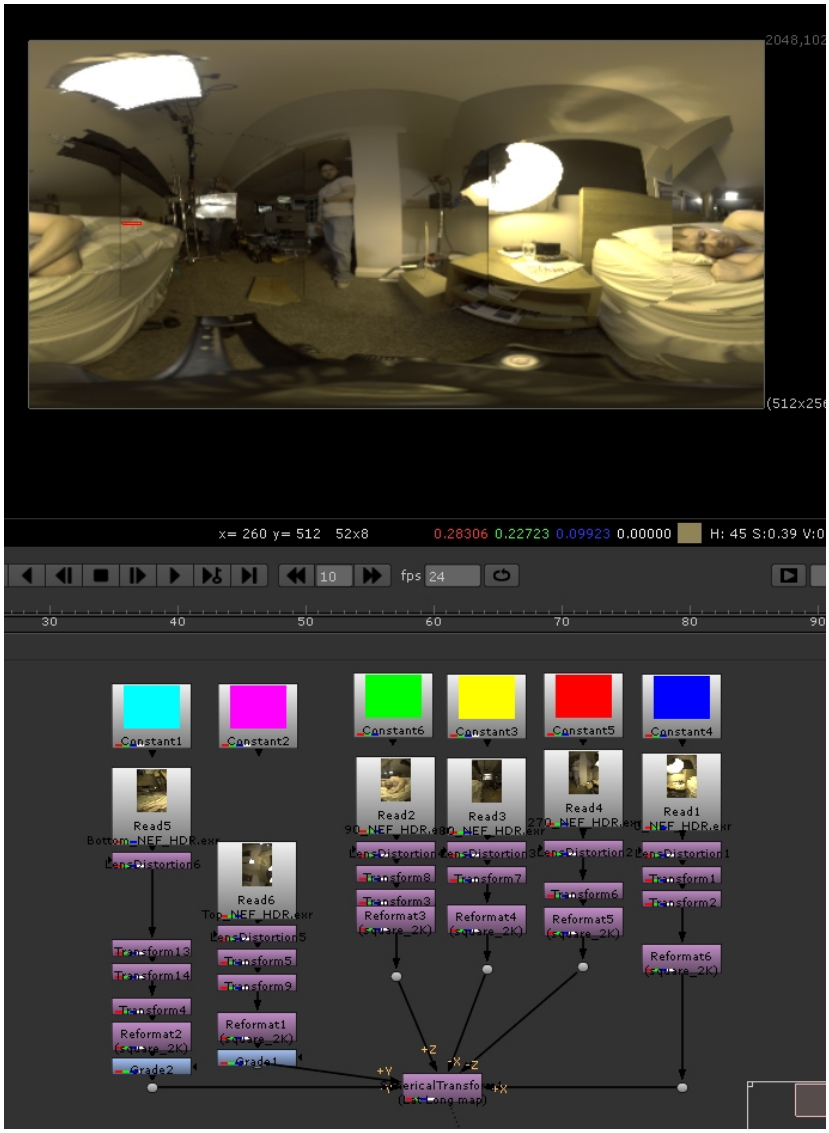



On the shoot the panorama should be taken every 45 or 60 degrees. Create the folders accordingly.
You can copy past the file structure from here: P:\PP_Library\Pictures\HDR File Structure

Open photoshop and go to File>Automate>Merge to HDR Pro
Select the first batch of photos (by using the rawfile you will have a greater range, but color might look a bit wrong at first.) If the pictures were shot properly you can untick "attempt to align".
When the HDR appears, switch to 32 bit. Press ok.
Save the HDR at the folder root as a EXR. As [Angle]_[Format]_HDR.exr i.e. 90_NEF_HDR.exr
Repeat for all the other angles.

Stitch in Nuke

Load all your HDR in Nuke and using the "spherical transform" merge the picture together to get an accurate LongLat map. Set the imput to cube and the output to Lat Long Map.
You will need to pre-merge and crop the hdr together to get 6 square images to input in each axis and negative axis:




Export your HDR (as 32bit) to the shot folder:
P:\Shots\[Shot]\EnvMap
as [Shot]_RAW_HDR.exr
You can open the HDR in photoshop to clean it and remove the tripod. Be careful to keep it in 32Bit!

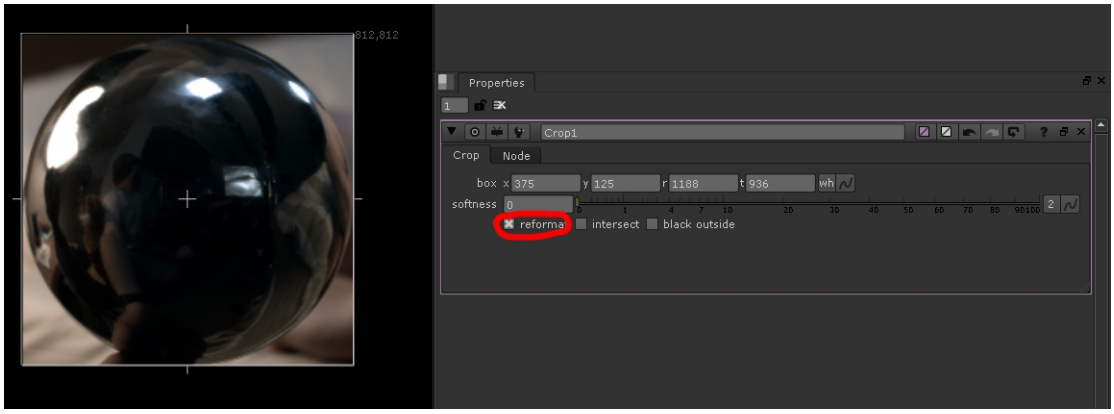
***HDR Conforming***

*You should have a Lat Long spherical HDR of your environment, (if you don't see HDR)* you need to conform it to what was shot by the camera, and export different versions of the map.

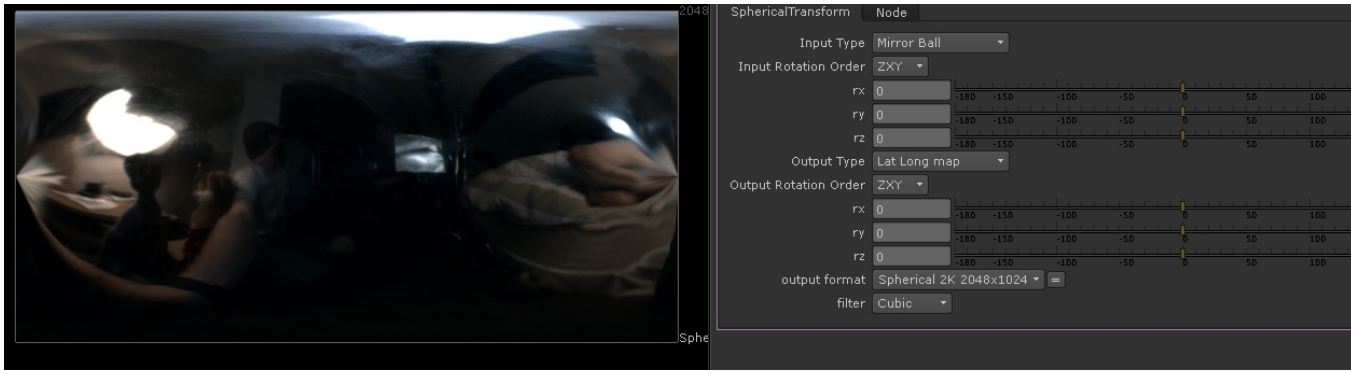
You need to locate the mirror ball shot for this specific setup. See Nuke Setup.

You first need to transform the Mirror Ball into a lat long map.

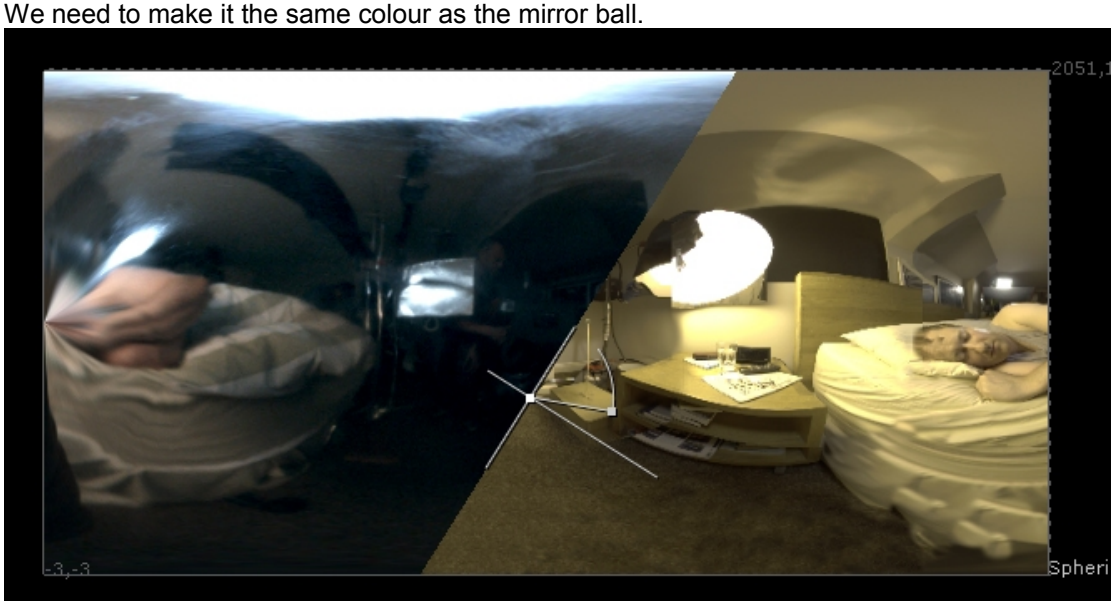
Load the Mirror ball footage in nuke, crop the footage so the mirror ball touches the borders of the picture. Tick "Reformat".




Connect it to a “Spherical Transform” with input set to “mirror ball” and output set to lat long map. Change the Output format to “Spherical 2k” or 2048x1024 if “spherical 2k” isn’t available:




Load the Raw_HDR created before and compare it to the unfold Mirror Ball.
We need to make it the same colour as the mirror ball.



You can roughly align them using “spherical transform” input= LatLongMap, output= LatLongMap output format = spherical 2k. Play around with the ry until the centre matches the mirror ball, you then need to mirror the mirror ball long lat so the object are in the same position. Don’t spend to much time doing this as it is just to help grading the RAW_HDR. It will be properly tuned in XSI.
Colour correct the RAW_HDR until it is as close as possible to the Mirror Ball. Be careful to keep the High Range in the whites (you can check it using the exposure tool)
You can grade it slightly lighter as the Mirror Ball is not 100% reflective.

You can save your comp into P:\Shots\[Shot]\EnvMap
As [Shot]_EnvMap_Comp.nk

***HDR Export***

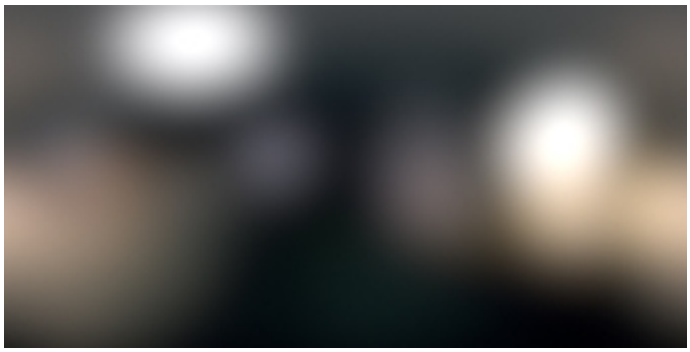
You will now need to export two environment map: One for reflection and one for Final Gather:

Reflection Map:

Export the RAW_HDR rotated and graded as a 32bit EXR 2048x1024 named:
[Shot]_RefMap.exr
in the following location:
P:\Shots\[Shot]\EnvMap

FinalGather Map:

Reformat the rotated an grade RAW_HDR to 256x128 or “Spherical 256” if available.
Clamp it to 0 – 2 (maximum value should be 2).
Apply a gaussian blur of 25.
Export it as a 32bit EXR 256x128 named:
[Shot]_FGMap.exr
in the following location:
P:\Shots\[Shot]\EnvMap



**NAMING CONVENTION**

***Textures***

Size can be 256, 512, 1k, 2k or 4k.
By default textures should be saved as 1k and uprezed or downrezed on request.

- Diffuse Colour: [Asset]_ [size]_COL
- Bump Map: [Asset]_ [size]_BMP
- Reflection / Spec Map: [Asset]_ [size]_REF
- Displacement Map: [Asset]_ [size]_DISP

***Modeling***

***Passes***

Passes are separated in two categories: Beauty and Utility.
Beauty is everything that will be seen in the final frame. Utility is everything that will be used to improve the beauties.
Beauty passes are named after what is in the beauty: Beauty_[Asset]:
Examples and main appellations:

- Beauty_Main would have most of what will be rendered in the scene.
- Beauty_Char would only have the Characters
- Beauty_Hair would only have the Hair
- Beauty_Env would only have the environment

Utility passes are named after the type of utility and what's in the passe:
Utility_[Asset]_[UtilityType]
Examples and main appelations:
Utility_Char_AmbOcc
Utility_Env_Shadow
Utility_Char_Mates
Utility_ID_Mates would be for unspecific mates

***Materials***

You do not need to create a library per model, we're using Reference models, and Xsi creates a library per reference model named: [Model]_[Library]_ DefaultLib_Ref(ref) if you create a new library with the name of the model, we would then have the name twice.
Keep all you materials in the DefaultLib.
Material name should be: [Mesh]_Mat
We try to separate mesh per materials.
If the material is shared across multiple meshes name it by its type:
[Type]_Mat i.e Gold_Mat

Each render scene (made from the template scene) contains a Utility_MatLib that contains the most used utility shaders and helps having materials once:
Cst_[Colour] is a constant shader used for mates, its radiance should be at 0.
Cst_[Colour]_No_Alpha is a constant shader used for mates, its radiance should be at 0 and it won't appear in the Apha.

