Renderman on Film

Combining CG & Live Action using Renderman
with examples from *Stuart Little 2*

Rob Bredow
Sony Pictures Imageworks
rob@185vfx.com

Abstract

We present a complete (albeit brief) summary of the digital production process used for creating computer generated images in the context of a live action motion picture citing examples from the films *Stuart Little* and *Stuart Little 2*. Special attention is paid to issues relating specifically to Renderman including considerations for *shading*, *lighting* and *rendering* for use in feature film effects. We also touch on several compositing techniques required to complete the production process.

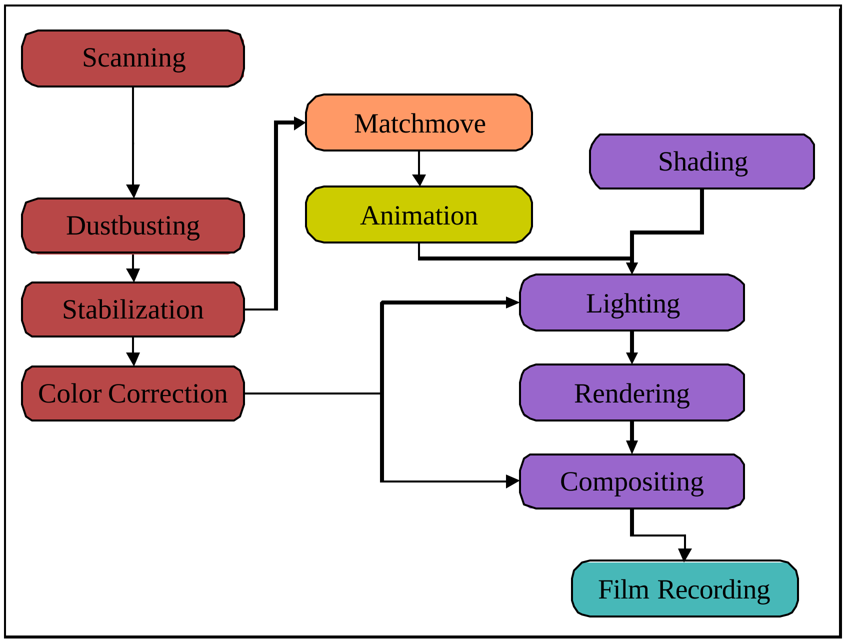


Figure 1 -Summary of the digital production pipeline

1.1 Pre-Renderman

There are several steps when working on a live action film that must take place before any rendering can begin. For the purposes of this course we will skip over the many aspects of live-action photography and editing and start with the digital production process.

1.1.1 Scanning

Once a shot has been in the editorial process as the "take" that will be in the movie and requires an effect, it can be scanned. This is generally done based on a series of keycode in and out points designated by the editorial department. Each frame is scanned individually and stored in the computer as a sequence of numbered images.

The digital film scanning process is designed to record as much of the information stored on the film negative as possible while considering the downstream ramifications of creating extremely large files for each frame. There are two aspects to consider when determining how much detail you are interested in preserving from the original negative: *Image resolution* and *color resolution*.

Image resolution is simply the number of pixels you want to generate in the scanning process. An image resolution of 4096 wide by 3112 tall is commonly referred to as a "4k" image and is often considered the highest useful resolution scan for today's 35mm motion picture negative.

Color resolution is the number of bits of data that you use to store each pixel. The most common standard of 8 bits for each of the red, green, and blue channel (a 24-bit image) yields only 255 shades of gray which is not sufficient to record all of the values that can be represented with film. A more complete representation can be stored by doubling the storage to 16 bits for each of the channels resulting in over 65,000 shades of gray which is more adequate to describe film's contrast range and subtlety. It is most common however to use the standard detailed by Kodak's Cineon file format (for more information see the Kodak paper “Conversion of 10-bit Log Film Data To 8-bit Linear or Video Data for The Cineon Digital Film System”). This standard is specified as scanning 10 bits of data per channel and storing the data in a logarithmic space to preserve as much detail in the areas of the curve that correspond to subtle gradations that the eye can clearly see.

Many visual effects films today will choose to work at a resolution of 2k and 10 bits per pixel. Often this level of detail is high enough that the differences between the work print generated off of the original camera negative and a print generated from the "digital" negative are not perceptible.

1.1.2 Dustbusting

Once the film has been scanned, it invariable has imperfections that need to be removed. These issues can manifest themselves as scratches on the negative which show up as white lines or dots, lens imperfections that need to be removed, or dark specs and lines that correspond to dust and hair. All of these are removed in the *dustbusting* process that consists of a series of both automated tools and careful frame-by-frame paintwork to prepare the plates for the downstream process.

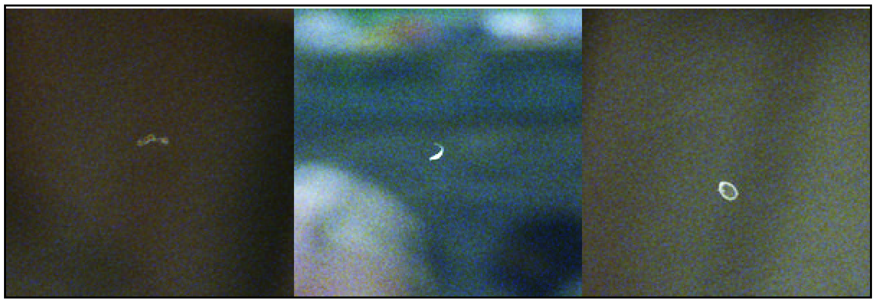


Figure 2 – Three samples of dust and scratches from negative damage

1.1.3 Stabilization

Even with advances in steadying both the in-camera and scanning technology, most every plate requires stabilization before it is ready to be enhanced with visual effects. Most frequently, the stabilization process is simply required to take out a small amount of weave inherent to the film shooting and scanning process. This can be done with an automated 2-d tracking system and some simple math to smooth the resulting curves that will "unshake" the footage on a frame-by-frame basis.

In some more complicated cases, there are bumps in the camera move or a shake introduce by a long boom arm or some other on-set tool that needs to be stabilized out or locked-down to improve the look of a shot. In this case, an entire suite of tools may be needed to stabilize and perspective correct the photography.

1.1.4 Color Correction

Scanning negative and viewing the resulting image directly does not generally result in aesthetically pleasing image. The director of photography (D.P.) shoots a movie with the process of printing in mind and being able to control the exposure level and color balance during that process. In order to be able to, later in the pipeline, match colors and exposures of computer-generated objects to the plate photography, the plate must be "timed" or color corrected to match the DP's requirements.

This is accomplished with a careful process of adjusting the exposure level and color balance with digital tools that emulate their real-world counterparts in the "print timing" process. These color corrections are then filmed out and approved by the director of photography before any lighting work can begin.

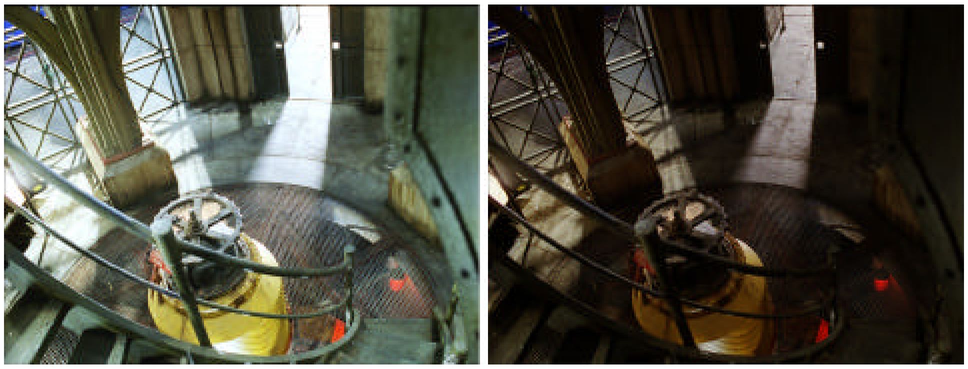


Figure 3 - Original scan (left) and color corrected plate (right)

1.1.5 Matchmove

The matchmove process is a fundamental step when working on a live action effects film. It involves duplicating, in the digital environment, the complete shooting environment that was used on the set including, most importantly, the camera.

