

Unit 10: VFX CRAFT

From my experience at Double Negative, I haven't worked very closely with any one *creative* department in particular, but instead I have picked up a more basic (although broader) understanding of the challenges and responsibilities of a wide range of departments. So here I will give a brief overview of each of major roles, which will exist in almost every large VFX house.

Step 1: Concept Design

Concept Design is always a vital first step of any visual creative project that requires cooperation, whether a marketing campaign, a computer game or, more importantly, visual effects. In feature films, this is where many of the sets, costumes, and creatures are often first visualized, after only previously existing as words in a screenplay and in the imagination of the director.

Concept artists might be involved from the client side, the VFX house or even both, depending on the nature of the project. They often have a wide range of creative skills, as concept might include 3D sculpts, 2D digital art and sketches.

Artists in this area are able to have a lot of creative impact, as concept work will be used time and time again as reference in the work of many other artists.

Step 2: Build (Asset Specific Work)

In this section I will cover the main departments involved in asset design.

Modelling

Modellers create pretty much all of the 3d assets (with the exception of 3d geometry created by FX TDs) required by a project. They have to meet not just artistic goals set by a combination of client notes and reference, but also technical specifications, some likely examples being:

Correct and consistent detail level (or LOD for Level of Detail)

Models will often be created with a number of LODS, so that the full detailed object doesn't always need to be rendered when it isn't required, for example when it is in the far distance of a shot. One workflow for creating these models is by starting from reference (or, if fortunate enough, a LIDAR scan) and, starting from a simple shape, building in progressively more detail, so that the highest LOD is the last to be created.

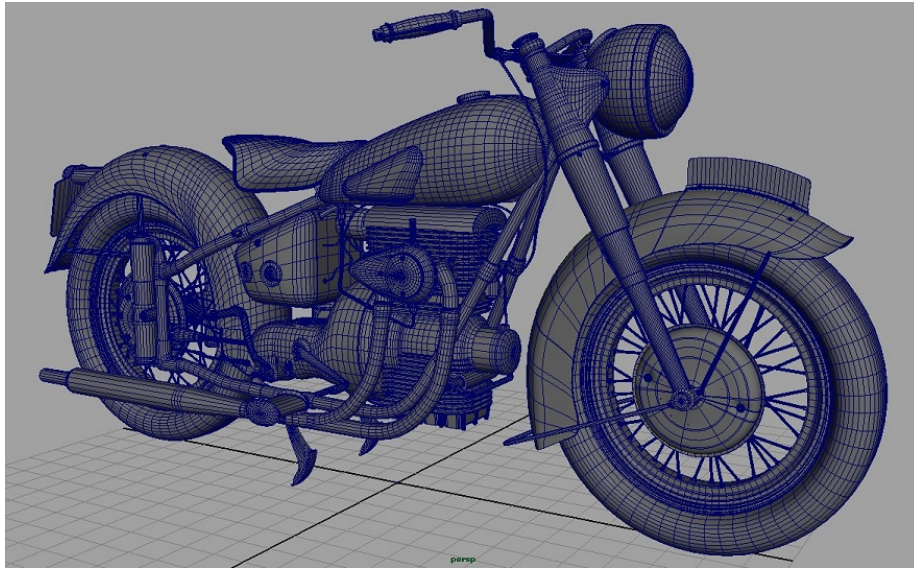


Figure 1: Model of Motorbike (*image: <https://seraphinacorazza.wordpress.com/2012/12/28/>*)

Efficient UVs, so that the ratio between UV space area and 3D surface area is consistent throughout the model.

Unless some parts of the model are more likely to have attention paid to than other parts. e.g. character models, where faces often have higher resolution textures.

Efficient mesh topology, so its structure deforms well.

This is very important in the games industry. They don't have the luxury of being able to sub-divide their meshes until they deform well, as the resource cost is exponentially expensive with each iteration.

Types of Modelling

Modelling is normally split into two categories: Hard surface and organic modelling.

Organic modelling encompasses anything with an irregular shape, or something that is mathematically hard to express. Some examples include plants, poop and most characters & animals. Sculpting is the preferred modelling method for organic modelling, which means using a program with specialised brushes that push, pull or otherwise deform parts of a model. ZBrush and Mudbox are both popular software packages that are used at studios.

Hard surface modelling is the skill of creating models that are *not too difficult to express mathematically**. But this does not in any way mean that they are any more difficult create than organic models. Hard surface modelling tasks include buildings, rigid props from bottles to mobile phones, and most furniture. These shapes can be created using the more traditional and mathematical modelling tools, such as extruding and adding edge loops. Programs used to create these models include Autodesk Maya and SideFX Houdini.

These models are then passed of to texturing and rigging (if they are to be animated).