## SHADING

Each model should have its proper shading scene, named “Rendering”.
P:\[Project]\Assets\[Type]\[Asset]\Scenes\Rendering

Every Render models are exported from there rendering scene.

It should contain a neutral lighting similar too the template scene:
Neutral lighting is achieved by using a Black and white HDR, and a white lighting. The light Setup should match the one that will be used in production, i.e. exponential attenuation on/off, area light or regular lights etc.
You can create a second pass that contains a more artistic lighting, but the goal of the shading scene is to make sure that object react properly in every setup.

The scale of the object should be at 1 and should match the one in the animation scene. This way the point cash would not change the relative scale. (This is important because cashes don’t scale normales, which affects shading dramatically, i.e bump, sss etc).

The model shouldn’t be affected animated or have any transforms as this would be transferred in every scene.

This scene should be kept very tidy as it will be shared and used across freelancers. Make sure to respect the naming convention. Clean the material names and render trees. Be careful when importing new assets in the scene and make sure to remove unused source clips.

## RENDERING

### *Template Scene*

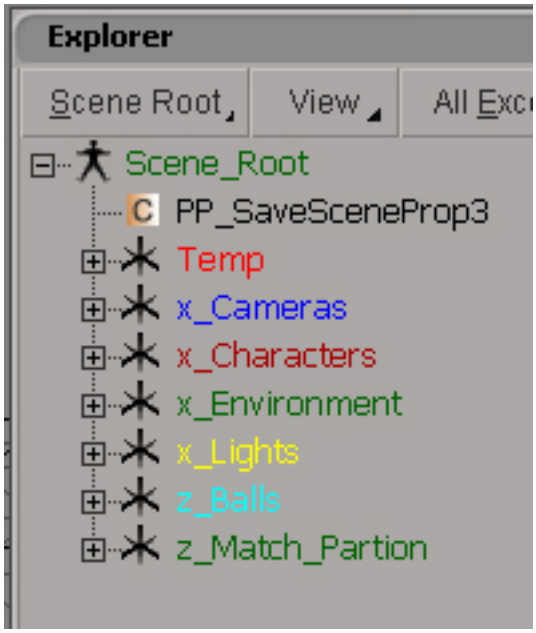
The Generic template scene is a scene where most of the passes, render settings and partitions are ready. The main characters are already set up and a standard lighting and environment is ready.
In order to be ready for the first render you will need to:

- Import the right Camera and environment and set it properly, as explained Camera & Set.
- load the animation and/or import the missing characters.
- set the right environment and lighting.

It is preferable to Create a New Template Scene from the Generic one, per project.
The template scene are located here:
P:\[Project]\Asset\Template\Generic
Or
P:\[Project]\Asset\Template\[SEQ]

Hierarchy

The scene hierarchy is made so you can easily find different element and also easily use the “Match Partition” Script:




Temp

contains Stand-In Meshes that you can use to set the first scene or the Project based Template (see Create a New Template Scene). Once you matched the Characters to the Stand-Ins they can then be deleted.

X_Cameras

Contains the cameras of the scene (without the Environment see Camera and Set.)

X_Characters

Contains all the Characters Referenced models.

X_Environment

Contains the environment meshes (see Camera and Set.)

X_Lights

Contains all the lights of the scene (see Lighting)

Z_Balls

Contains a GreyBall and a Mirror Ball and any other reference that might be useful. They will be use to set up the lighting. (see Lighting)

Z_Match_Partition

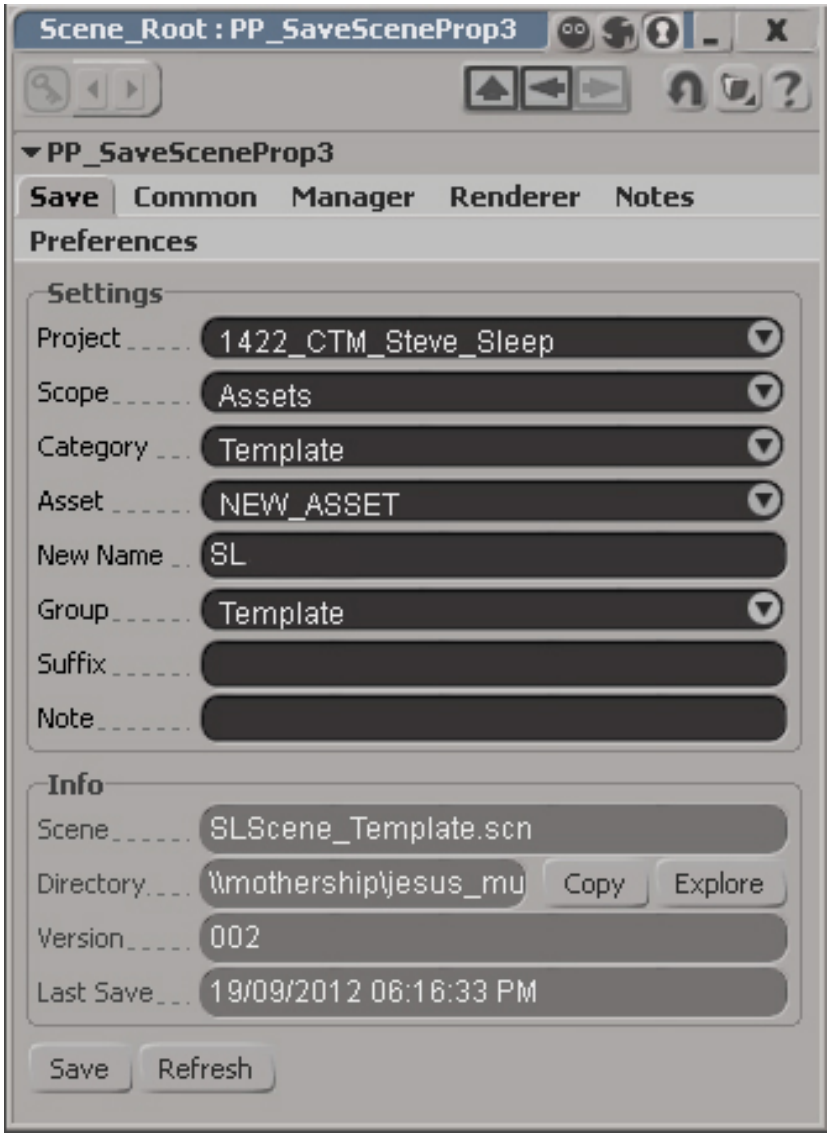
Contains implicit cubes (non renderables) that are placed in the right partition. By running the “match Partition” script, newly imported mesh will be assign to the right partition, see Match Partition

Creating a New Template Scene

At the start of a project it is better to create a new template scene from the Generic Template Scene, to avoid loading the same models and setting up mattes etc all over again.
If there is an over all consistency between shots it is worth lighting the template scene with the main lighting of the project.
If not, the template scene can be quickly set and released to the different lighting artists.

To create the New template Scene:
Open the Generic template scene. It should have been copied in P:\[Project]\Asset\Template\Generic (If not contact your coordinator)

Save it as Template>[SEQ]:




Import as Reference models, theRendering Models  of the characters that will be featured in the spot.
Parent them to x_Characters.
Select the character mesh and Match Partitions to the right implicit cube: i.e. select all the Aleks Meshes and match it to z_MP_Aleks.
Do the same with the characters Hair: i.e. select all Aleks Hair and match it to z_MP_Hair

Go to the “Utility_ID_Mates” Pass (See Passes for more info) and assign the character meshes as follow:
BLUE:
Eyes – Nails – Everything that goes on top of Cloth (i.e. tie, cravat, belt)
GREEN:
Body
RED:
Cloths – Gums – Teeth - Tongue

If you don’t need to light the [Project]_Template scene then that’s it.

If you do need to light it then follow Camera & Set and Lighting and then save the template scene.

Passes

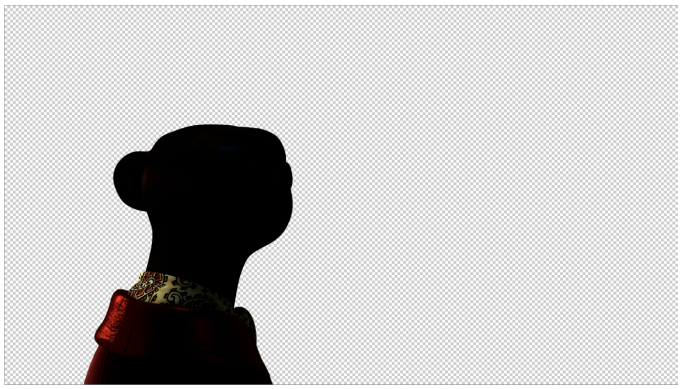
Each scene should have a Default Pass, named z_Default_Pass, that shouldn't be changed from the XSI default one. This partition will be used to debug the scene and shouldn't be rendered.
Each passes contains a Background_Objects and a Background_Light partition that should be Hidden. This is so that new lights or new meshes don't affect passes previously created.

The template scene contains 6 passes. Separated in two types:

Beauty

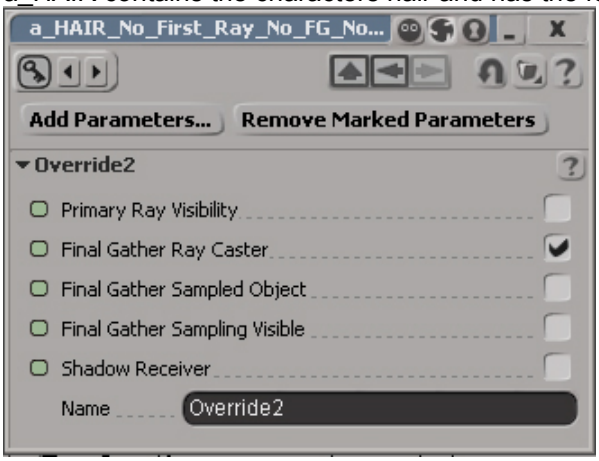
Beauty Passes are every passes that will be seen in the final image.
The template scene should have Characters, hair and Environments.
We are separating Character and Hair to be able to light the characters with area light as Hair can't be lit by area lights.

*Beauty_Char:*



Contains 5 partitions.

- a_GEO contains the characters geometries and has no override.
- a_HAIR contains the characters hair and has the following overrides:




This partition can be hidden if the scene is lit with area light and the render time is too high for not many changes. See Character Lighting.

- b_ENV contains the environment meshes and has a "No Primary Ray" override.
- z_Balls Contains a grey ball and a mirror ball and should be hidden by default.
- Area_ON contains all the light that should be ON in the scene and has no override. See Character Lighting.

*Beauty_Envirtonment:*

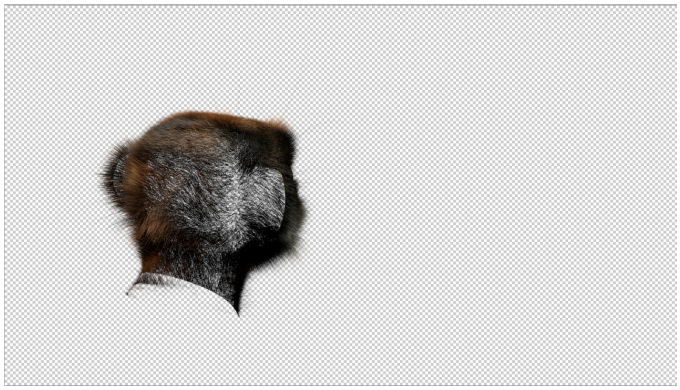
This pass should be rendered only if a CG environment is used.

It contains 4 partitions:

- a_ENV contains the environment geometry and has no override
- b_GEO contains the characters geometry and has a “No Primary Ray” override.
- z_Balls Contains a grey ball and a mirror ball and should be hidden by default.
- Area_ON contains all the light that should be ON in the scene and has no override.

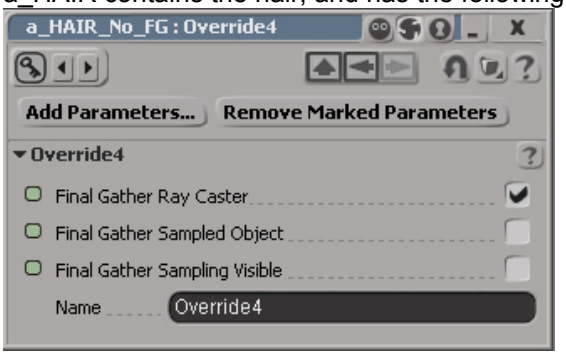
Note that hair is in the background partition.

*Beauty_Hair:*



This pass renders only the hair, character geos are use to mate the hair. The shadows are changed to Shadow Maps. See Hair Lighting.
It contains 5 passes

- a_HAIR contains the hair, and has the following override:




This is so the hair doen’t cast final gather but receive it.

