

Keying and compositing in NukeX

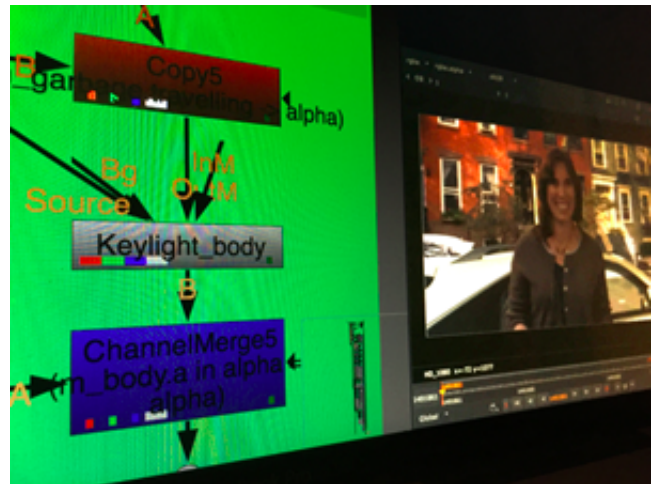
Skitch a ride up its steep learning curve using my template and tutorial

By [Don Starnes](#)
02.15.16

You've seen it before: composited people with hair helmets. If you've done any keying yourself, you probably notice it all the time. And you've seen worse: webbed fingers, grey outlines, green, magenta or grey reflections on skin. Perhaps you see these in your own work. How can you make your keys look as good as the ones in, say, *Iron Man* or *Transformers*?

Good cinematography is obviously a great place to start. However, even a perfectly photographed green screen shot can be difficult to key: frizzy hair, shiny skin, semi-transparent or reflective surfaces are common. In fact, pretty much anything that you'd want to photograph is inherently challenging.

If you have a compositing project and want the best quality, my advice is to hire a professional compositor. They don't call these people artists for nothing: great compositing requires a trained eye, a lot of experience and good taste. The software is a distant second in importance.



Perhaps, however, you are a mere mortal, stuck with the compositing chores on an under-funded movie, wondering if there is any way to improve your keys. That was me, recently, working on *#LookUp*, my short, no-budget passion project. I photographed the movie using an inexpensive camera and compressed video (never my first choices when shooting green screens). It was essential that the movie seem realistic, but I wasn't happy with my compositing. How could I retain the actress' fine, windblown hair?

I googled and googled; I watched tutorials and got advice from seasoned professionals. I learned better technique that made a huge difference in my compositing.

The Three-Pass Method

Initially, I did my keying and compositing in Final Cut Pro (7 and X), After Effects or Resolve. All of these have fine keying tools; some of them have automated tools that practically make the key for you. No matter what software you use, though, some part of the image is often stubbornly resistant to the algorithms' best efforts. A solid key of the body makes an artificially hard edge. Adjustments often don't help much in getting a good overall key. Even if the green

screen is perfectly even, different parts of the foreground image have different properties: therefore one-pass keying is almost always insufficient.

Friends don't let friends use one-pass keying. Following treatises by [Mark Christiansen](#) and others, I started making three basic, overlapping mattes for each key:



- **edge**, which is focused on preserving the sometimes semi-transparent pixels of the delicate edge
- the **core**, which hammers out a solid body for the object but excludes the edge
- a **light wrap**, which wraps a bit of the background around the edges of the foreground (in the way that light refracts around objects in real life)

In addition, I divided any problem areas of the image into separate mattes. For example, I'd make a matte for the hair portion of the foreground image if it required different keying than the body.

Although my skills were, shall we say, developing, I was able to get significantly better composites using these professional methods. Still, I was dissatisfied: despite my best efforts, the edges were a bit noisy and I couldn't preserve as much hair detail as I'd wished.

Expensive software is suddenly free

I was almost ready to stop there. Then The Foundry announced that they were releasing a non-commercial version of their £4,800 industrial-grade compositing software— for free.