**Not an exact definition, but this was the result of reading lots of artists' individual opinions*

Rigging

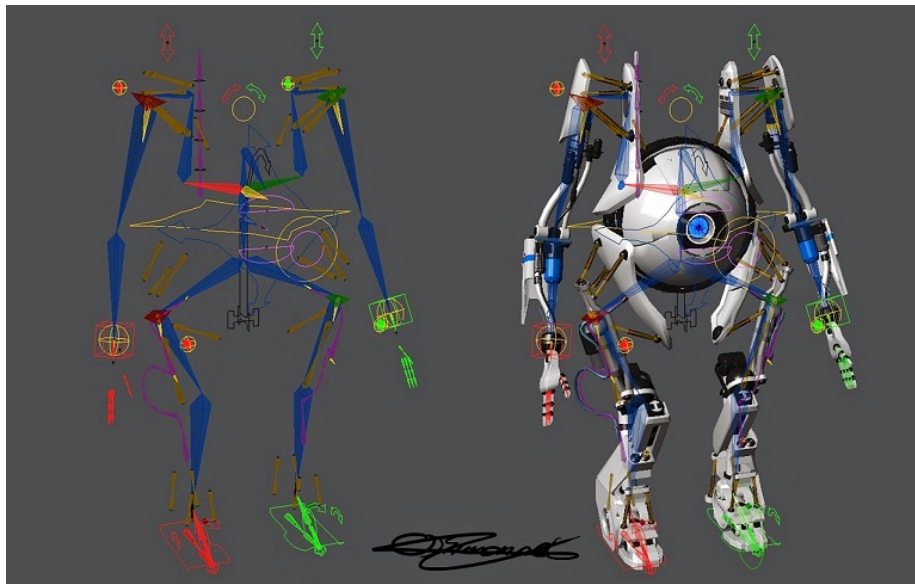


Figure 2: Portal Rig (image: James Ball - <http://ballzy247saeblog.blogspot.co.uk/2015/02/>)

A Rigger, sometimes called a Rigging TD, builds controllable rigs for models that require animation, such as vehicles and characters. For characters, this can involve building a skeleton which they bind to the mesh. Per-vertex weights are then painted for each skeletal bone, which define how much influence that bone has on the vertex. These bones are normally hidden and more user-friendly controls are created.

It takes a lot of attention to detail and experience to make photorealistic,

believable deformation. One challenge is the issue of volume preservation. With a skeleton of joints, as described above, even the most accurate painted weights will not maintain a consistent volume with the most complex animation, as the system has no concept of matter under the skin.

Riggers also work with muscle systems, which enable a more complex way of deforming an animal/living character. Shapes, defining individual muscles, are to be created and connected to the skeleton rig. This results in more realistic definition in the skin, and can help with volume preservation problem.

The most commonly used software package for this (in VFX) is Autodesk Maya, though other 3D packages, such as SideFX Houdini and Autodesk 3DSMax also have their own rigging systems.

Texture & Shader design



Figure 3: Logan Ref/Lookdev (*image: Image Engine*)

A texture artist designs the look of the surface of the model, by creating 2d maps, based on the model's UVs, that are “wrapped around” the model. These can be drawn digitally in a 2d program such as Adobe Photoshop, or artists can use more specialized texturing software such as Autodesk Mudbox and The Foundry's Mari, which provide lots of useful utilities which allow you paint directly on the model, and export textures afterwards*. This is advantageous as it means the texturer does not need to worry about issues related to making textures line up at seams (seams being the edges of separate UV shells that correspond to the same edge on the model).

Examples of maps that a texture artist might have to produce:



Figure 4: Door Texture Examples

Diffuse Maps

These maps show the colour light that an object reflects. Ideally with no lighting effects baked in (such as reflections or specular highlights), these define the main colours of a model.

Bump Maps (or Height maps)

Bump maps are generally monochrome maps which shows displacement of a texture perpendicular to the object. This is calculated by the value at a point of a bump map being multiplied by a constant, with a value of 0 being the most “pushed-in” point and 1 being the most “sticky-outy” (technical term). Though It is simply a lighting trick, and adds no detail to the geometry.

Normal Maps

These bump maps can then be used in the process generate normal maps, a more accurate type of bump map. These are RGB maps that store the direction of the surface normals of a texture in a clever way. The R G & B channels respectively correspond with the X Y and Z channels effectively giving you a 3D vector for each pixel of the UV map. These are then used to more realistically calculate the shading of a model.

Specular Maps

These are RGB maps that define the “shinyness” and specular colour of an object. The pixel intensity is proportional to the amount of light reflected from the point, and the RGB colour defines the colour of the reflection.