- B_GEO contains the characters geometries and has a material override : Cst_BLACK_No_Alpha from the Utility_MatLib. This is so the hair is not rendered where the geo is.
- b_ENV contains the environment meshes and has a “No Primary Ray” override.
- z_Balls Contains a grey ball and a mirror ball and should be hidden by default.
- ON_ShadowMap_Override contains the hair lighting (see Hair Lighting) it has “Shadow Map On”, ”Area light Off” override. (Also “Specular ON” optional)

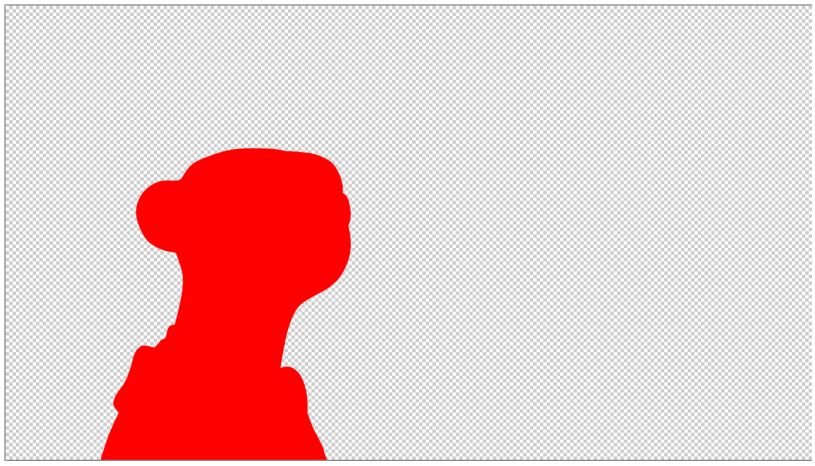
Utility

Utility passes are every passes that will be used to improve the beauty in comp or to help integration.
There goal is NOT to rebuild a beauty in comp, it’s to fine tune renders.
You should always go back to the beauty for lighting and shading rather then try to fix it in com, this is to keep renders energy conservatory and consistent.
Utility passes should not be rendered with lens shader on.

*Utility_AmbOcc:* 

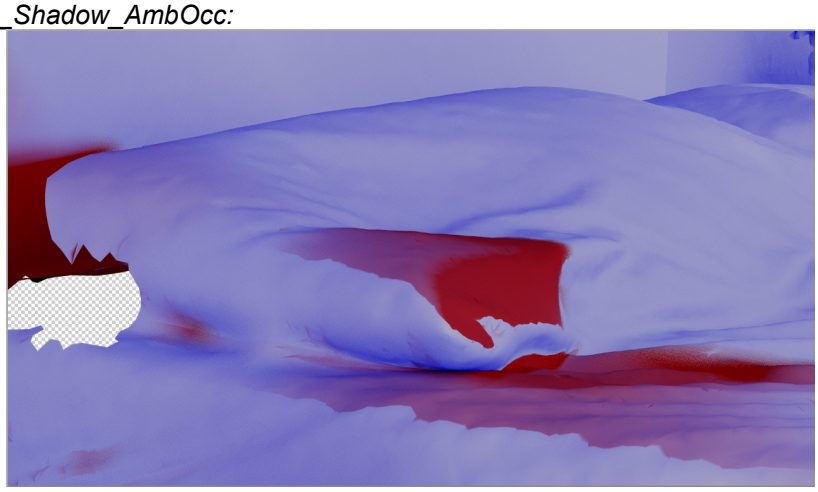
This is the character ambient occlusion.
Partitons:

- a_GEO_AmbOcc contains the char geo with AmbOcc material override. (You can override it with a RED ambocc if you want to separate it from hair)
- a_HAIR_FakeAmbocc contains the char hair and has a Hair_Fake_AmbOcc material ( a black and white gradient from the root of the hair to its tip  you can create a blue gradient if you want to separated it from the char geo)
- b_Env_No_1st_Ray contains the environment geo with a “primary ray visibility OFF“ and a lambert material override. (this is to have a simpler shader to compute since colour is not relevant.)

*Utility_Char_Mates:* 

This pass should only be rendered when there is more then one character or when there are props.
RED = Aleks
BLUE = Sergei
GREEN = Props

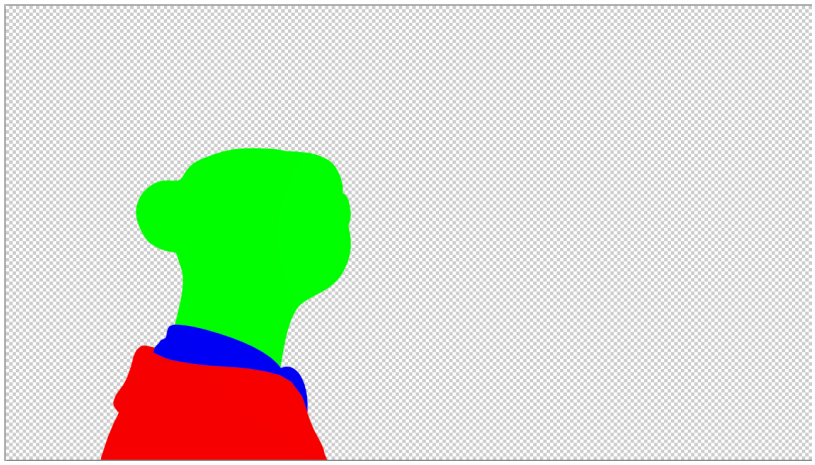
*Utility_Env_Shadow_AmbOcc:*



This pass is the ambient occlusion and shadows of characters on the environment.
Red channel = ambient occlusion
Blue channel = Shadows
Green channel = both
Partitions:

- a_ENV contains the environment meshes and has a “Shadow Caster OFF” “secondary Ray OFF” “Reflection visible OFF” so it wouldn’t cast shadows or ambient occlusion on it-self
- b_Char contains all the characters Geo and has a lambert material and a “Primary Ray OFF” override.

Utility_ID_Mates:



*This pass is to separate different part of characters and props. They should be used in combination with the Utility_Char_Mates.*

BLUE: Eyes – Nails – Everything that goes on top of Cloth (i.e. tie, cravat, belt)
GREEN: Body
RED: Cloths – Gums – Teeth - Tongue

***Match partitions***

Match partition is a script that assigns any selected object to the partition of another object.
To use it:
Select what needs to go in different partitions (Mesh, Hair, Model) go to the rendering tab > Partition>RT_MatchPartitions when ask pick the object that the objects should be matched to.
There is 3 different scripts:

- rt_MatchPartitions simply matches anything selected to a single object’s partition.
- rt_MatchModelPartition matches the partition of the mesh with the same names in a model. Select a model, run the script, when asked pick another model, mesh with the same name will go to the same partitions.

- rt_sendToBackGroundPartition the selected object will be assign to the background partition in all the passes. This is useful for SFX mesh that should never be rendered.

***Camera & Environment***

*At this point you should have a Camera model containing the camera and the environment with the correct hierarchy, in the right position and at the right scale. If you don't see CAMERA & SET*

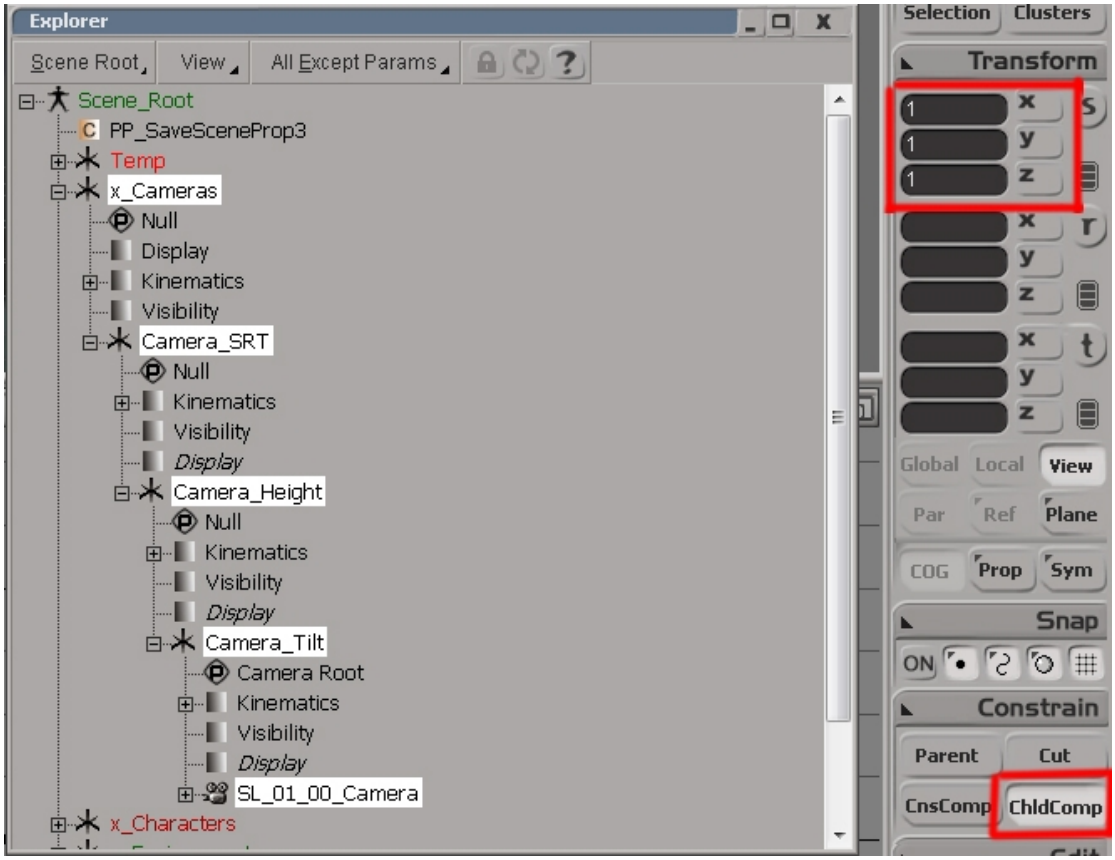
Open the Project Template scene: [Project]_Template (created here)
Save it as the shot you are going to work on.
It is important to do that first so the template is not overwritten.
Import the Camera Model (locally)
Import Animation and check that the camera and characters match the Animation edit.
Delete the "Camera_To_Be_Deleted" model if it exists.
Parent "Camera_SRT" to x_Cameras and "Environment_SRT" to x_environments.
It is very important to remove any scale on the camera parents or on the camera it self. We are using "environment Blur" witch doesn't work with scaled cameras:


Mirror Ball reflecting a NoPic Environment. Notice how with a Scaled camera (scale here is on the Camera_SRT null) the environment projection is wrong.

To remove all scales on the camera:

Static Camera

Static cameras are cameras WITHOUT any animation nor keyframes.
Select the camera and all of its parents (Don't branch select) and enter 1,1,1 in the Scale transform with chdComp ON :




Animated Camera

If a camera as a single KeyFrame follow the Static Camera process and then change the scale keyframe to 1,1,1
If the camera has a proper movementanimation, duplicate the camera, rename the old camera [Shot]_Camera_Old and the new camera, [Shot]_Camera.
Parent the new camera to x_Cameras.
Remove all transform animation and all constrains from the new camera. Be carful to not remove All animated parameters as the focal length could be animated.
Make the camera scale 1,1,1.
Select the camera and Pose constrains it to the old camera.
Plot all animation on the new camera.
Go to animation>remove animation>From Scaling.
Set the camera scale to 1,1,1
Scroll through the time line with both camera views on to check that they are identical.
Get ride of the old camera and all its hierarchy.

Camera setup:

You can load the Rendering BackGround as Rotoscop, start by the mirror balls.
Add a photographic Exposure shader on the camera.
If you have more then one camera, make sure that the right camera is assigned to every passes.

Environment Setup

Select environment meshes and Match partition to z_Match_Partition\ Z_MP_Environment
Get rid of Environment_StandIn if needed.
If your environment doesn't have a ground, it is preferable to create a simple grid with a black material, in order to stop final gather to coming from the ground.
Export a frame of the Background Plate as an exr in:
P:\[Project]\Shot\[Shot]\EnvMap
name it [Shot]_Projection.exr

assign a constant material to the environment meshes and plug in the projection map using a camera_map_mip node. Change the de-gamma to 2.2. Set the constant radiance to black (so it wouldn't receive FinalGather).
With this setup only what is in the camera FOV will be rendered on the mesh. The rest will be taken care by the environment map.
Do not assign this material to the ground.

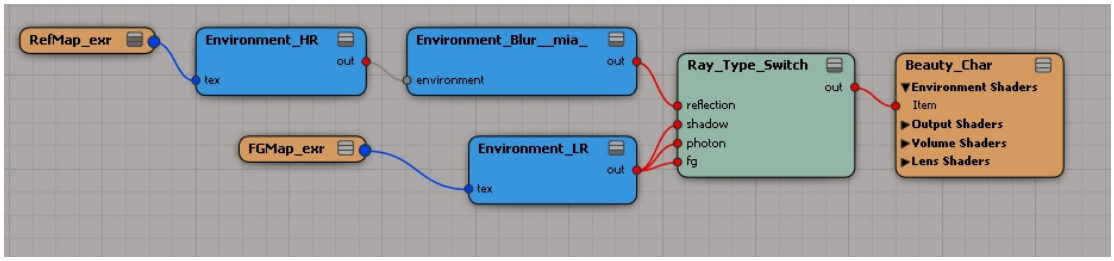
***Lighting***

At this point you should have the camera and environment set, the animation loaded, the Environment Maps ready.
For any partition or pass problems, read Passes.

Scene Environment:

The environment will take care of the reflections and the ambient lighting (Final Gather).
At no time your scene should have any colour in Render>Ambience.
Switch to Beauty_Char pass, open the environment render tree and switch the path of FGMap_Exr and RefMap_Exr to the shot specific environment maps.
Or open the explorer, go to image (o) in source locate FGMap_Exr and RefMap_Exr and switch the path.
By doing this, instead of simply importing the new environment maps, updates all the passes environment and not just the one your working on.