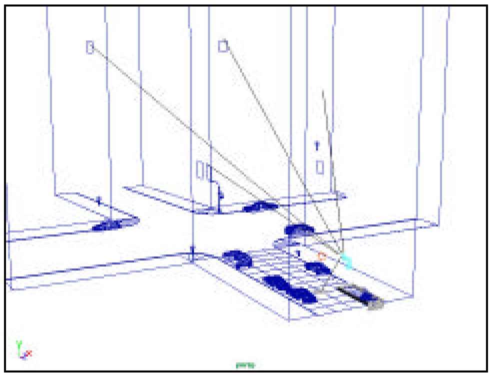


Figure 4 - 3d model of the live action set

First, a relatively simple model of the set is generated in the computer whose proportions match as closely as possible to the live action stage. This model will be used later in the process to help the animators know where to place the characters and, in the rendering process, it will catch shadows and reflections as needed.

The second major part is the tracking of the camera in relation to this model. This is a very precise and challenging task which consists of matching the lens and camera properties to the camera on set as well as the camera's position and orientation over time for each frame of the plate.

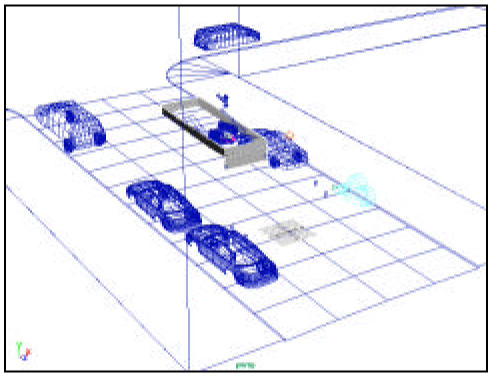


Figure 5 - Camera placed in the virtual set

There are many tools (some semi-automated) that help in both the reconstruction of the digital set and the tracking of the digital camera to this set. The specifics of which are too lengthy to detail here.

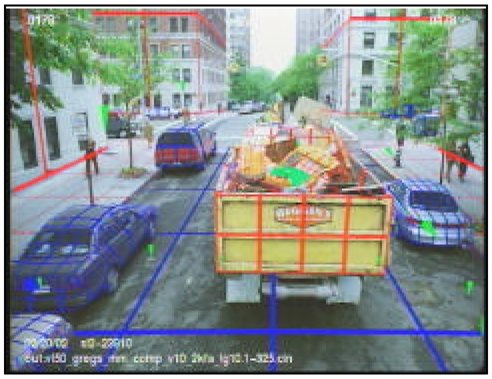


Figure 6 - Single finished matchmove frame

1.1.6 Animation

It would be naïve to attempt to summarize in any meaningful was what takes place in the animation step of the production process in our limited space. For our purposes, the design and movement of all the props and characters can now be accomplished and approved before any rendering can take place.

1.2 Renderman

It's at this point that the Renderman related processes can begin. The "look development" process gets things started by designing the shaders and setting the look for each of our characters and props. Once the look has been established, the lighting can begin and the CG will be lit to fit into the scene and then be enhanced for dramatic effect as needed. Then the process of rendering the various passes can be undertaken.

1.2.1 Shading

There is one key concept to keep in mind when writing shaders for visual effects: control. Most of the lighting models that are most useful in production are based more on the fact that a lighter can predict what the shader will do in a given case and less on physics or mathematical accuracy. This methodology has lead to a class of lighting models that could be categorized as "Pseudo-realistic lighting models." We will detail a couple of useful lighting models that would fall into such a category here.

1.2.1.1. Diffuse Controllability: Falloff start, end and rate.

Diffuse shading calculations are the most common and simple way to shade an object. Creating a flexible diffuse shading model greatly increases controllability.

The figure below illustrates a standard Lambertian diffuse shading model.

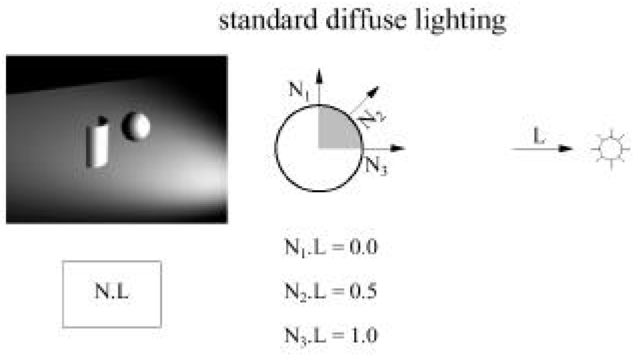


Figure 7 - Standard diffuse lighting model

1.2.1.1.1 Falloff End (Wrap)

The first step in generating soft and nicely wrapped lighting is to give the light the ability to reach beyond the 90 degree point on the objects. This has the effect of softening the effect of the light on the surface simulating an area light. This can be done with controls added to the lights which specify the end-wrapping-point in terms of degrees where a wrap of 90 degrees corresponded to the standard Lambertian shading model and higher values indicated more wrap. This control is natural for the lighting TD's to work with and yields predictable results.

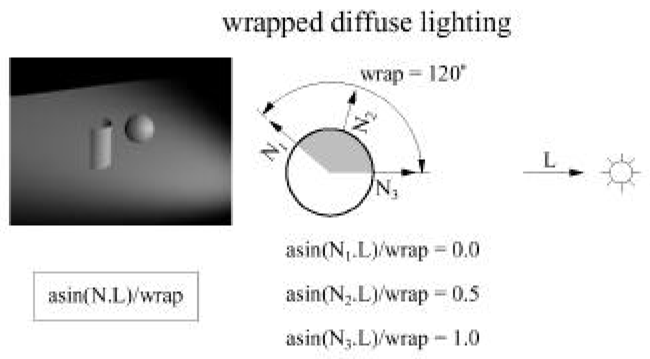


Figure 8 - "Wrapped" diffuse lighting

For wrapped lights to calculate correctly, the third argument to the illuminance() statement must be set to at least the same degree as the wrap value for the highest light. Otherwise lights will be culled out from the lighting calculations on the back of the surface and the light will not correctly wrap. In our implementation, the wrap parameter was set in each light and then passed into the shader (using message passing) where we used the customized diffuse calculation to take into account the light wrapping.

As an aside, wrapping the light "more" actually makes the light contribute more energy to the scene. If the desired effect is to keep the overall illumination constant, it will be necessary to reduce the intensity light control while increasing the wrap.

1.2.1.1.2 Falloff Start

The natural opposite of the "Falloff End" parameter which controls wrap is the "Falloff Start" parameter that controls the area that is illuminated with 100% intensity from a light. The "Falloff Start" parameter is also specified in degrees and is defaults to a value of 0. By increasing this value, the light's 100% intensity will be spread across a larger area of the object and the gradation from the lit to the unlit are of the object will be reduced.

The "Falloff Start" parameter has no logical counterpart in real life and has fewer uses than the "Falloff End" parameter previously discussed. The most common use that we found in production was in the case of a backlight or a rim around a character. When increasing the "Falloff End" or wrap to a high value, sometimes the rim would not look strong enough or have a hard enough falloff (since it roughly simulates the effect of an area light). By increasing the "Falloff Start", you get a sharper falloff around the terminator of the object and effectively increase the sharpness of the rim light.

1.2.1.1.3 Falloff Rate

Gamma is one of the most useful and convenient functions in computer graphics. When applied to lighting, a gamma function leaves the white and the black points unchanged while modifying the rate of the falloff of the light. This has the effect of modifying the value of the sphere at the “N2” point in the diagram above.

The apparent softness or sharpness of a light can be dialed in as needed using these three controls:

"Falloff End", "Falloff Start",and "Falloff Rate".

1.2.1.1.4 Shadows

The first problem we encountered when putting these controllable lights to practical use was shadows. When you have shadow maps for an object, naturally the backside of the object will be made dark because of the shadow call. This makes it impossible to wrap light onto the backside of the object.