Displacement Maps

As far as I can tell, these are identical to bump maps, in the way that they are monochrome maps that store perpendicular depth data. Though the difference is that displacement maps are actually used to deform the mesh, usually after being subdivided. I can only assume that these are used because working with super-high poly models is very computer-resource intensive, whereas working with 2d monochrome images isn't.

**I should mention that Photoshop had the ability for painting onto models for a number of years, but it's 2D compositing capabilities are more common and much stronger*

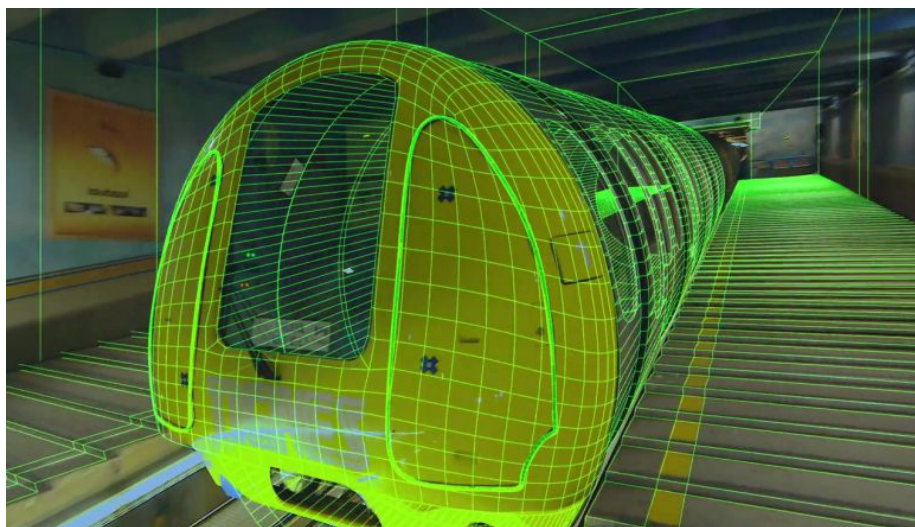


Figure 5: Matchmove geometry over scan (*image: QLBEANS*)

Step 3: Shot Specific Work

Matchmove

Matchmoving is a massively important technique in visual effects which enables CG elements to have matching translation, rotation and scale values to elements shot in the live-action scan. The most common way this is done is by individually tracking a number of 2D points in a scan, at different depths in real-world space, and by reconstructing a 3d camera using complex math. This is the core of how the tracking system working in popular matchmoving software such as 3DEqualizer and PFTrack, though this system can sometimes struggle with footage that doesn't have many high-contrast, easy to track points. Another method, used in mocha Pro, that manages this issue quite well, is planar tracking, which aims to track whole surfaces, rather than individual points, though it doesn't work so well when there are no flat 2d planes, so there is no all-round best solution. Similar techniques to these can also be used to reconstruct 3D geometry and textures too.

Bodytracking is another important matchmove task, which is the skill of matching a CG digi-double's animation to an actor in a live-action plate. This geometry can then be cached out and used later down the line by other departments, for example by lighting, if a full-CG replacement is required, or by FX if they need a simulation to react with the actors body.



Figure 6: Matchmove pic (*image: Digital Domain*)

Animation

3D animators make the characters passed down by modelling and rigged by rigging TDs come to life through keyframe animation, which is a technique I explain below. The goal of animators is to create a believable performance using a combination of reference and animation theory.

Keyframe Animation

All rig controls will have at least some channels, possibly for controlling translation and/or rotation. These channels are then assigned certain values at specific frames, creating a keyframe. The channel's value between these keyframes are interpolated, either linearly or through using Bezier splines. These channels can be normally be edited by moving the handles in the viewport of your 3D program, or by manually editing the curves in the program's animation editor (sometimes called a graph editor), which will show you an interactive graphic visualisation of the animation curves.

FX



Figure 7: Houdini FX Sim (*image: Carlos Parmentier*)

FX TDs (short for Effects Technical Directors) are responsible for using simulations in order to create creative 3D geometry caches which might model explosions, fire, destruction or fracturing shapes, or any other abstract or complex animation that doesn't easily fit into the role of other departments.

There is also another branch of FX, sometimes known as creature FX, who are responsible for using simulations upon animated geometry of characters, in order to make them seem more real. The kind of tasks these artists might have to simulate could include character hair (also called a groom), skin & muscles, and costumes.

There are always new simulation methods being published, and many well-known techniques are implemented and developed upon in software such as Autodesk Maya and SideFX Houdini.