Environment RenderTree:

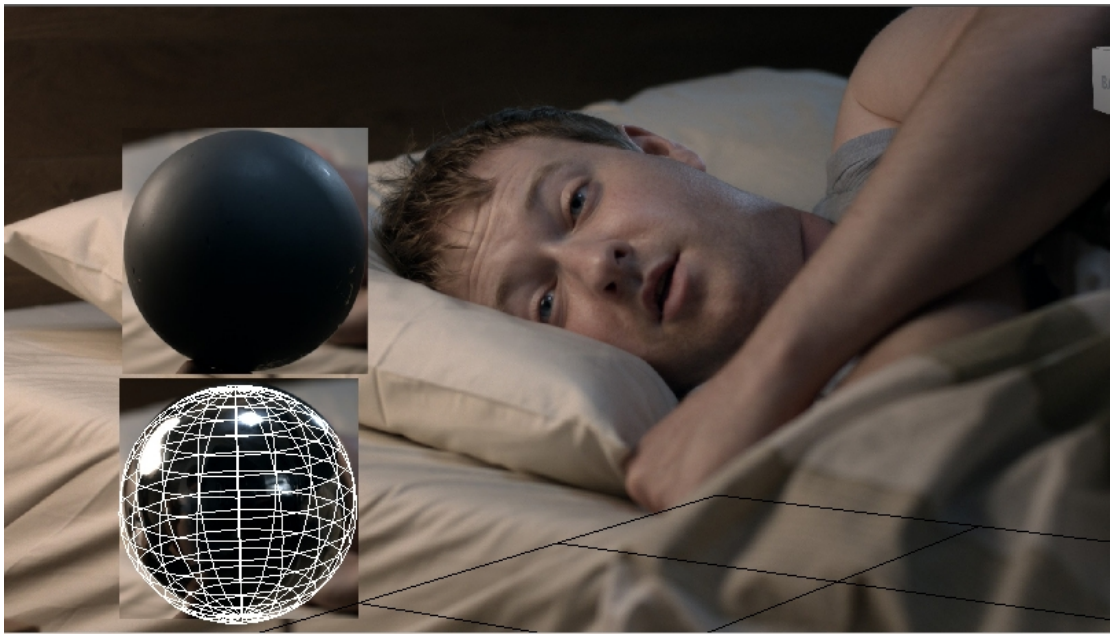



From right to left:

- Ray Switch allows to have different environment depending of the type of "Ray" i.e. eye, reflection, refraction, final gather. Eye and Refraction are set to BLACK, if you're dealing with a lot of refracted objects you will need to plug an environment in the "Refraction" slot.
- Environment blur mia makes glossy environment reflection extremely fast. It renders different environment blurred maps end use them as reflection where no geometry is reflected. Warning: If you have an almost sharp reflection but not 1 (i.e. glossy = 0.9) Mental ray will use a very slightly blurred environment but because the environment map resolution is set to 200 pixel per default, so the object will look to mush glossy. If the glossiness is set between .9 and 1 I would advice to keep it at 1.
- Environment HR and Environment LR are simple environment nodes. Make sure that the two always have the same transformation. They can have different image and reflection intensity as they won't affect each other.

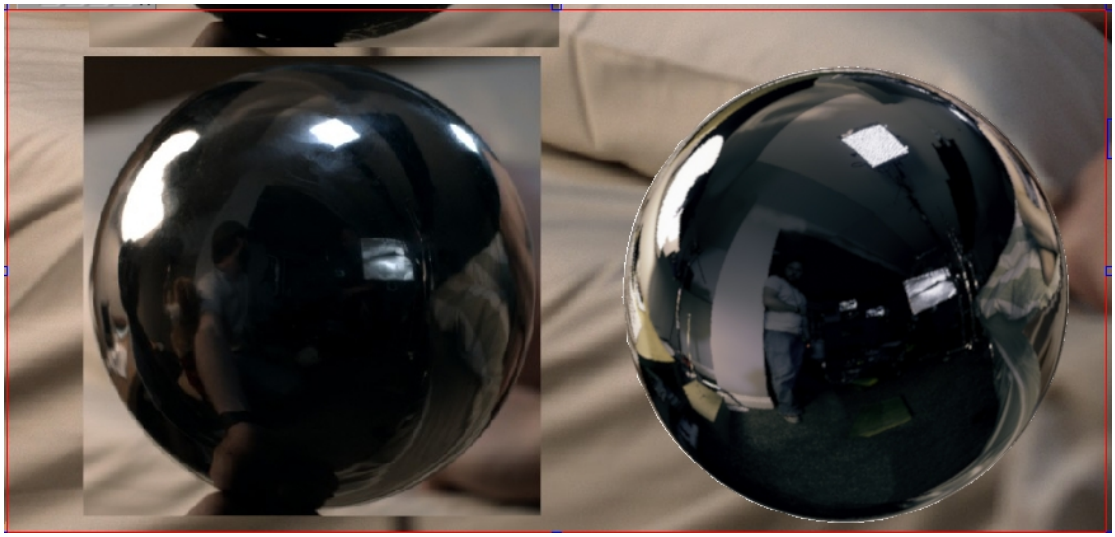
Environment Alignment:

You need to align the environment so the orientation matches the scene and the live action.
In the Beaty_Char Pass, hide "Area_ON", "a_Geo" and "a_HAIR_No_[…]_Shadows" partitions, and UnHide z_Balls
Roughly position the mirror ball where the characters are and where the mirror ball is on your Rendering BackGround:



Render region to see what the Mirror ball looks like.

You will need to rotate the environment_HR until the reflection matches the reference as much as possibl:



The CG Balls is brighter, this is because the mirror ball shot on the set wasn't perfectly reflective.
The most important is to have the high light in the same place as we are going to use them to position our lights.

For reference this are the transform I get.


It is preferable to mach axis per axis as the transforms are in world space, not camera space.
Do not change the Scale.

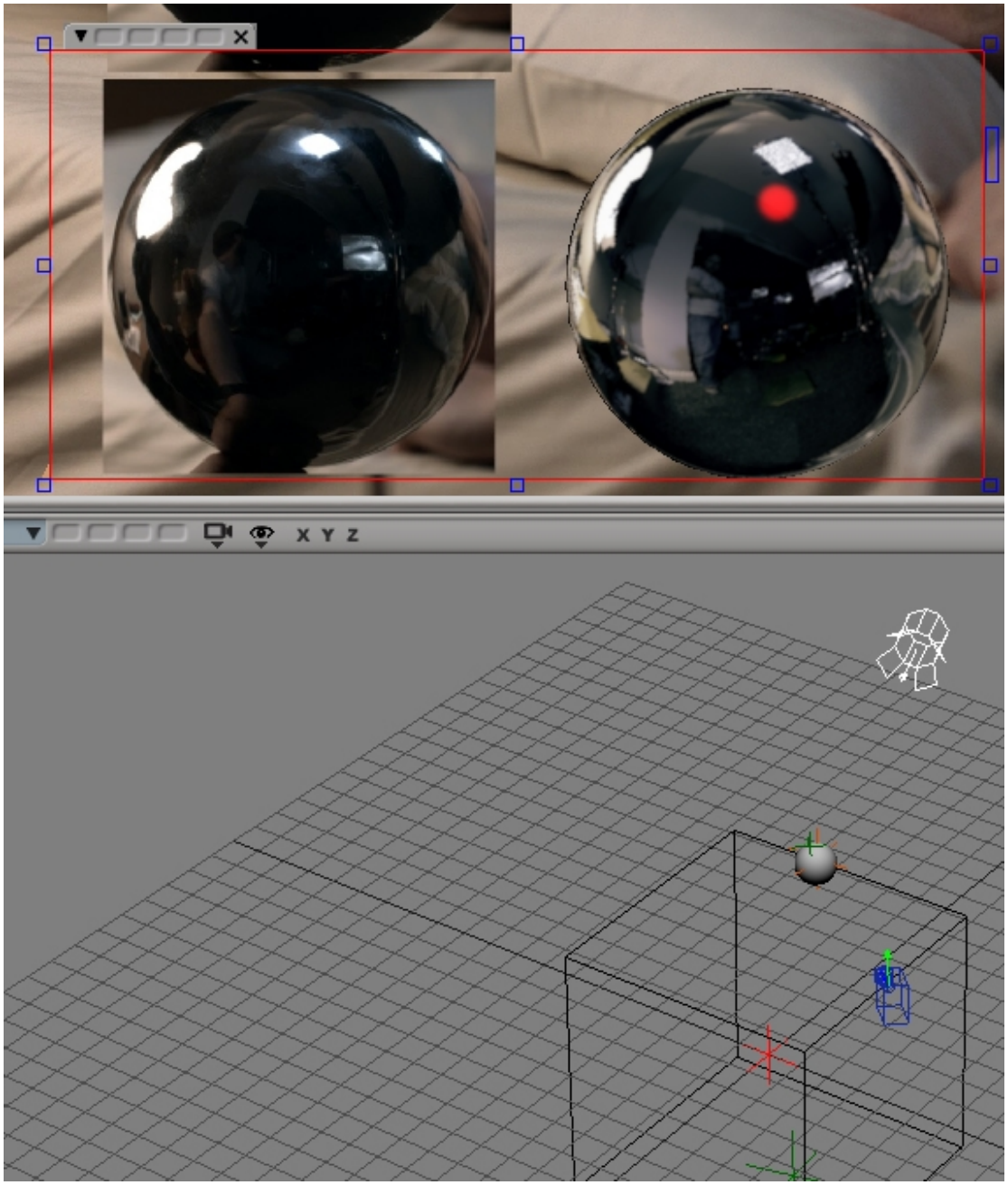
Once you're happy with the alignment copy the translation over to the environment_LR.

Scene Lighting:

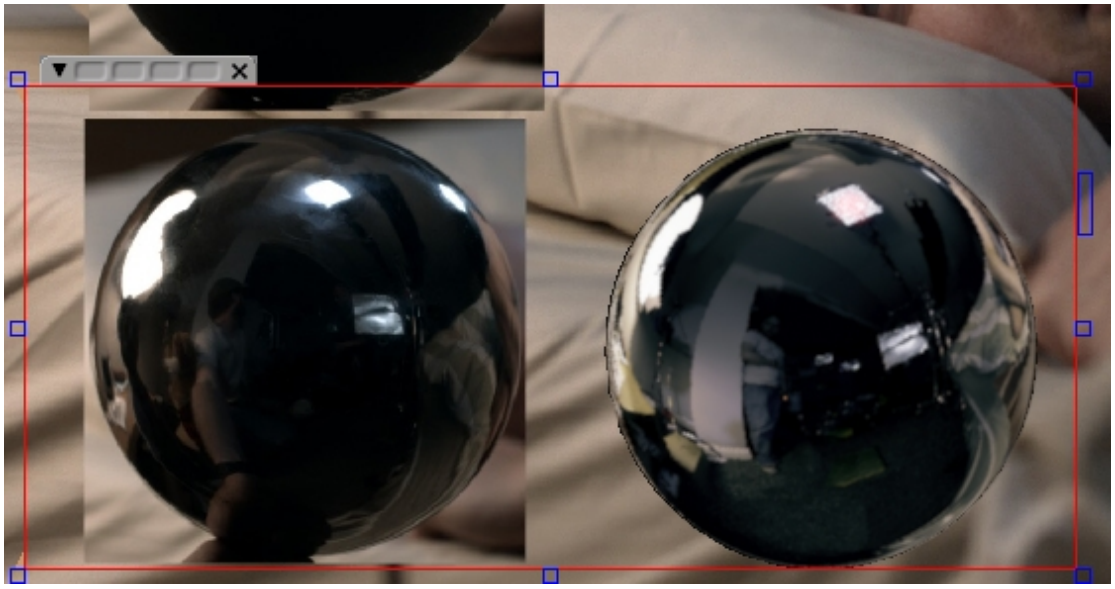
The scene lighting will take care of the direct lighting. Geometry can be lit with area lights but hair shouldn't. This will be taken care with the partitions.
It is extremely important that the lighting of all the passes comes from the same lights.
The Lighting should ALWAYS match the live action. This is the only way the integration will work.
The ONLY freedom you have Lighting wise when doing integration, is adding rim lights and feel lights. This is because on the shoot they would be character specific and so they would be turned off while shouting the plates. See Extra Lighting.

Light orientation

The first step is to place the lights around the scene.
UnHide Area_ON partition.
Create a new Spot (It is important to use spots when dealing with hair, see Hair Lighting)
Parent it to x_Lighting. Match partitions to the light in the Temp Null. Get rid of that light.
Reset all the light transform and place its Interest where the character is.
Set the light color to an obvious color (i.e. Red) and roughly place it around where the keylight will be. (On set photo reference might be available, it's usually wide shots of the setup where you can see all the lights.
First Render Region:




Move the light around until the specular matches the set light:



You can berly see the red spec anymore.

Do the same for all the other lights: Duplicate the spot you have created. (Duplicate the whole hierarchy to avoid having shared interest – this causes a lot of bugs)

Some element of the lighting might be in the scan. Use this kind of info to get a more accurate lighting.