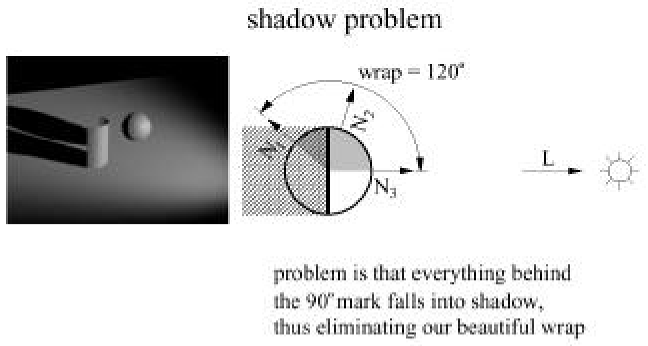

shadow problem
wrap = 120°
N2
N1
N3
L
problem is that everything behind
the 90° mark falls into shadow,
thus eliminating our beautiful wrap

Figure 9 - The shadow problem with wrapped lights

Fortunately Renderman affords a few nice abilities to get around this problem. One way to solve this problem is by reducing the size of your geometry for the shadow map calculations. This way, the object will continue to cast a shadow (albeit a slightly smaller shadow) and the areas that need to catch the wrapped lighting are not occluded by the shadow map.

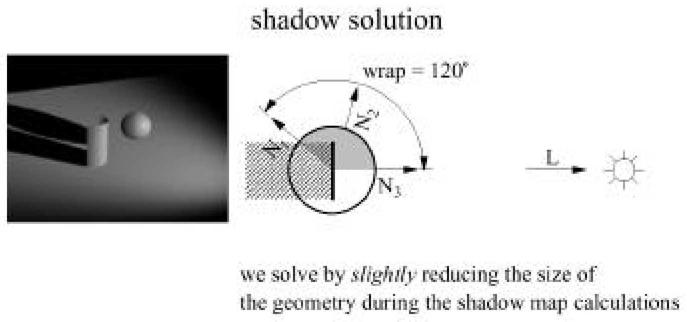

shadow solution
wrap = 120°
N2
N1
N3
L
we solve by slightly reducing the size of
the geometry during the shadow map calculations

Figure 10 - The solution to the shadow problem

As a practical matter, shrinking an arbitrary object is not always an easy task. It can be done by displacing the object inwards during the shadow map calculations but this can be very expensive.

In the case of our hair, we wrote opacity controls into the hair shader that were used to drop the opacity of the hair to 0.0 a certain percentage of the way down their length. It is important to note that the surface shader for an object is respected during shadow map calculation and if you set the opacity of a shading point to 0.0, it will not register in the shadow map. The value that is considered “transparent” can be set with the following rib command:

```
        Option "limits" "zthreshold" [0.5 0.5 0.5]
```

Lastly and most simply, you can blur the shadow map when making the "shadow" call. If your requirements are for soft shadows anyway this is probably the best solution of all. It allows for the falloff of the light to "reach around" the object and if there is any occlusion caused by the shadow, it will be occluded softly which will be less objectionable.

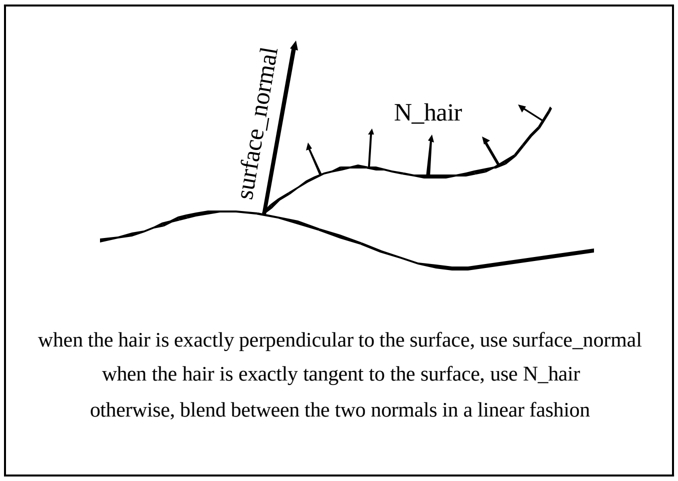
For both *Stuart Little* 1 and 2, we used a combination of shortening the hair for the shadow map calculations and blurring our shadows to be able to read the appropriate amount of wrap.

1.2.1.2. Hair controllability

Solving the problem of how to properly shade hair and fur in a way that is both realistic and easy to control is a challenging project. In our work on *Stuart Little* we experimented with conventional lighting models before deciding to create a model that was easier to control.

When using a more conventional lighting model for the fur (essentially a Lambert shading model applied to a very small cylinder) we came across a few complications. First, a diffuse model that integrates the contribution of the light as if the normal pointed in all directions around the cylinder leaves fur dark when oriented towards the light. This may be accurate but does not produce images that are very appealing. In our testing, we felt that the reason that this model didn't work as well was because a lot of the light that fur receives is a results of the bounce of the light off other fur in the character which could be modeled with some sort of global illumination solution, but would be very computationally expensive.

The other major reason that we chose to develop a new diffuse lighting model for our fur was that it was not intuitive for a TD to light. Most lighting TD's have become very good at lighting surfaces and the rules by which those surfaces behave. So, our lighting model was designed to be lit in a similar way that you would light a surface while retaining variation over the length of the fur which is essential to get a realistic look.



when the hair is exactly perpendicular to the surface, use surface_normal
when the hair is exactly tangent to the surface, use N_hair
otherwise, blend between the two normals in a linear fashion

Figure 11 - Obtaining a shading normal

In order to obtain a shading normal at the current point on the hair, we mix the surface normal vector at the base of the hair with the normal vector at the current point on the hair. The amount with which each of these vectors contributes to the mix is based on the angle between the tangent vector at the current point on the hair, and the surface normal vector at the base of the hair. The smaller this angle, the more the surface normal contributes to the shading normal. We then use a Lambertian model to calculate the intensity of the hair at that point using this shading normal. This has the benefit of allowing the user to light the underlying skin surface and then get very predictable results when fur is turned on. It also accounts for shading differences between individual hairs and along the length of each hair.

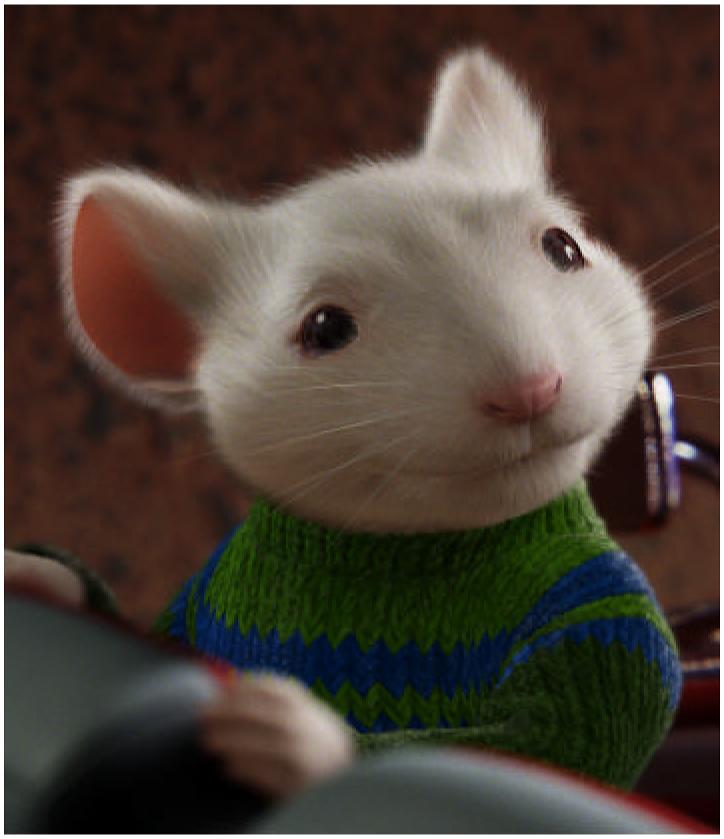


Figure 12 - Finished *Stuart Little 2* rendering with fur shading

1.2.1.3. "Ambient Occlusion" technique

In the past, most computer graphics lighting techniques have relied on a very simple model to approximate the contribution from the ambient light which does not come from a specific light source but rather bounces around a set and lights an object from all directions. Because modeling all of the actual light bouncing throughout a set is very computationally expensive, many shaders substitute a constant for this ambient value (commonly referred to as the "Ka"). This constant is generally dialed in by the lighting artist and is a rough approximation of the light from the bouncing from nearby objects.