NukeX is a descendent of code originally developed in-house at Digital Domain. It has since been used at Disney, DreamWorks, Sony and ILM, among other places. It is one of The Foundry's suite of applications, which includes two other versions of Nuke (NukeX has more features than Nuke and fewer features than Nuke Studio).

I wondered if Nuke was more capable than the apps I'd been using and if its node-based approach would be easier to use and more effective. I found both answers to be yes— after a steep learning curve.

Nuke was designed to be used by trained artists at professional effects companies with in-house engineering staff. Keying and compositing are essentially doing mathematics on pixels' data values. Where After Effects attempts to conceal fundamental compositing concepts behind apprehensible metaphors, Nuke rather expects you to have some understanding of these precepts. Where After Effects automates some compositing

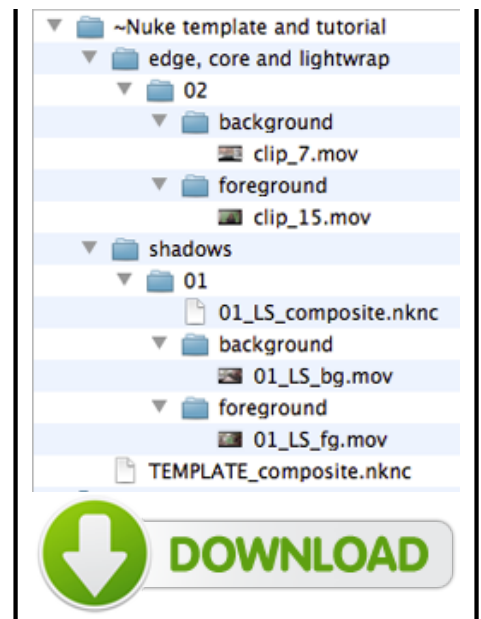
My NukeX script file and sample shots:

tasks for you, Nuke requires you to properly handle certain things manually.

Plus, Nuke is buggy. This is ok if you have engineers who are going to customize the code anyway. For the indie filmmaker: not so much. I expect that The Foundry has released Nuke 9 into the wild so that we can help them improve the code for Nuke 10, which is coming out in the next few weeks.

Nuke required me to be a better compositor. Once I put in considerable time and came to understand NukeX, I found it to be far better at compositing than the other software I'd been using.

In this tutorial, I'll share my NukeX keying and compositing template with you and explain how to use it with sample footage.



<http://donstarnes.com/pub/NukeX/>

Follow the instructions on your seat-back card

Resources

NukeX:

[Nuke product page](#)

Some good places to start learning Nuke:

[Step up to NUKE](#)

[Nuke for After Effects Users](#)

[Nuke in Production Series: Getting Started](#)

[Nukepedia](#)

The finished movie that I composited using this NukeX script: [#LookUp](#)

Terms

- The non-commercial version of Nuke uses compiled composite files (the commercial version can use text script files). Although this template is compiled, I'll refer to this template as a "script" (because that's how others refer to Nuke's files).
- The principal photography background surface is referred to herewith as a "green screen"; it can be a blue or red screen as well, or whatever contrasts with the foreground

element.

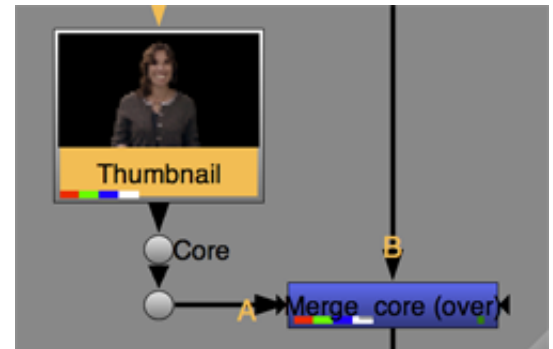
- I call the foreground element, well, the foreground. I call the background plate the background.
- Despill means to replace the green screen colored pixels with other pixels.
- You can add an optional clean plate to the script, which is a shot of the green screen without the foreground element.
- I refer to the script's edge, core, light wrap and shadow elements as "layers" (despite Nuke not being a layer-based compositing software).