Light orientation matched:

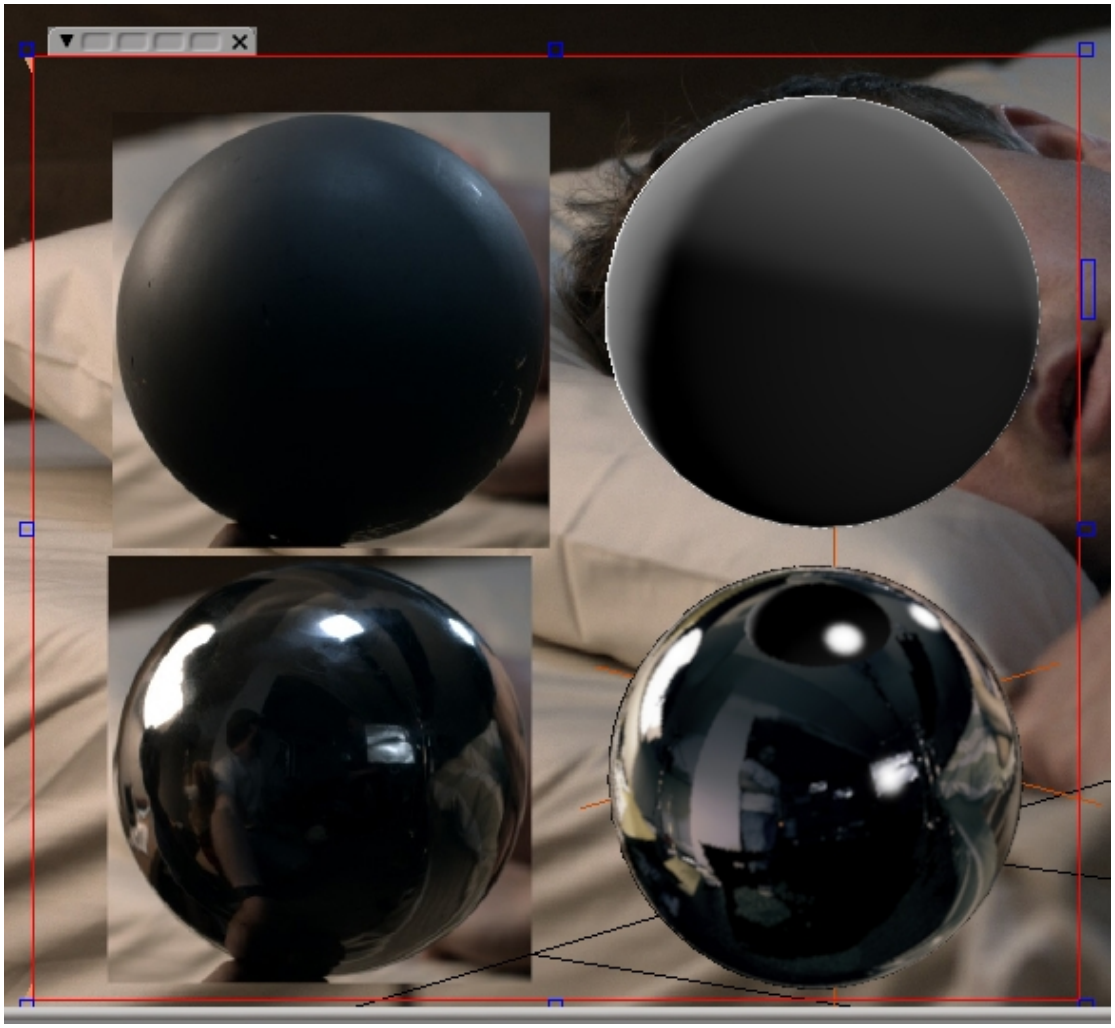
From left to right: Reference Mirror ball, CG Mirror ball, CG Mirror ball without environment.

Light Intensity, scale and position

Light intensity will depend on a lot of parameters.
Open the reflection map and measure the pixel values of the lights:


This will give you relative values. i.e. the Chinese lamp on the right is 14, the top light is 4, the the metal reflector is 1 and the tiny light in the back is 20. Do not take into account colour just yet.
Select all your lights and set their colour to white and their intensity to the one in the HDR.

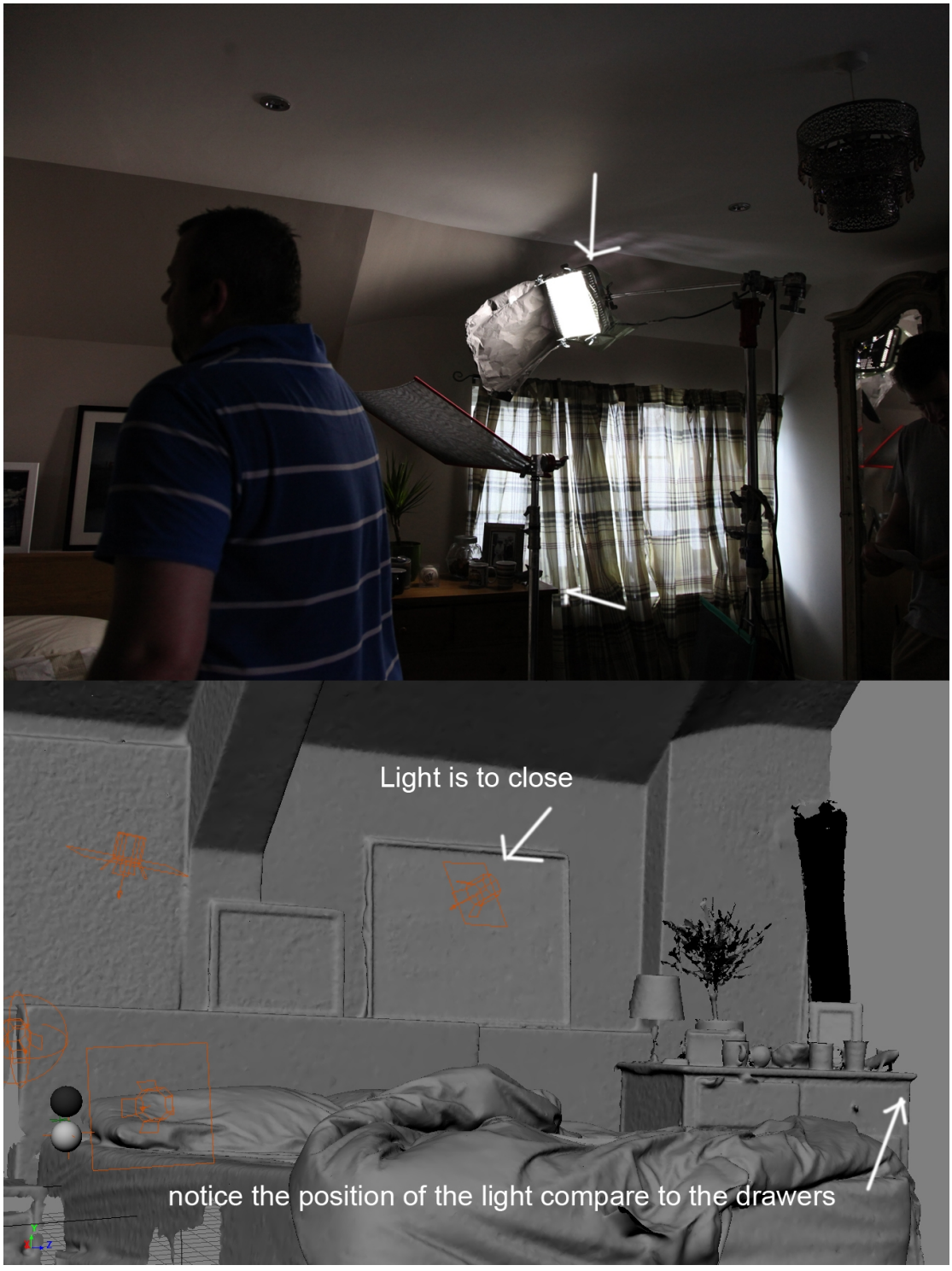
If you are rendering interiors it is much better to use Exponential attenuation. That way when characters move around the light interaction will look much more natural.
When turning on the attenuation, multiply your light intencity by a 10.
Activate the light shadows, and set the Umbra to 0.
Make sure Final gather is off in the render region.



The light position seams right, the light is to strong. But the biggest problem is the softness of the light.
By default in XSI the light intensity is linked to the Aera Light size.
Turn on area light and visible in render for all the light, hide the grey ball.
Match the size and rotation of the area lights to the lights on the set. You can fine tune their positions. (I switched back to a bright color)

Because of the exponential attenuation, Intensity also depends on the distance.
Using the shoot reference end the reflection map, move the lights to the right distance, using only the Z axis:

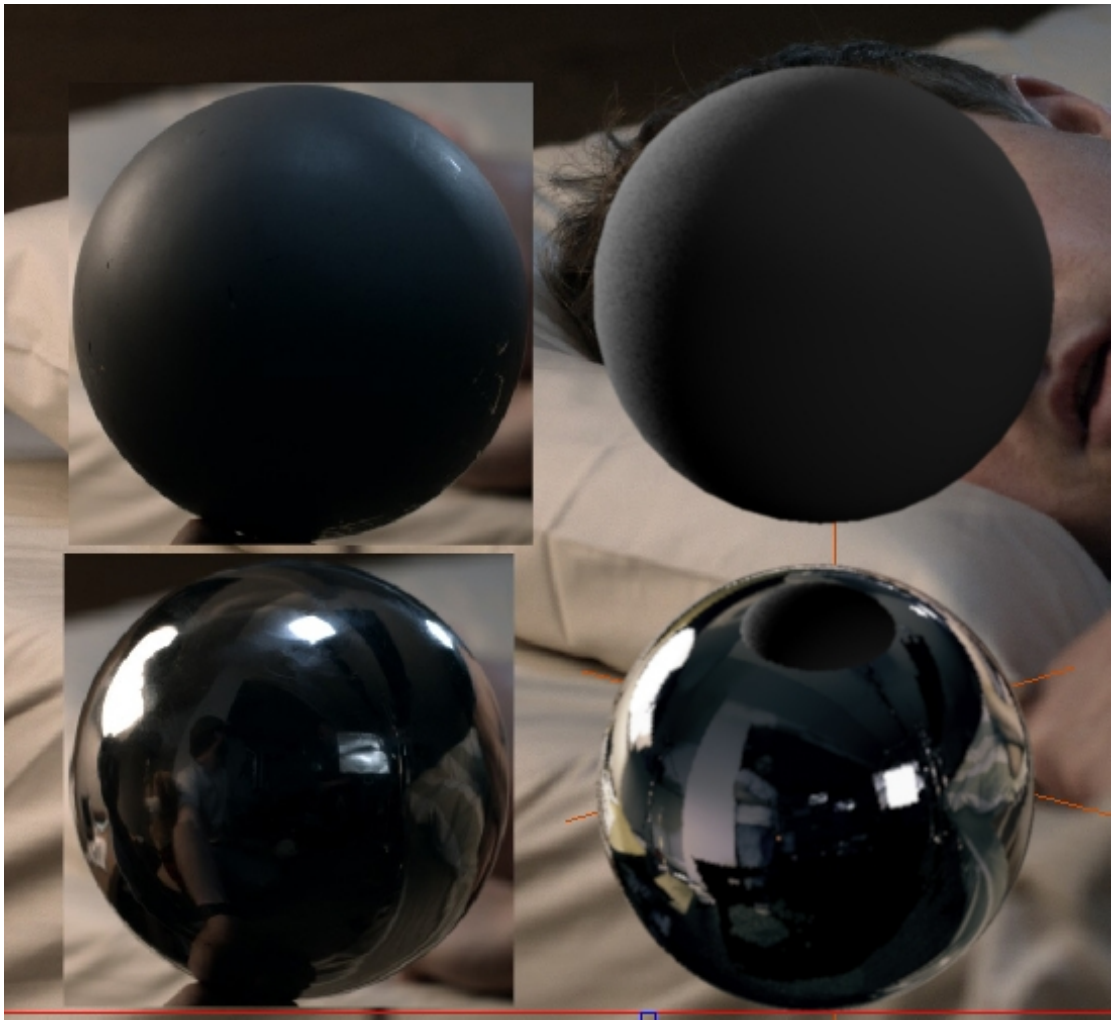

The top light should be closer to the ceiling




The back light should be further back.

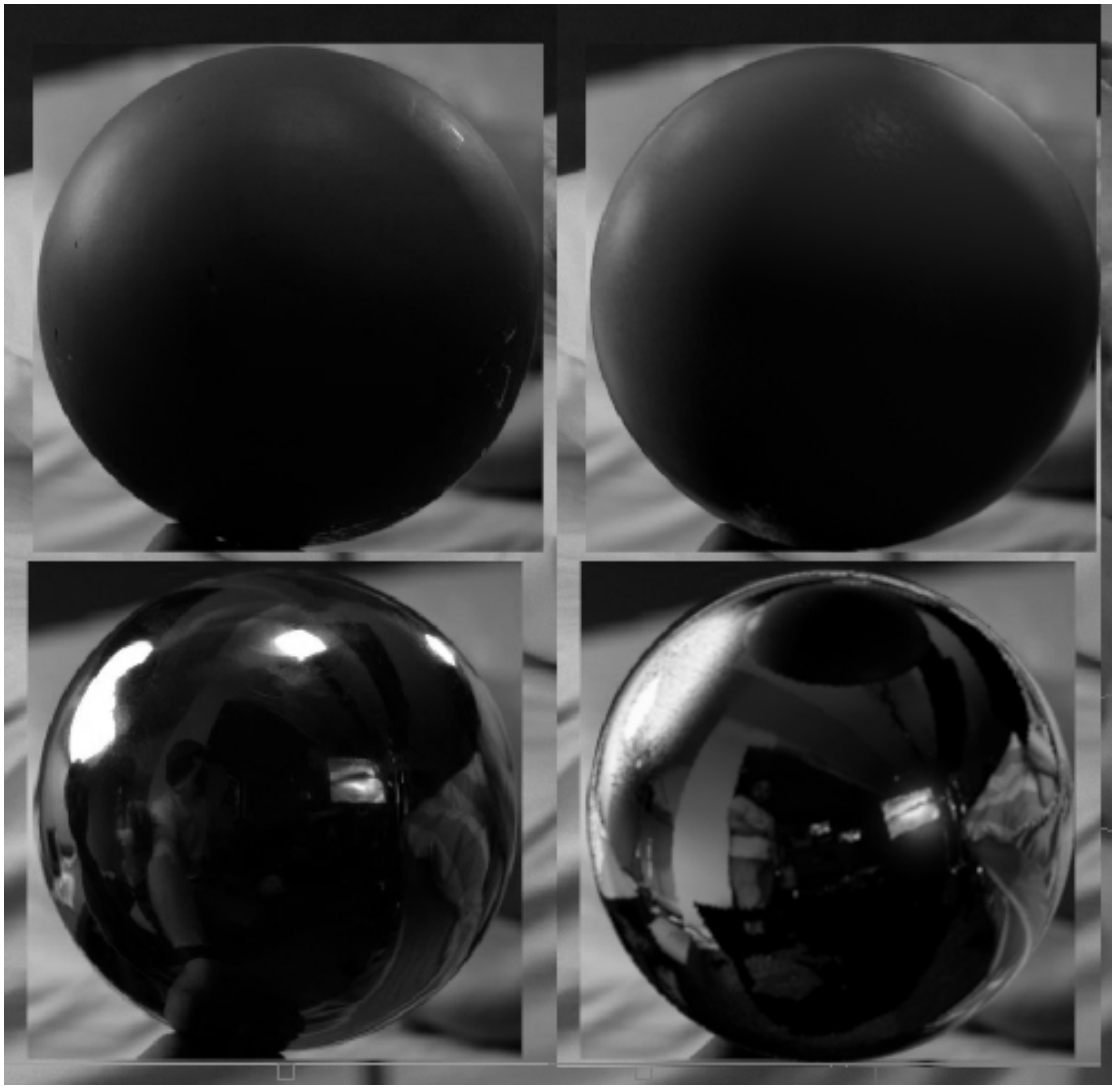
Notice that they used Blockers to create shadow areas.