The disadvantages of using such a simple ambient lighting model are obvious. All the different sides of an object will get the same ambient contribution no matter which direction it is facing or what shape it is. That can lead to objects looking flat and uninteresting if the "Ka" is turned up too high.

One workaround that has been used for years in this area is to leave the "Ka" value set to 0.0 (or nearly 0.0) and use more lights to "fill" in the object from different directions. It is common when lighting for visual effects to have your primary lights that are the key, rim and fill and then complement those lights with a set of bounce lights from the ground and any nearby objects. This can produce very convincing results in the hands of a skilled lighting artist.

But in the case of rendering large flat objects with some small details (like a building), none of the previously mentioned techniques give desirable results. The subtleties that we are used to seeing in real life which include the darkening of surfaces when they are near convex corners and the soft blocking of light from nearby objects are missing and the visual miscues are obvious.

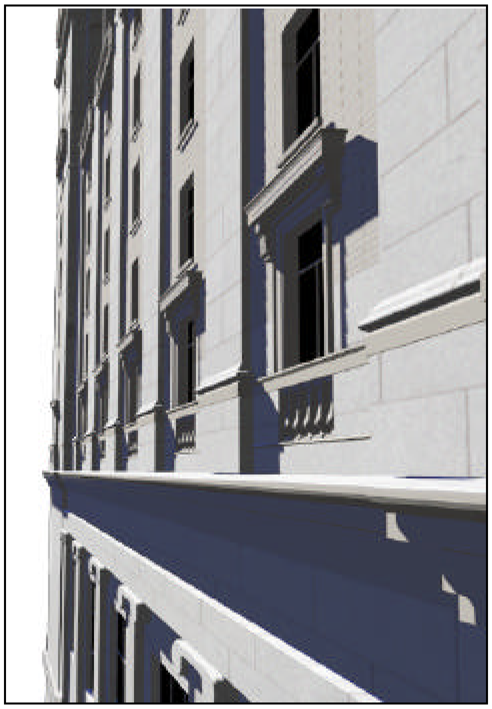


Figure 13 - Building with texture, key light, and constant ambient

The "Ambient Occlusion" technique that we used for *Stuart Little 2* gave us an accurate and controllable simulation of the ambient contributions from the sky and the ground on our computer generated objects. The technique, in concept, consists of placing two large area lights, one representing the ground and the other for the sky, into the set and accurately modeling the contribution of these lights on the object of interest.

These calculations are most convenient to implement with ray-traced shadows and area lights so we chose Exluna's Entropy renderer to generate our "Ambient Occlusion" information. The setup simply consists of two area lights (implemented as solar light shaders) and a very simple surface shader that samples those lights appropriately, taking into account the objects self-shadowing with ray-tracing.

```
Surface srf_diffuse_pref (
  Varying Point Pref = (0,0,0);
  float DiffuseFalloffAngleEnd = 90;
)
{
  point PP = 0;


  vector NN, NF;

  /*  Initialize  */

  PP = Pref;
  NN = normalize(Du(PP)^Dv(PP));
  NF = NN;

  /* Light Calculations */

  Ci = 0.0;

  illuminance (PP, NF, radians(DiffuseFalloffAngleEnd))  {
      LN = normalize(L);
      Ci += diffuse(NF);

  }
  Oi = 1.0;

}
```

Given enough samples into the area lights, this process generates very smooth soft shadows and a natural falloff of light into the corners of the object.

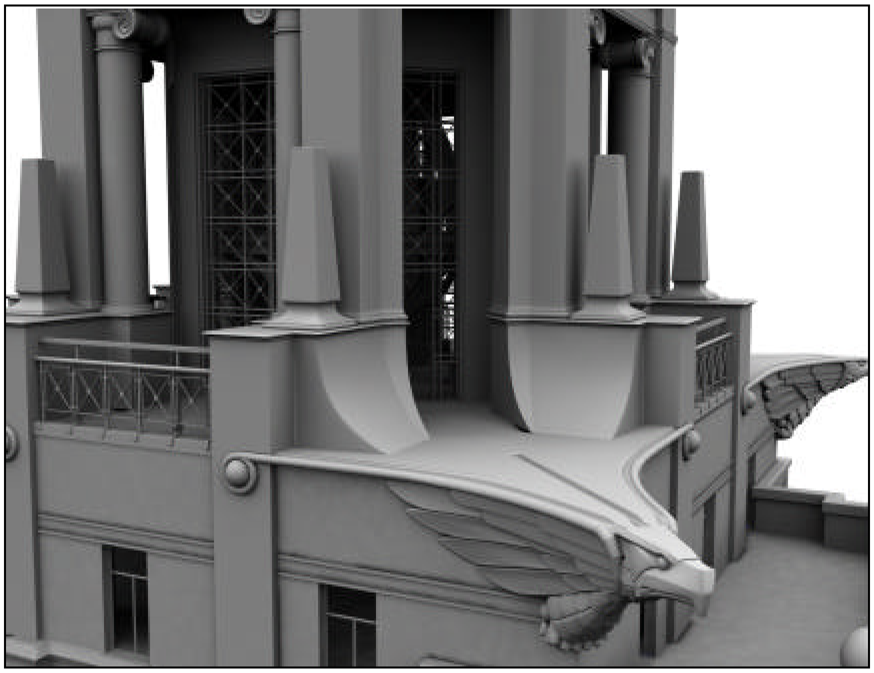


Figure 14 - Test rendering of Pishkin with "Ambient Occlusion" lighting only

In the case of the Pishkin Building for *Stuart Little 2* these lighting calculations can be made once and then stored for all future uses of the building since the building would not be moving or deforming over time. This was accomplished by pre-calculating the "Ambient Occlusion" pass and storing it as a series of textures – one texture for each patch or group of polygons on the building.

The Renderman implementation of this pipeline consisted of creating a rib for each patch on the building. This rib consisted one patch and a shadow object. In the rib generation, the patch's "P" coordinates were moved to be located directly in front of the camera with it's coordinates normalized to the screen space of the camera so that the patch completely filled the view. The unmoved vertices of the patch were stored as "Pref" data for shading purposes. The entire building object was stored in the rib as a shadow object.

The shader than ran the shading calculations on the "Pref" geometry which took into account the shadow object of the building and returned the results to "Ci" which effectively generated a texture map for that patch.

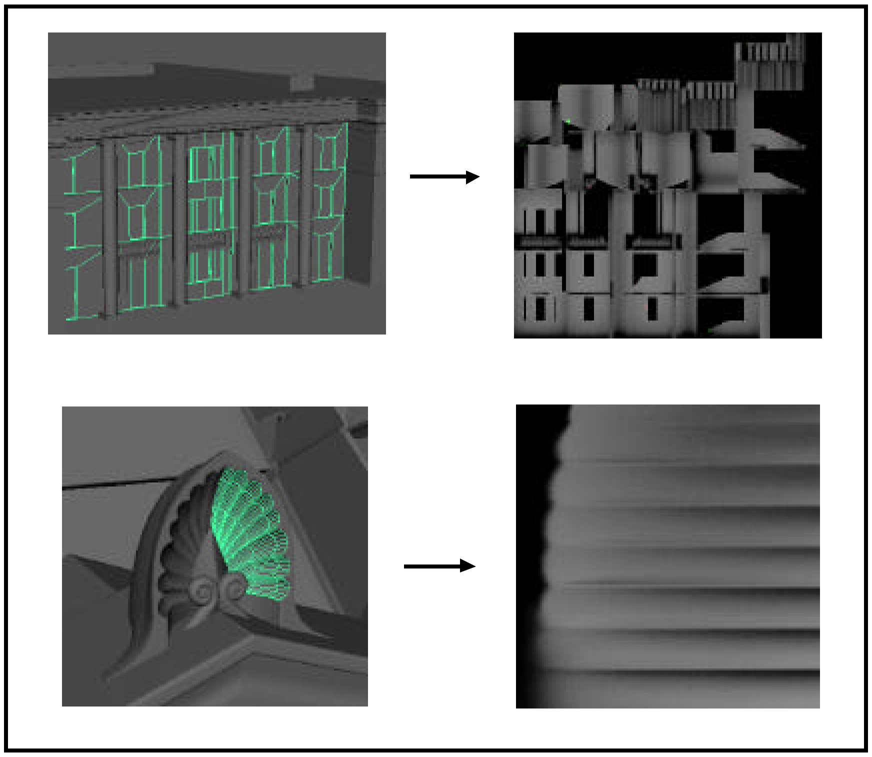


Figure 15 - "Ambient Occlusion" texture maps for 2 sections of the Pishkin

In our implementation we generated a separate set of texture maps for the sky dome contribution and the ground bounce contribution and then dialed the two new ambient levels in the shading for the building at the final rendering stage. If the sky was more blue or more overcast for a shot, the sky dome contribution color could be changed on the fly at the final rendering stage without performing any expensive calculations.