Lighting



Figure 8: Lighting Before/After (*image: Dylan Sisson*)

A lighting TDs main role is to light 3D geometry in a way that matches another plate/scans lighting. This can be done either by using lots of different types of cg lights to create a rig, or also by using a HDRI (High Dynamic Range Image). HDRIs describe 360 degree images created by stitching together images of an environment taken from the same nodal point, at multiple exposure levels. This results in an image that ideally has data from all light sources from the darkest blacks to the brightest whites, with no clipping values, to realistically simulate the original environment's lighting conditions.

Another important responsibility of a lighting TD is make sure there is consistency in the lighting in all of the elements in the shot. This sometimes mean that the most physically realistic solution is not always the best.

Render configuration and efficiency, managing motion blur and noise, and

writing/editing custom shaders might also be included in a lighting TD's responsibilities.

Step 4: 2D

Roto



Figure 9: roto example

image + in-depth tutorial: <https://pirayaganjanakulnon.wordpress.com/2016/01/12/module-01-assignment-01-rotoscope-shot/>

Roto artists are responsible for creating alpha mattes from scans which isolate certain features. Ideally, a key would be easier and more efficient, but it is not always possible or practical to shoot in front of an evenly lit screen. The most common way this is done is by trying to break down the movement of the desired element in a way it can be defined by as few 2d spline-shapes as possible. Then these shapes are transformed and translated to follow the movement in the scan. Compositors can then take these elements and layer them over CG elements or combine them with other scans.

Prep/Paint

Prep have the job of cleaning up scans, ready for comp, mainly by painting. Painting, in the context of Prep, refers to the skill of believably reconstructing detail in a part of a scan that has been partially or completely occluded, by sampling from similar areas on the same or surrounding frames, or reference images. Some common paint removal tasks are required for:



Figure 10: prep example (*image: <http://www.btlnews.com/crafts/visual-fx/level-256-produces-126-vfx-shots-for-the-bourne-legacy/>*)

- Dust shadows from footage shot on physical film.
- Wires from actor harnesses.
- Accidental or unavoidable boom poles, camera rigs and film crew members.
- Items that conflict with latest creative decisions or that create continuity issues.

Comp



Figure 11: Mad Max Fury Road Comp (*image: Iloura*)

A comper's role is to combine the work of all of the earlier artists departments into a final shot. This involves combining any CG elements, DMP (Digital Matte Painting) work, prep elements and roto mattes with live action scans, in a way that looks believable enough to fool a viewer that it was shot once, through the lens of a single real camera.

An organised example of a Nuke node graph:

The most commonly used industry compositing software package is currently Nuke, which has been developed by The Foundry since 2007. It uses a node-based system, rather than the layer-based system found in compositing software such as the also very popular Adobe After Effects. One significant advantage of the node-based workflow is that it makes collaboration much easier. You can easily get an idea of what is happening in a well-organised nuke scene by zooming out and looking at the entire node graph, something that is much more difficult to do in a layer based system, especially when you are dealing with very large scene's (which is very likely in a VFX studio).

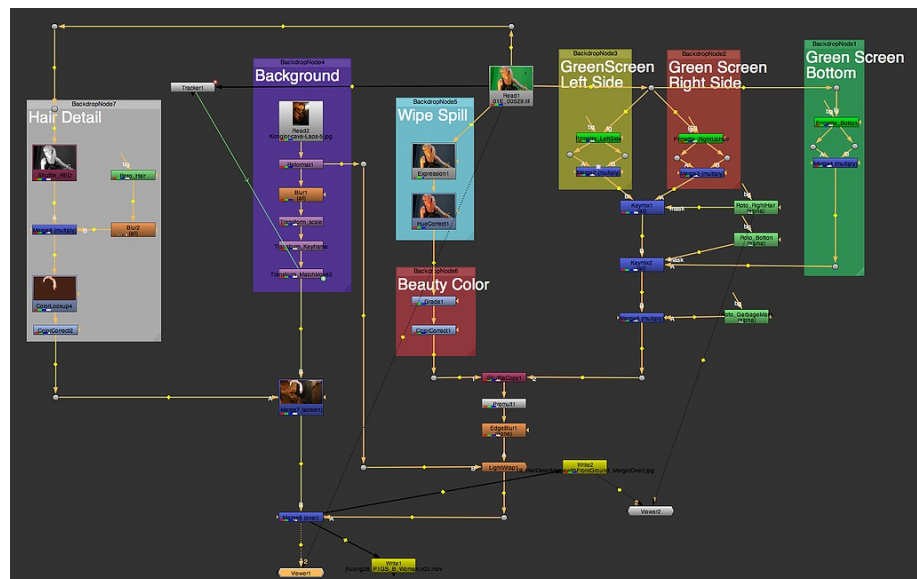


Figure 12: Example Node Graph (image: <http://www.jinguanghuang.com/compositing>)