UnHide the grey ball and ReRender:

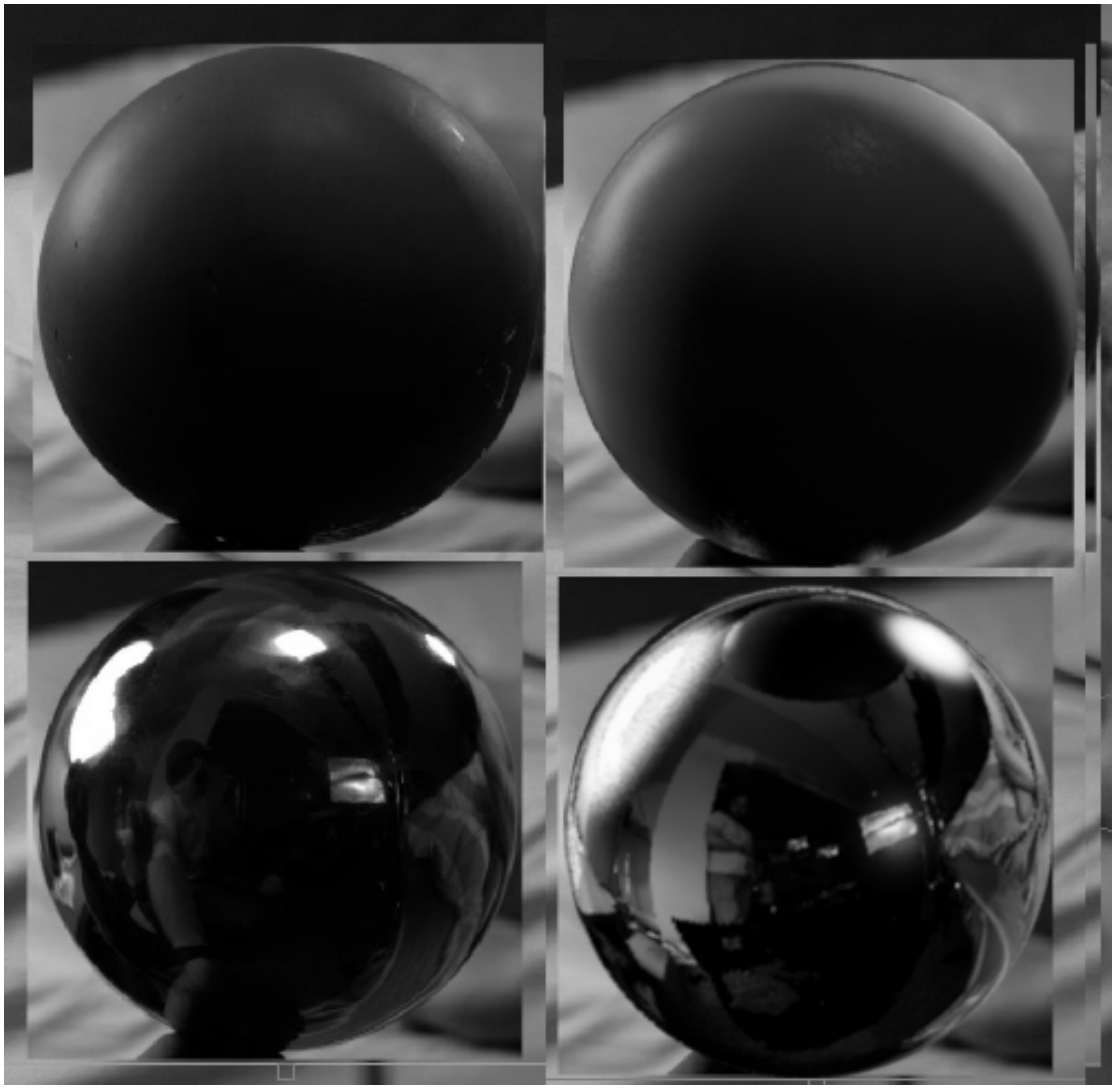


Light seams to diffuse properly on the sphere but the intensity are all wrong, change the intensity of the light until getting a grey ball as close as possible to the one on the shoot. Moving the CG grey ball directly on top of the reference one, adding a little bit of spec and checking the render in black and white might help. (See Speculars Turn of “visible in render” in the area light setting.

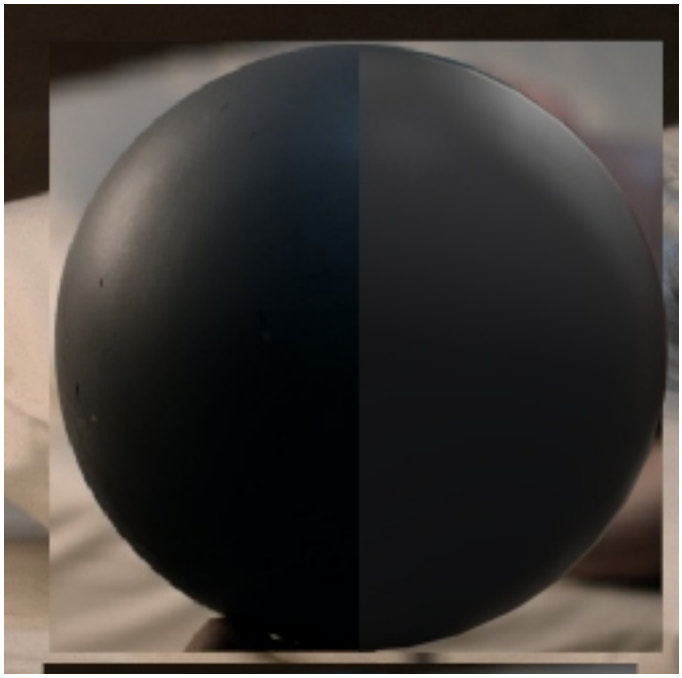
In this case I couldn’t get the same lighting without adding the Blockers:
With Blockers:



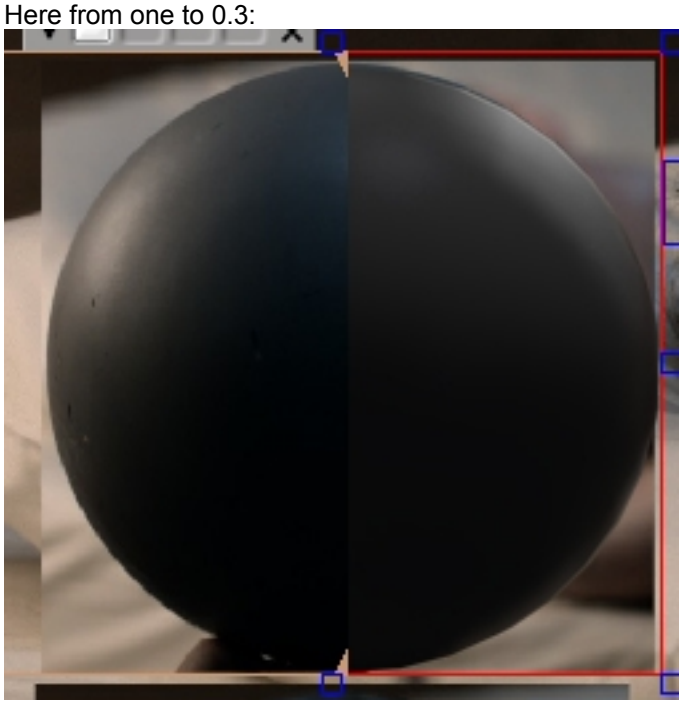
Without Blockers:



Once the lighting is as close as possible as the reference grey ball, you can turn the final gather on.
The result usually is way to bright.

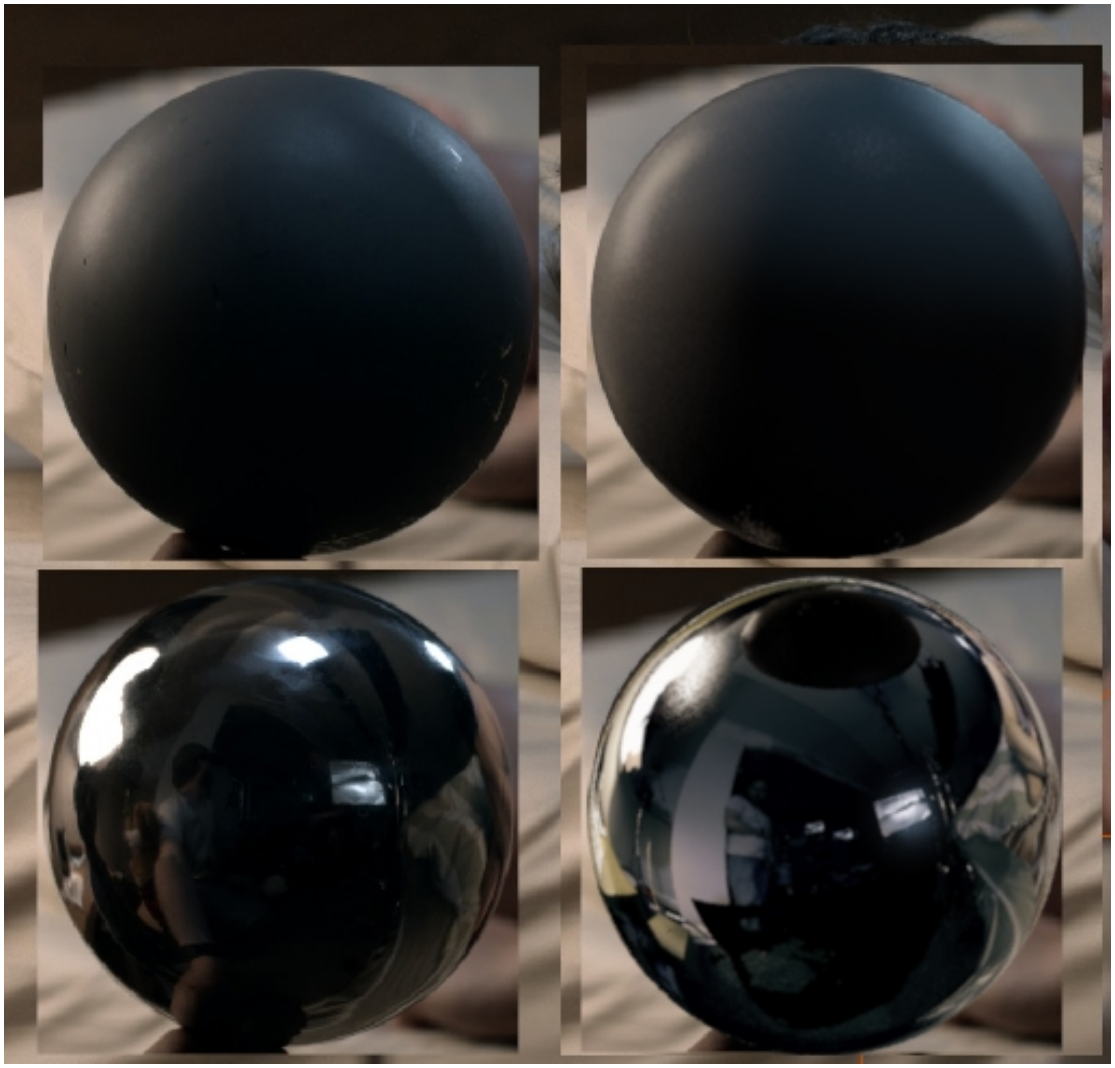


Reduce the image intensity in Environment_LR until getting something closer to the reference.
Here from one to 0.3:



You can now give some color to the light, you can get theme from the Reflection Map, in nuke, the same way as when you got the intensity, in xsi you can pick the colour, keep the hue and change the saturation and lightness or just do it by eye…

This is the most important stage and will take a bit of time to get it right.