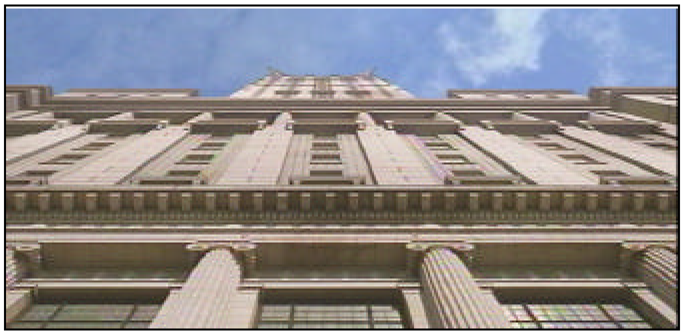


Figure 16 - Final version of the Pishkin as seen in *Stuart Little 2*

The end result was a more physically accurate shading model for ambient lighting which was both controllable and efficient since the more expensive calculations could be executed once and saved as texture maps.

1.2.2 Lighting

There are several things that provide input and reference to the lighting process including both on set reference and the creative process that is controlled by the director and the visual effects supervisors. In addition there are the technical aspects that are worth mentioning here including the lighting color space and keeping objects from clipping when they get too bright.

1.2.2.1. Reference balls on the set

One of the first things we inspect when starting a new visual effects shot are the reference balls that are shot on set. For each setup, the on set visual effects team clicks off a few frames of a 90% white sphere, a 50% gray sphere and a chrome sphere. These spheres can be visually inspected to tell you a few things about the photography and the lighting setup on stage.

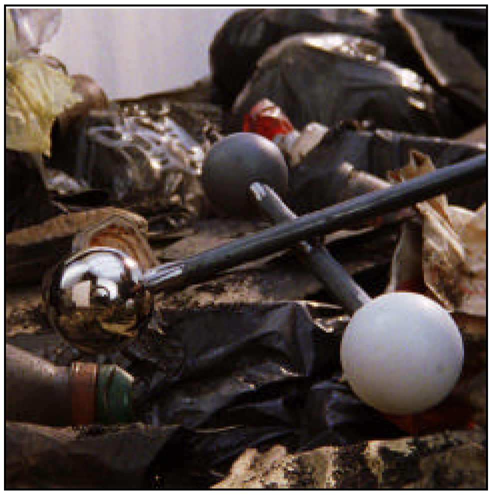


Figure 17 - Reference orbs photographed on set

The 90% white sphere and 50% gray sphere are very useful to see the general lighting directions of the keys and fills and the relative color temperatures. Very soft lights and bounce cards will occlude more softly and point lights will have sharper falloffs. The relative warms and cools of the various lights can also be seen on these objects.

The chrome sphere is perhaps the most interesting of the three because it actually contains in it's reflections a map of the entire shooting environment (missing only what is directly behind the sphere). From this reference you can pinpoint with some degree of accuracy the direction from which the lights are illuminating the set and even relative sizes of bounce cards and color temperatures. If a C.G. sphere is matchmoved to the on set sphere, software can unwrap the sphere and give you an accurate reflection map for the set.

1.2.2.2. Lighting cues from the photography

The next items to get serious attention when the lighting begins on a shot are the cues from the plates themselves. Generally, the plates are carefully inspected by the lighting artist for any cues that help to determining how the characters or objects should be lit to fit into the plate. Shadow density and color, location and orientation are all things that are checked for. Highlight and specular features are also used to provide both spacial and color reference for the lighter. The plates are also examined for any shiny objects that should catch a reflection of our character or bright objects that should bounce some light onto the subjects.

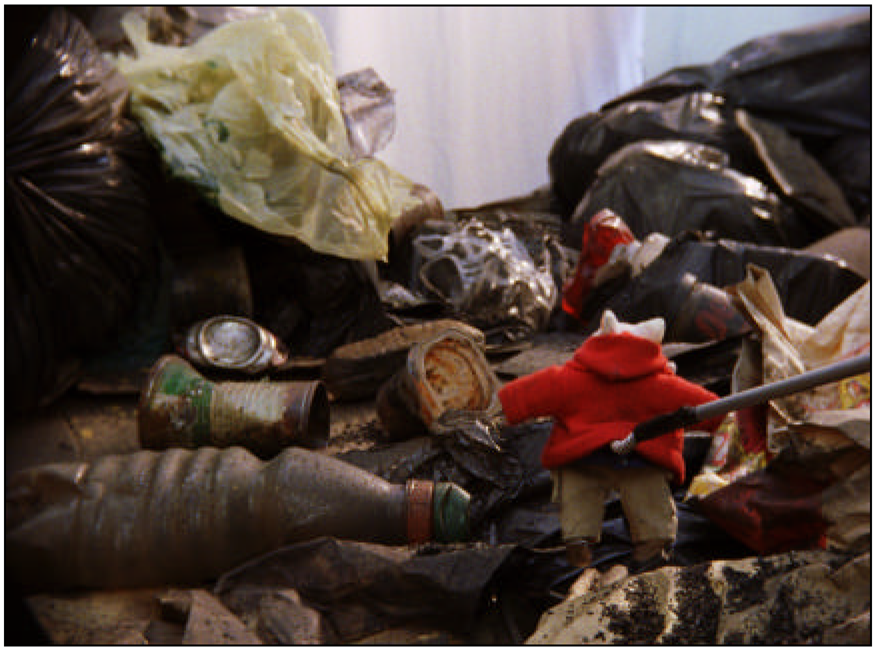


Figure 18 - Reference material shot on set with Stuart "stand-in" model

Basically, common sense combined with careful observation are used to dissect the photography and setup a computer lighting environment that matches the set as closely as possible.

1.2.2.3. Creative input

Once the character and objects are basically integrated into the live action photography, the creative process can really begin. This is probably most important aspect of the lighting process and it can be provided by a number of people on a film including the director, visual effects supervisor, or computer graphics supervisor and the lighting artist themselves. The goal is to enhance the mood and the realism of the film by either reinforcing the set lighting or breaking the "rules" a little and adding a light that wouldn't have been possible on the set.