A clean plate is a happy plate.

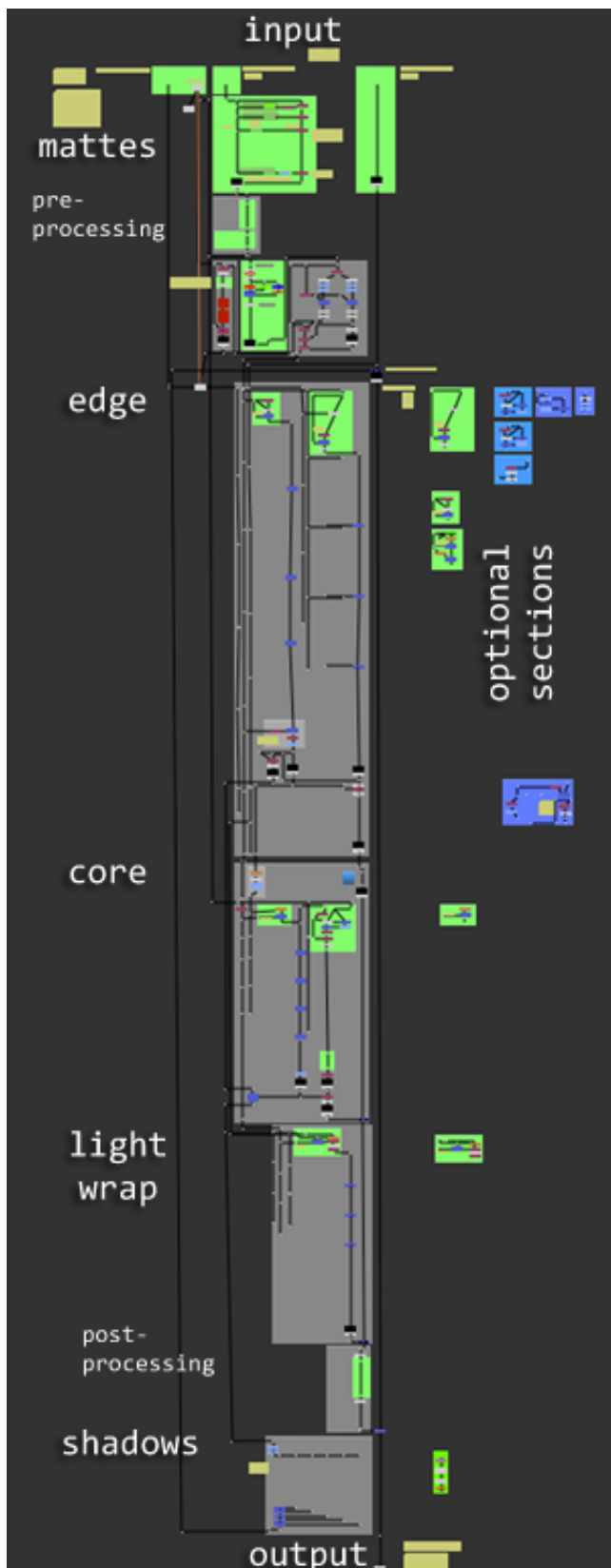
Concepts and requirements

In Nuke's node-based approach, image data flows through branches (called pipes) in a node tree. In this template, the image data enters at the top (**input**) and flows to the bottom right of the tree. Importantly, this template observes Nuke's primacy of the B pipe. Many Nuke nodes mathematically combine one image (routed in the node's A pipe) with another (routed in its B pipe). A node can get confused if you, say, try to superimpose an image from the B pipe over the A pipe.



My **TEMPLATE_composite.nknc** script is designed for general-purpose green screen compositing of a single shot.