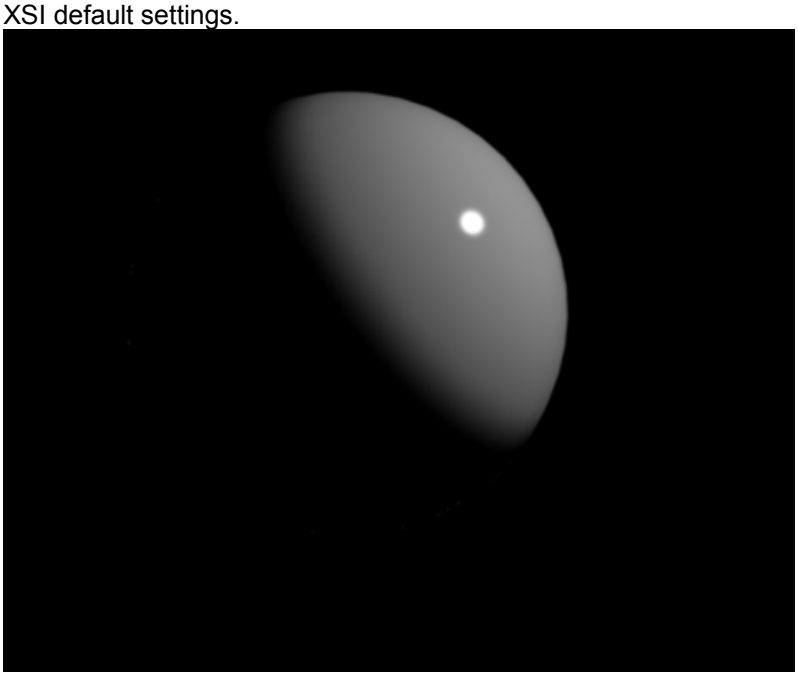


The lighting is now matched, the speculars are wrong (see speculars)

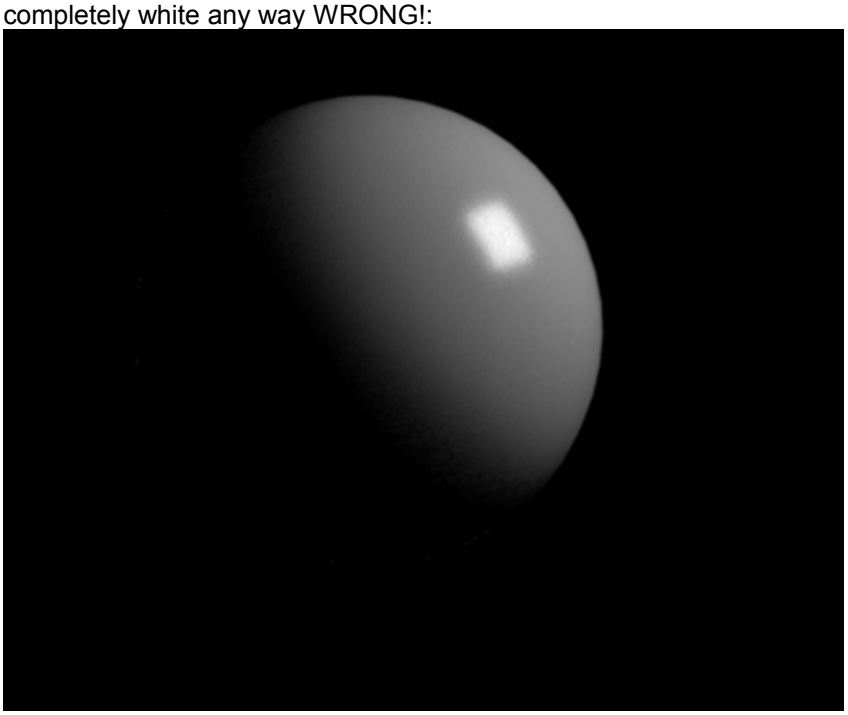
Speculars

Because of the historical way Specs were rendered, we now have to choose a way of rendering the speculars. There is various type of speculars:

- Speculars: Is the historical “high light” a blurry gradient where the light hit the surface. XSI default settings.

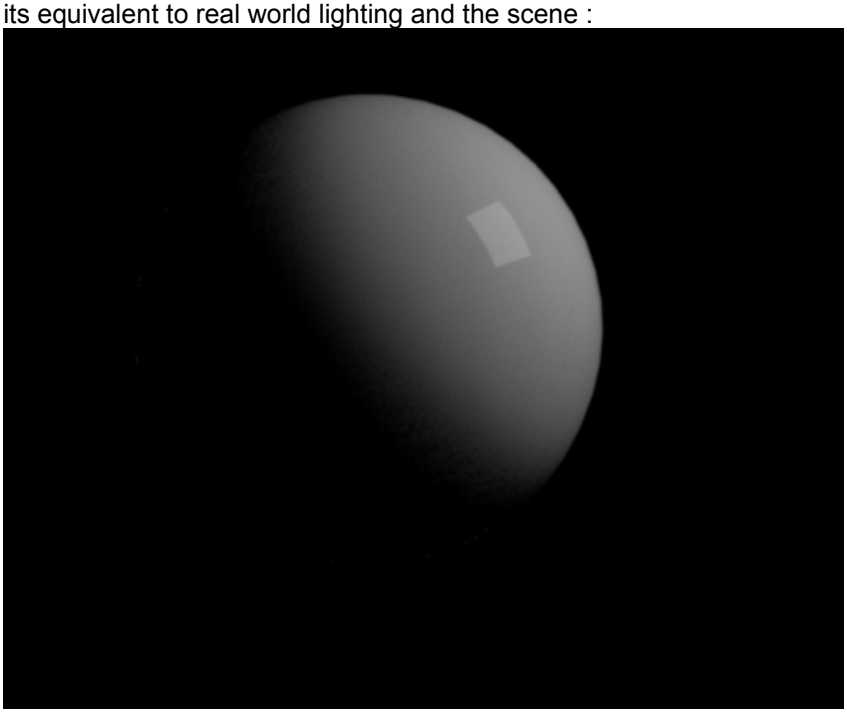


- Area Light Speculars: This happens when “area Light” is turned on and the mesh has an Architectural shader. (You should ALWAYS use an architectural shader, see SHADING) It looks already much better, but my light is at .75 and the spec is completely white any way WRONG!:

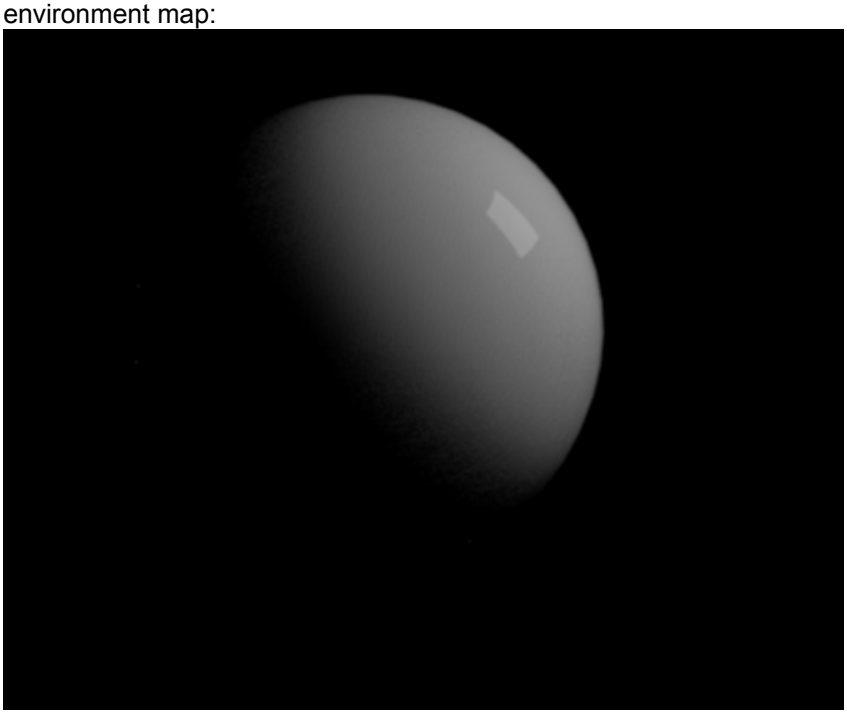


- Reflection High Light. This happens when you have bright polys in the scene or when “Area Light” and “Visible in render” are on. This is the most consistent technique as

its equivalent to real world lighting and the scene :



- Environment High Light. This is the reflection of the light points contained in the environment map:



Each of these techniques has some drawbacks.

The raison why we need to take care of the speculars is because by default we have ALL the SPECULAR TWICE

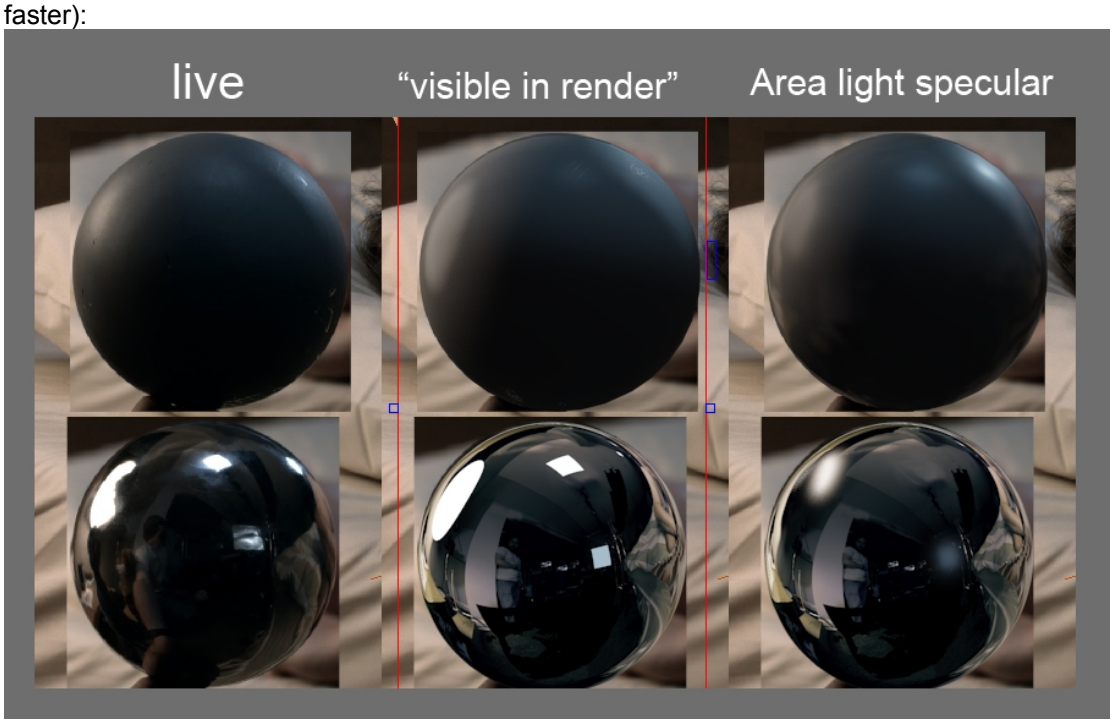
Here is 3 setups that would fix that. They can be mixed together. I'll detail the pros and cons of each setup.

Those setups can be mixed. The important is to be aware of the problems caused by double speculars.

You can have for exemple a main light full CG and all the other Full Environment etc.

*Full CG*

This is achieved by removing all the light sources in the Environment Reflection map,  with all lights as Area light (if “visible in render” is on it will be very accurate, if not it will be much faster):




render visible ticked off (Area Light Specular) makes the mirror ball look pretty bad.
I also had to change quite a lot the grey ball shader to have something similar for the two renders.

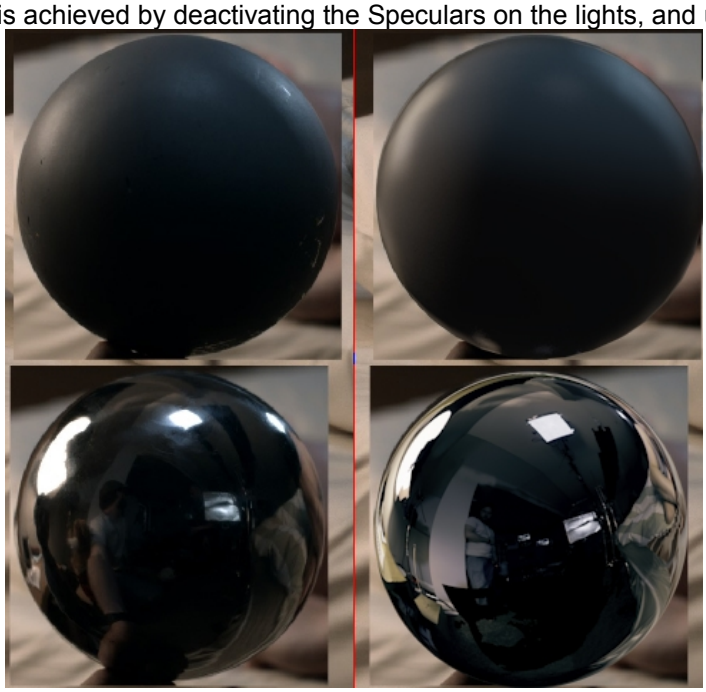
Pros:

- Extremely accurate physically and in shading
- Right in space (for moving objects)
- Interactive, you can fine tune the light direction and have spec that matches.
- You can turn down the environment reflection but keep pretty high lights, giving more contrast.

Cons:

- Expensive to render (Glossy reflection can get very noisy) > Slow to render
- Shading needs to be done specifically with this technique (See SHADING) and is a bit harder.
- Need to clean Up the Environment map
- Specific Light shape (i.e. Chinese lamps) will only be reflected as simple geometries.

*Full Environment*

This is achieved by deactivating the Speculars on the lights, and using the usual Reflection map: 
The mirror ball looks pretty good the grey ball is missing a bit of specs…

Pros:

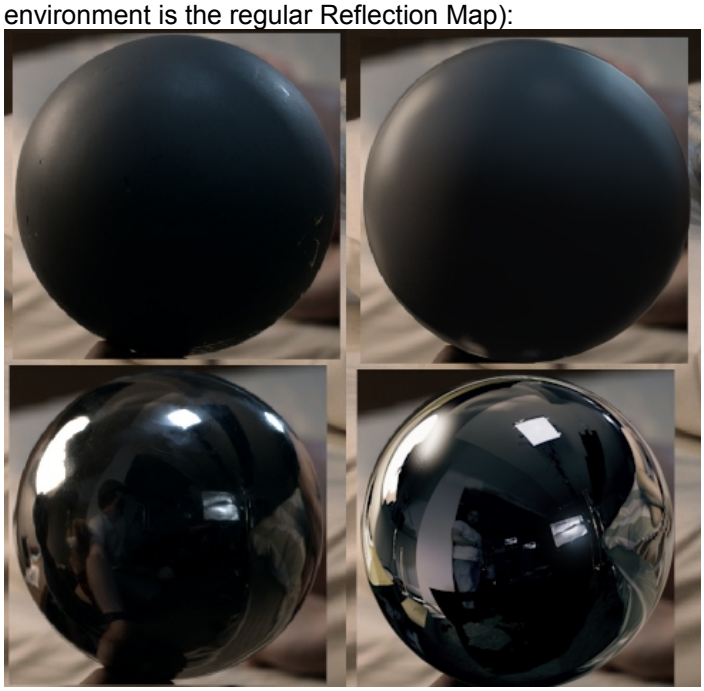
- Fast to render
- Matches the live action much more.
- Can add effects like flares and glow on the reflection map.