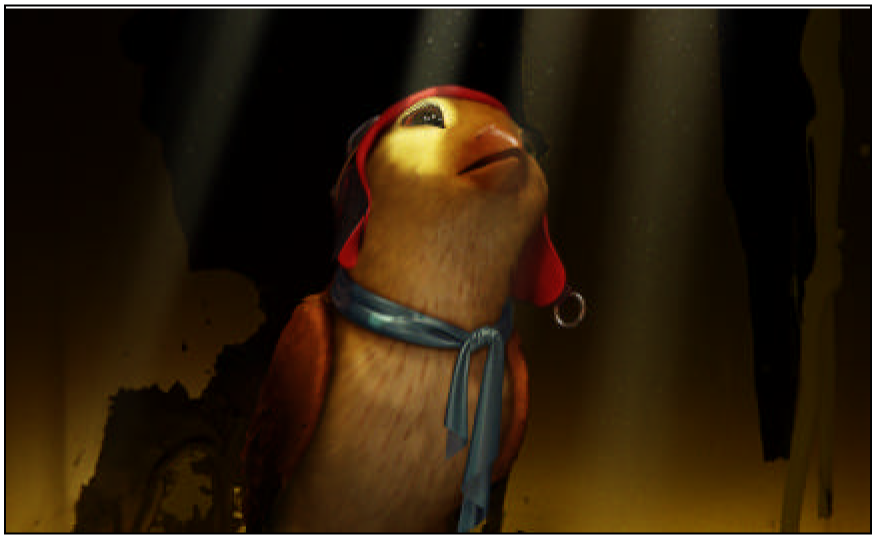


Figure 19 - Margalo in a can illustrating the use of "creatively" driven lighting

For Stuart, the creative decision was made that he should have characteristic rim light with him at all times. In some scenes, the lighter needed to be resourceful to find an excuse in the plate to rim light the mouse to fit his heroic character. Since we had plenty of control over the computer generated lighting, we could also do tricks like rim light only Stuart's head and top half of his clothes and let his legs and feet more closely match the photograph of the plate to accomplish both the creative tasks and the technical requirements at hand.

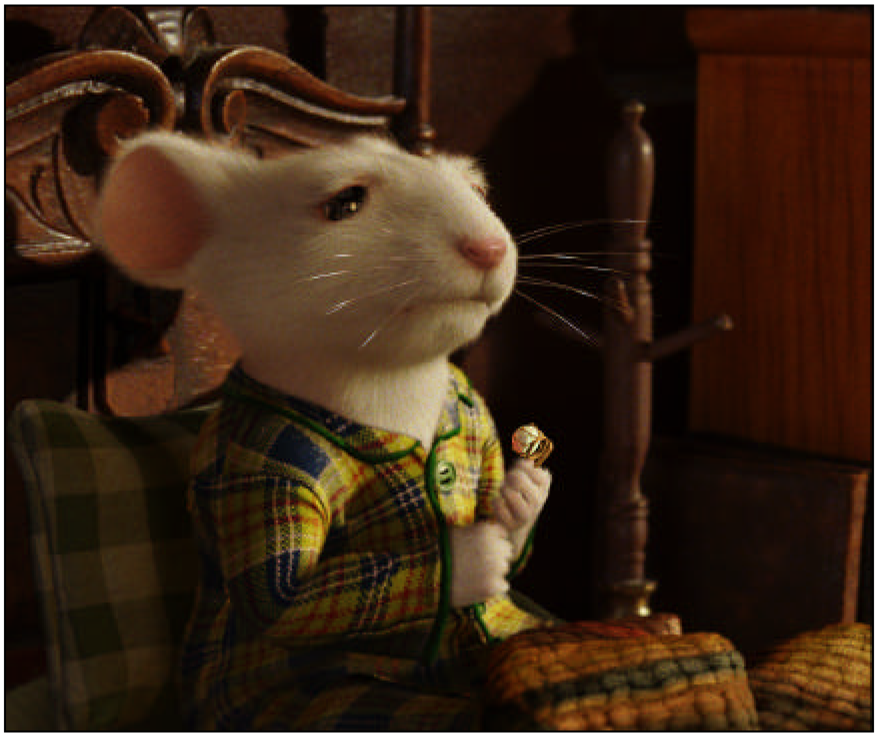


Figure 20 - Rim lighting in all environments for Stuart

Backlighting on the falcon was another opportunity to use lighting to an artistic advantage. Because the feathers on the falcon had separate backlighting controls in his wings and various areas of his body, the lighting could be used to accentuate a performance.

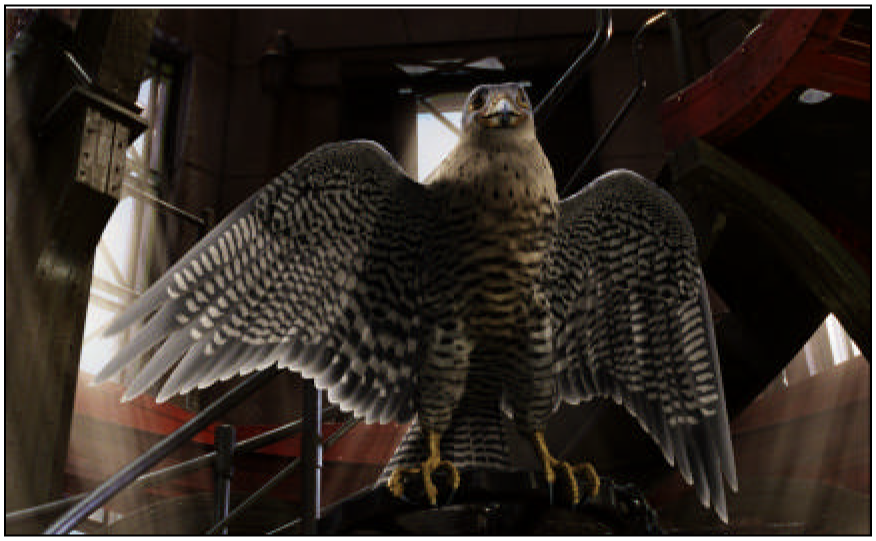


Figure 21 - Backlighting on the Falcon's wings

1.2.2.4. Color space issues

The topic of lighting color space is one of much debate and concern on each show that I have been a part of over the years and for good reason. The color space in which you choose to light controls the way that the light will fall from bright to dark and every value in between. It is also a complicated subject that could take many pages to cover in a thorough manner so the attempt here is to discuss the topic in a brief and pragmatic way that will perhaps encourage you to investigate further if your interest is piqued.

There are two significantly different color spaces that we will introduce briefly and then discuss how these color spaces work in production.

1.2.2.4.1 Linear Color Space : Gamma 2.2

Gamma 2.2 is also referred to as "linear color space" because, presuming a correctly calibrated display device, doubling the intensity of a pixel value actually doubles the amount of energy coming out of your monitor or bouncing off the screen at the front of a theater. This makes computer graphics operations like blur and anti-aliasing correctly preserve energy.

For instance, if you take 1 pixel with a value of 1.0 and spread it out over 2 pixels, both pixels will contain values of 0.5. If you sample the energy coming off of a calibrated display device in both the before and after case, the energy should be the same. This is because you are working in a linear color space.

The confusing thing about working in a linear color space is that our eyes are not linear in the way they perceive changes in intensity. For example, if you look at a 50-watt light bulb and compare it to 100-watt bulb, the 100-watt bulb will appear brighter. But when you compare a 100-watt bulb to a 200-watt bulb the relative difference in brightness will not be as great as the difference between the 50 and 100-watt bulbs. This is because our eyes are more sensitive to changes in dark values than in bright values.

So, when you are working in a linear color space, you find that in order to get a pixel value that looks visually to be about 50% gray you need to use a value of approximately 22%. This results in heavier use of the lower range of color values and necessitates rendering at least 16 bit images to preserve color detail in the lower range.

It is interesting to note that a point light illuminating a sphere from a long distance when viewed at gamma 2.2 looks just about right when compared to a real world experiment. This is because a simple lighting model correctly models the falloff of a light in linear color space.

All of these are good reasons to light computer graphics in a linear color space.

1.2.2.4.2 Logarithmic Color Space: Cineon

The color space defined by the Cineon format is another very useful color space. Rather than attempting to produce a space that is mathematically linear, this space is designed around the response of film negative to an exposure. Each time the amount of exposure is doubled, the Cineon code value is increased by 90 points. Since Cineon files are stored with 10 bits of data per channel, there is enough room to store the entire useful range of the negative in the Cineon format file.

This is a very useful format for color correction and other techniques that rely on a direct correspondence to the response of film. For instance, if you wanted to preview what a shot would look like printed a stop brighter (effectively twice as bright), if your image is in logarithmic color space you simply add 90 Cineon code points to the image.