- Node tree fragments are grouped together into "sections".
- Green colored sections are things that you must adjust for each shot.
- Grey colored sections contain fairly automatic functions.
- The edge, core, light wrap and shadows sections are layered over the background.
- The script can use a clean plate if you have one, but will generate one if you don't.
- To the right of the node tree are optional sections (in green and blue) that can be piped in for keying, despill, etc.
- It uses mattes (in the Foreground_mattes section) to isolate different parts of the image; these mattes are stored in the script's matte channels.



- There is room in each section for working on multiple parts of the image using these mattes.
- Edge and core alpha mattes mask out the green screen.
- A despilled version of the foreground shows through these mattes (to make their respective layers).
- A traveling garbage matte masks out everything around the foreground image.
- The Shadows section can add shadows of the foreground image.



Nuke is said not to work very well with movie files, which are insufficiently designed and unreliable for serious FX work. In this tutorial, you'll convert your movie files to EXR image sequences.



I prefer to retain each shot's original frame numbers when I convert it to an image sequence (as opposed to renumbering each image sequence's first frame as 1). For example a shot from early in a take could have frames numbered 289 - 388 and one from later in the take could have frames numbered 1491861 - 1491960. Retaining these numbers enables me to find a sequence's original movie file, if necessary.

This tutorial uses

- Final Cut Pro 7 (for editing)
- After Effects CS6 (for media conversion)
- NukeX 9.0

although other editing and media conversion software should work.

For each shot, you'll:

- convert the media to EXR image sequences
- read the media into a copy of the NukeX script template
- adjust the script for the shot
- write the resulting composition to an EXR image sequence
- convert the image sequence into a movie file



Each shot should take at least 75 minutes to composite.



Compositing using this script is a long and complicated process. You have to want it.

Using the NukeX compositing template

In this tutorial, we'll composite



shot 1, a long shot, and



shot 2, a medium shot.

We'll do shot 2 first, as it is best for explaining the script's essential concepts. Its foreground and background shots are in the tutorial's **/~Nuke template and tutorial/edge, core and lightwrap/02/** folder:



Clip_15.mov- foreground



Clip_7.mov- background

Prepare the project

1. In the project folder on your edit drive, prepare folders for the project files and for each of the shots: 01 is for shot one, 02 is for shot two:

/After Effects/Renders/EXR to QT conversions

/Final Cut Pro/to After Effects/

/Nuke/Renders/[shot number]

/Nuke/to Nuke/[shot number]/background

/Nuke/to Nuke/[shot number]/foreground

/Project Files/After Effects/

/Project Files/Final Cut Pro/

/Project Files/Nuke/

The project files, which are all in one folder, can be easily and frequently backed up (hint hint).

2. This tutorial doesn't include a clean plate; if you had a clean plate, you'd make a **[shot number]_[description]** folder for each clean plate that you have in your project directory's **Nuke/to Nuke/clean plates** folder. For example, the clean plate folder for the first 4 medium shots would be called 02-05_MS.
3. Unsurprisingly, you'll keep all of your Nuke project files in the **/Project Files/Nuke** directory:



4. Import the tutorial shots into After Effects.

If you were compositing shots from a Final Cut Pro project, you'd instead:

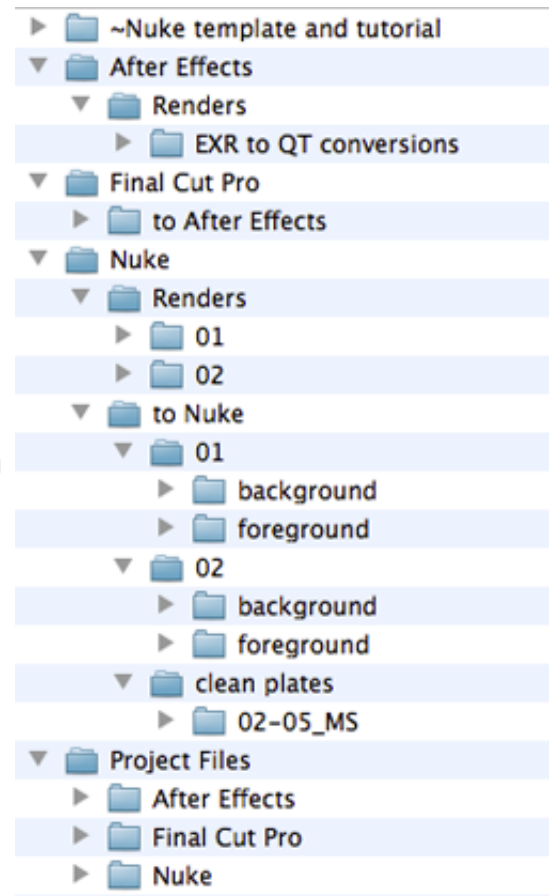
- a. Media Manage the footage to make a new clip for each shot's foreground and background.
- b. Export an XML and save these to the **/Final Cut Pro/to After Effects/** folder.
- c. Import the XML into After Effects.

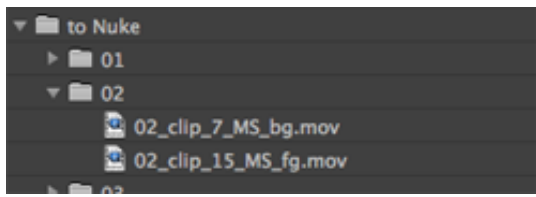
5. Convert each shot's media into EXR image sequences:

- a. Make a **to Nuke/[shot number]** folder for each shot in the After Effects Project pane.
- b. Move or copy the shot's foreground and background footage into the [shot number] folder; rename the foreground shot **[shot number]_[original file name]_[description]_fg.[ext]** and the background shot **[shot number]_[original file name]_[description]_bg.[ext]**. Make the description very short. For example, the Quicktime file for this shot, a medium shot, would be called 02_clip_15_MS_fg.mov.

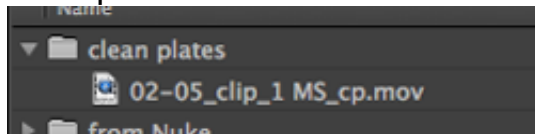


Don't use the # character in the file name; After Effects and Nuke use this character as a frame number variable in image sequences.