Cons:

- If a shader only has specular, it won't have any high light. (can be fixed by adding a spec only light, but would defeat the process)
- Intencity doesn't match light source.
- Can have disparity if the lights are slightly moved.
- It's a flat surroundings environment so if object moves the reflection interaction won't look as good.
- Reflection will need to be higher in shading (to pic up high light) and it will be harder to have a contrasted reflection look.

*Environment + Area Specs*

This is the default XSI behaviour when you turn on area light. (“visble in render” is off, the environment is the regular Reflection Map):



This is not too bad on the gray ball (I like the kick the specs get) but the spec of the left light on the mirror ball is really BAD.

NEVER use the regular environment map with area light with “render visible” ON, It would create way to much spec.

Pros:

- Works with every shaders
- Fast and clean

Cons:

- Looks good when the light are exactly in the same position as in the environment (looks like glow), bur really bad when its off
- Will give a cg look.

In this case I like the “Environment + Area Specs” better, but I will get rid of the specs on the light on the left (Chinese lamp)

Character Lighting

At this stage the grey ball and mirror ball should have been matched.
Hide “z_balls” partition and unhide a_GEO. Apply the grey ball material to this pass. (Can be found in Utility_MatLib)
In the camera rotoscop, load the Lighting Rendering Background.
This is what you should get (I’ve rendered with the brighter lambert – still in Utility_MatLib):


The lighter grey lambert looks a lot like the white-ish dog.

Let’s render with the Character shaders:

UnHide the Hair partition and re-render (keep the previous render for comparison)


In this case the difference is not enough to justify the increase in render time (Area light are too soft) so im gonna keep the hair hidden.

The render is quite dark since most of the light will be on his fur.

At this point it is best to laugh the whole shot on the farm to see it moving. And start setting up the hair lighting.

Extra Lighting:

Once the lighting is set and you've rendered the characters and tweaked the lighting, you might find that character needs extra lighting.
This needs to be done with lots of PRECAUTION as this step can ruined the integration.
As a general rule you can add back lights and fill lights BUT no key light.
You need to be careful to not change the lighting direction.

In this case rendering the characters with the correct lighting shows that Aleks face is to black on screen right, and so I'll add a light to create a nice gradient on his face:

Extra Fill light:

Trying to keep the contrast quite high.

Rim light on hair can easily be achieved in comp. But you can add them in cg too.

Hair Lighting:

At this point you should have matched your lighting and rendered the Beauty_Char. If not go to Scene Lighting

All the lighting should be in the "ON_ShadowMap_Override" partition. This deactivates the Area light and activate shadow maps.

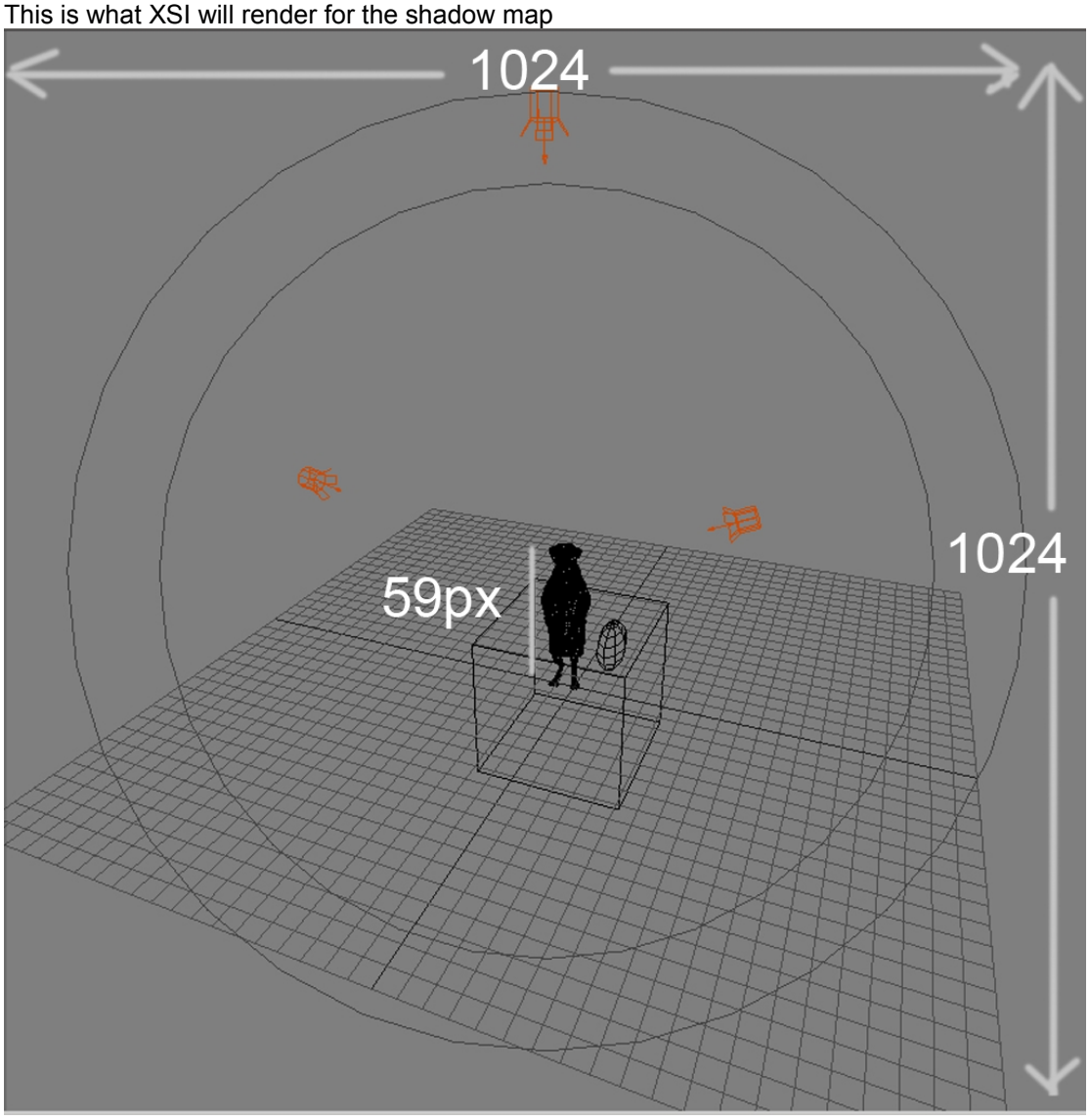
It is almost impossible to render hair with area light. And it's realy not far to be the same with raytrace shadow.

Shadow maps, usually do the trick.

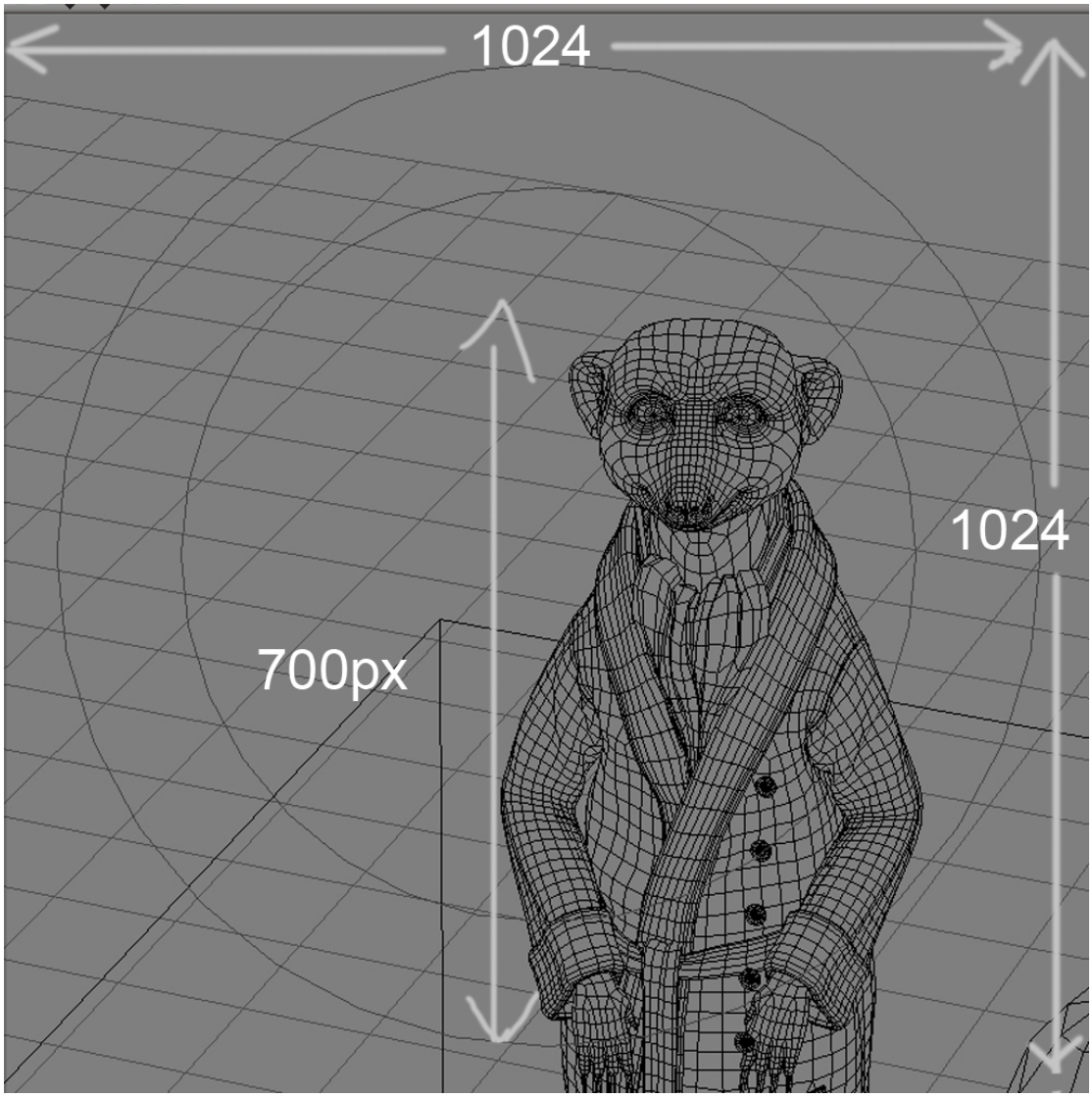
Before rendering you will need to set parameters specific to Shadow maps.
Shadow maps are a black and white render, rendered from the Spot view. It has the size difined in Shadomap size. And the limits are defined by the cone angle. The bigger the cone the less efficient the render will be and the greater size you'll need.

Switch to one of the spot in the view port

This is what XSI will render for the shadow map




The default size is 1024 pixel. The only thing we render is the character, and here it's only 59pixel wide. Even if we would render a 10k shadow map it will only be 590 pixels. This would create creasy render times and flikery renders.
Change the light cone angle to something much closer to the Character:




We just see the head in the shot, so don’t bother with the rest of the body. Just by doing this we have a shadow map of 700 px for this character.

Do the same with all the lights. Render at least once the whole shot to see if you need to update the shadow map size. A 4k map size is the maximum you should use. If you have two characters too far apart, it is preferable to duplicate the lights and aim them at each characters. (be careful to not change the Beauty_Char when doing so)

In the environment Match the setting you’ve changed in the beauty_Char pass.

Hide the hair and geo partition and unhide the Hair partition:

The lighting is not as soft. And the shadows are harder. As expected.
How ever because hair is quite fuzzy this wont be to bothering.
There is a big sharp shadow from the light ion the left that could be blurred. I'll change the shadow softness from 0.002 to 0.009. Keep in mind that when you do so, the quality of the shadow lowers, the contact can become nasty. Also the higher the shadow resolution is the sharper the shadow will be.
Hide the z_Ball and un-Hide the hair and geo partitions

I'm hiding too of the blockers because it cast too much shadow on the fur.

We deactivated the specs and its worth reactivating them in the light partition override:

Because the light is too soft it doesn’t travel as much on the hair. This happens on big area light close to objects. It’s worth creating a new light to make the lighting travel more. Duplicate the light and move it a bit so it matches the area light volume.


added specular
added Light

***Flickering***

Flicker can appear in the renders, this can come from various raisons:

If the Character beauty flickers it can be:

- Area light – when there is a kind of buzzing noise on top of the picture – Switch to raytrace shadows to double check and if it comes from the area light up the samples gradually.
- Area light are visible in reflection – High lights on object are very grainy – Untick visible in render and if you realy need the reflection, place a polygon of the same size with a constant material with the same colour as the light but with lower intensity. (Use a ray type end plug the constant in reflection. All other inputs should be black)
- Final gather – there is patches of colours glowing/pulsing – check that hair is not emitting FG and that you don't have shaders or environment emitting high intensity on small surface.

If the Hair Beauty pass flickers:

- Final gather – there is patches of colours glowing/pulsing – check that hair is not emitting FG and that you don't have shaders or environment emitting high intensity on small surface.
- Shadow maps – Dark dots or lines are changing on every frame - if shadow maps are in use, are they properly set up? (see Hair Lighting)
- White dots appears on the hair / hair appears and disappear – is the render set to raterizer? Are the parameter good enough? (1 – 2 is for preview, 6-8 is good enough for final renders.)

**COMPING**

***PLATES & ROTOSCOPY***

***INTEGRATION***

Colour Correction

Environment integration

**GENERIC**

***Import Animation***