1.2.2.4.3 Workflow

How does this work out in production? In our production environment we scan and color-time our images in the Logarithmic color space using Cineon files. All of our computer-generated objects are lit in linear color space with a gamma of 2.2 and stored as 16 bit linear images. In the compositing stage, the background plates are converted to linear color space, the composite is done in linear color space, and then the images are converted back to Cineon files to be recorded back to film.

1.2.2.5. De-clipping

Now that we have introduced the issues of color space and color space conversion, we can do something truly useful with it. A problem that is often encountered in computer graphics rendering is making colorful objects bright without letting them clip. Particularly when you are lighting in a linear color space, in order to double the brightness intensity of an object, you have to multiply it's color by a factor of 2. If your object already has a red value of 0.8 and the green and blue channel are each 0.4, the red channel is going to clip when the brightness is doubled to a value of 1.6. When you have deeply saturated objects like a bright yellow bird, these issues show up with great frequency.

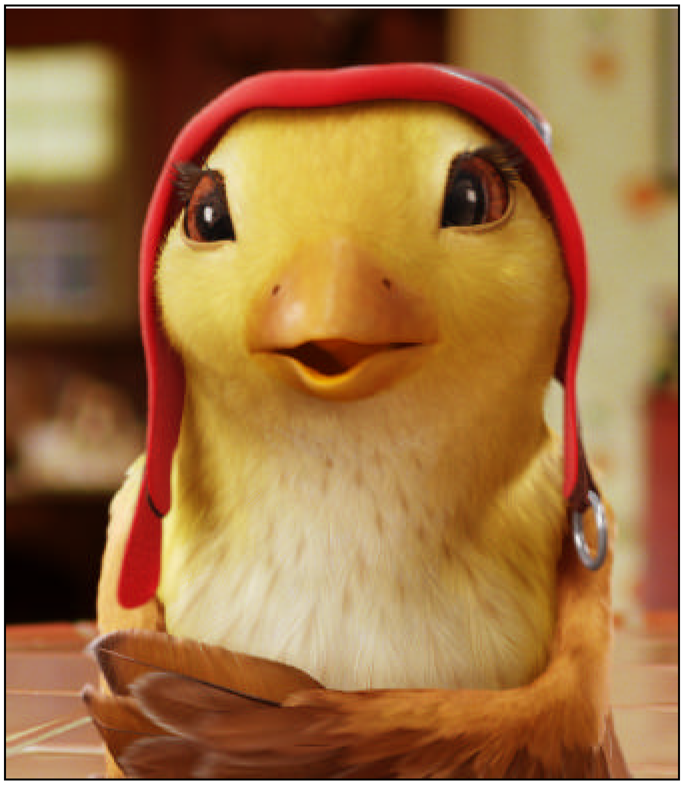


Figure 22 - Margalo's bright yellow color would tend to clip under high intensity lights.

The solution is to use a logarithmic space to handle the brightening of the objects that are likely to clip. Our implementation was written into the Renderman shaders. Since shaders store colors as floats they can exist outside of the 0.0 - 1.0 range. Our de-clip shade op was fed floating point color and returned a "de-clipped" version of that color by darkening the color by an automatically determined multiplier, converting that color to logarithmic space, and then adding the appropriate number of Cineon code points back to the color to preserve the original intensity. This results in a color that smoothly desaturates as it gets over-exposed and has a very filmic look.

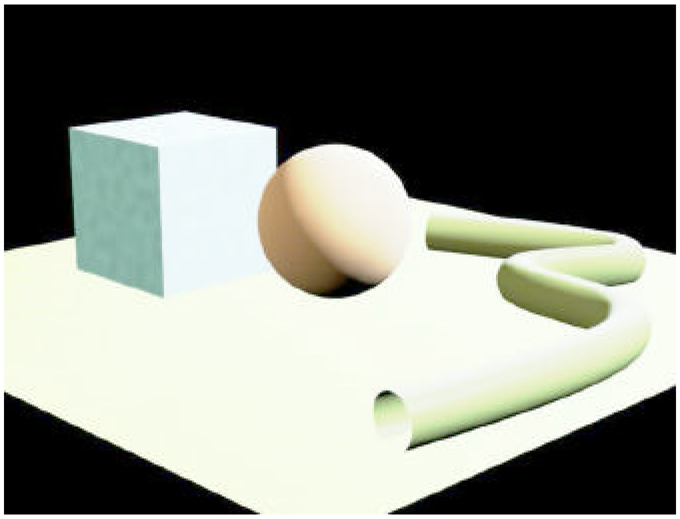


Figure 23 - "Declip" test which shows colors naturally desaturating with intensity

1.2.2.6. Multiple Passes

In order to have more control over both the rendering and the compositing process, it is common to render out many passes for a single character and each of the props in the scene.

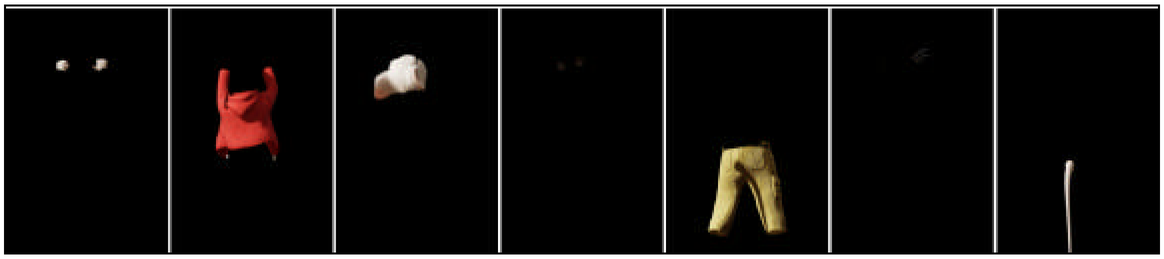


Figure 24 - From left to right, hands, jacket, head, eyes, pants, whiskers and tail

1.2.2.6.1 Separate shadow maps for each "type" of object

Not only do we render separate color passes for each type of object, but we also break out our shadow maps into lots of passes. For instance, having one shadow map for Stuart's head and a separate map for his clothes gives the artist more control over setting the blur and bias controls for each. Because the fur is not a hard surface, the blur and bias controls need to be pushed slightly higher than for the clothes shadows to avoid objectionable and noisy self-shadows.

1.2.2.6.2 Cloth beauty pass

The cloth is usually rendered on it's own. Usually the different layers of cloth can be rendered together but sometimes they are also broken out into layers.

1.2.2.6.3 Character skin/fur beauty pass

The character's skin, fur and feathers are rendered together without any other elements. Since this is generally the most expensive of the render passes, the goal is to only render these elements once or twice if possible. The skin and fur is rendered with the eyes and costume as a hold out so that everything else can simply be layered behind in the composite stage.

1.2.2.6.4 Eye beauty pass

The eyes are rendered by themselves and placed behind the skin pass.

1.2.2.6.5 Cast and contact shadow passes

For each character we have several different types of shadow passes that are rendered out. Each character has near and far contact shadows which are generated from a depth map rendered from below the character and then projected onto the surface on which the characters walk. Each character also has various cast shadow elements that are rendered from the key lights and projected from those lights onto the set geometry.

1.2.2.6.6 Reflection passes

If there are shiny objects in the scene, a reflection camera is set up to render the characters from the perspective of the reflective object.

1.2.2.7. Tile-rendering when needed

Even with improved processing power and increases in memory configurations (our rendering computers have 1 gigabyte of ram per processor) there are always some frames that just won't render. In the cases where it is related to memory usage, we break the rendering problem down into tiles so that Prman can work on just a section of the image at a time.

In some cases where the character or prop is motion blurring right by the camera, we may break a single frame into more than 200 tiles to split up the job on many processors and reduce the memory consumption. Once all of the individual tiles have been completed on the render farm, the complementing scripts then re-assemble the tiles back into a complete image and the frame is finally complete.

1.3 Post-Renderman

In visual effects for live action, about half of the work is done in the lighting stage, and the other half takes place in compositing. The various passes need to be combined and massaged into the plate. We will detail here several of those 2d techniques which are widely used to integrate rendered images onto live action photography.

1.3.1 Compositing

1.3.1.1. Black and white levels/Color balancing

One of the first things that is done at the compositing stage is to make sure that the computer rendered images conform to the plate in the area of contrast and color temperature. Because the compositing tools are both powerful and provide quicker feedback than rendering the complicated geometry, the compositing stage is often the best place for these minor adjustments.

When composting a shot, one quick check is to see that the computer generated portion of the image do not get any darker or brighter than any of the cues in the live action photography without good reason. There can be exceptions but in general, the CG elements will pop if they don't match the contrast range very closely.

In the case that a particular light begins to feel out of balance with the others, it may require going to back to the rendering stage to make an adjustment but in general, with a number of compositing tools the color balancing can be accomplished efficiently.

1.3.1.2. Edge treatments

Almost nothing in real life is as crisp as what can be generated in the computer. Lines are generally smooth and nicely anti-aliased and geometry tends to be a little crisper than what we see in the real world.

In order to match the CG to the plate we treat the edges of the geometry carefully. Before composting we extract a matte of all of the edges of all of the CG elements.

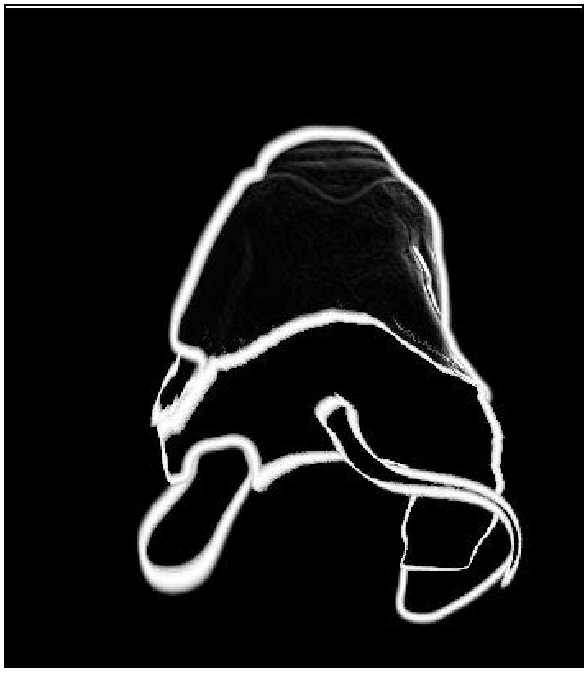


Figure 25 - Stuart's "Edge Treatment" area as generated in the composite

Then, once the CG elements are composited over the live action background, then we use that "edge matte" to soften the image inside that matte only to blend the artificial elements into the photography.

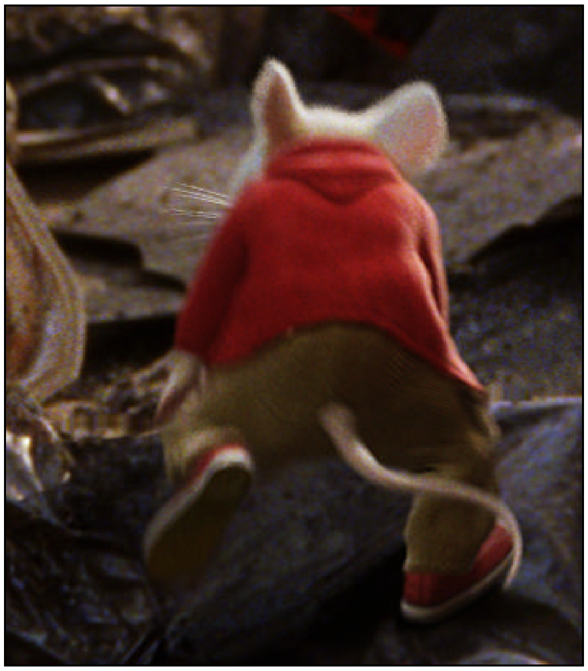


Figure 26 - Stuart as seen in the final composite with edge softening

1.3.1.3. Film grain

Every film stock has it's own characteristic grain structure that introduces some noise into the scan. This noise actually has it's own response curve and shows up at different intensities depending on the level of exposure. Generally a film is shot on one or two different types of stocks so for each show we dial in our grain matching parameters to add grain to our synthetic elements that should closely match the grain in the plate. Each compositor also adjusts the grain parameters on a shot by shot and sometimes an element-by-element basis as needed to match the individual plate.

1.3.1.4. Lens warp

Even with advances in lens technology, the wide lenses used for today's feature films are not perfectly aspherical. They slightly warp the image near the corners of the frame and since the cameras used in computer graphics are perfect perspective projections the images will not register perfectly in the corners of the frame.

This is most noticeable in the case where you have long straight lines that are represented both in the live action plate and in the CG elements. In this case, a 2d image warp which mimics the behavior of the live action lens can help register the two more precisely to each other and solve the problem.

1.3.1.5. 2d contact shadows

One of the most useful tricks in the composting steps is the ability to add little contact shadows between elements. Often, because of the blur or the bias settings on a shadow from a light, or simply because of the positioning of the lights in the scene, you don't get a little dark shadow from one layer of the cloth to another or from the cloth around the neck to the neck's skin and fur.

Fortunately, since the objects are already rendered as separate elements, generating a contact shadow between two elements is straightforward. The matte of the "top" element can be offset and used to darken the color channels of the "below" element. If needed, the offset can be animated and the color correction for the shadow can then be dialed in to taste.

1.3.1.6. Holdout "gap" filling

All of these separate elements give lots of control at the compositing stage but do come with one drawback. Because the "expensive" renders of the skin, feather and fur are rendered with 3d holdout objects or matte objects and the other renders are not, when you composite those images "over" each other and they move significantly, the motion blur can cause gaps between the elements.

These gaps come from the fact that what we're doing is a cheat. To be correct, each of the elements should be rendered with all of the overlapping elements as a matte object and then the various passes should be added together to form a solid matte and perfectly colored RGB channels.

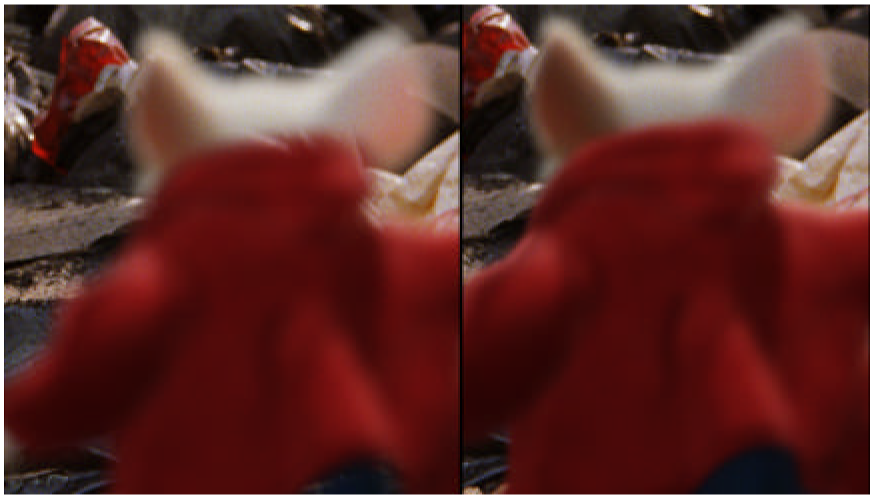


Figure 27 - Before (left) and after (right) the "gaps" have been filled between the shirt and head

This is possible but not very cost-effective when you are talking about increasing the processing resources by a factor or 4 to accomplish such a workaround. In most cases it was adequate to find the gap areas by pulling a matte of the gray areas of the matte and finding where that overlaps the background objects. Then the fringing (whether it's showing up as light or dark fringing in a particular shot) can be color corrected to not be noticeable.

1.3.2 Film Recording

Once the final composite is completed, the shot is recorded back onto film with a laser recorder and the negative is processed, a work print is generated and the work print can be screened generally the next day to evaluate the work and hopefully final a shot!

1.4 Conclusion

We have discussed the use of Renderman in the context of a live action feature film with examples from the film *Stuart Little 2*. It is hoped that this introduction will serve as a jumping off point for of the topics discussed.

It should also be noted that a team of very talented artists are behind the examples and images presented and much credit is due them for the look and content of these notes.