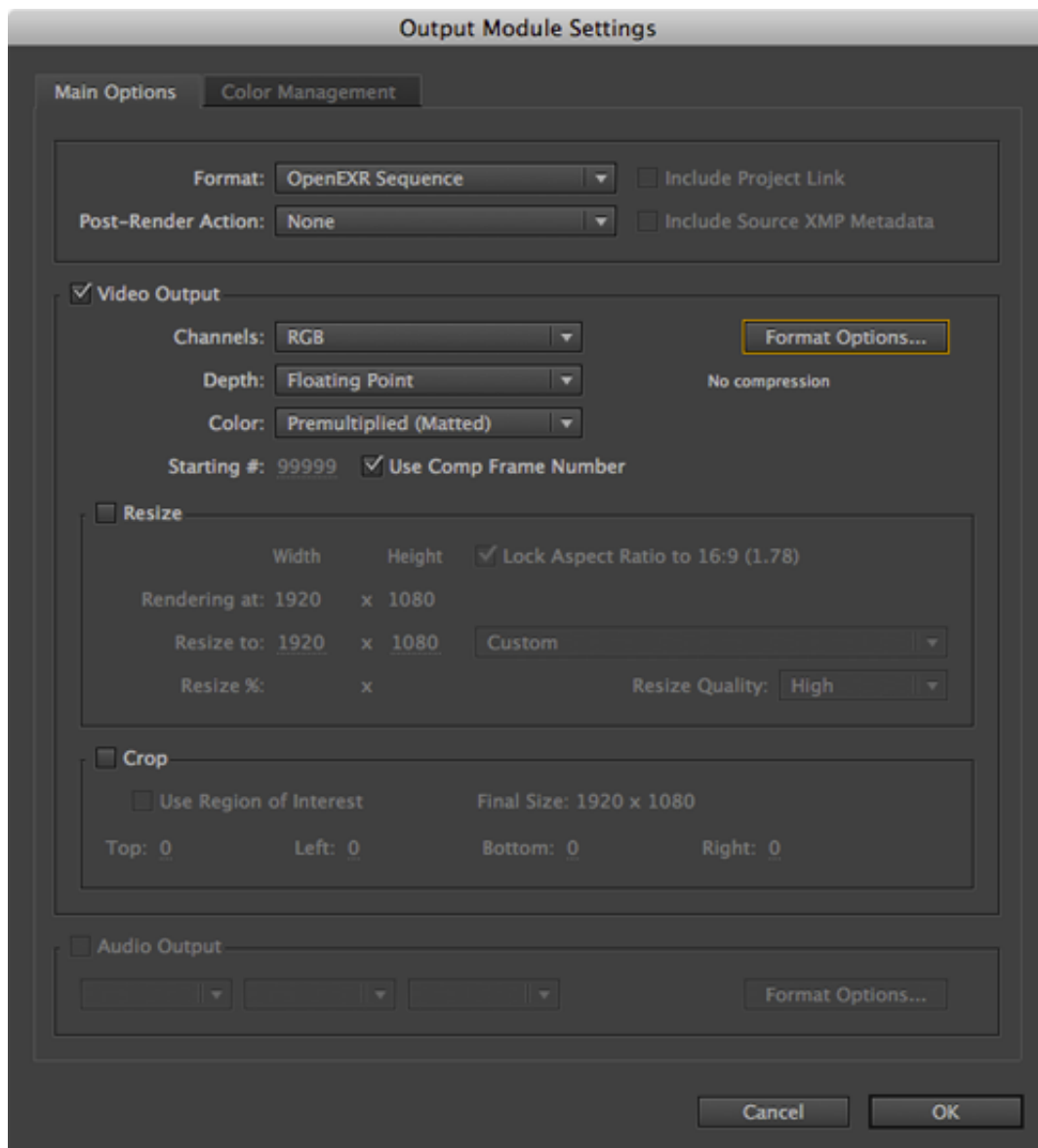


- c. If you had a clean plate for the shot (or a series of shots), you'd move it to a **to Nuke/clean plates** folder and rename the shot **[shot number]_[original file name]_[description]_cp.[ext]**. For example, the Quicktime file for the clean plate for the first 4 medium shots would be called **02-05_clip_1_MS_cp.mov**.

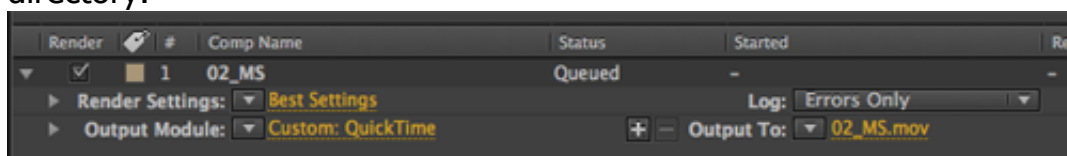


- d. Render the foreground footage:

- i. In Render Settings, confirm that the frame rate is correct (the **Use the comp's frame rate** setting should work).
- ii. In Output Module settings, set the
 - Format: **OpenEXR Sequence**
 - Format Options.Compression: **None**
 - Color: **Premultiplied**
 - **Use Comp Frame Number** (which retains the shot's original frame numbers)



- e. Set the Output to the project's **Nuke/to Nuke/[shot number]/foreground** directory.



6. Do the same for the background footage; set the Output to the project's **Nuke/to Nuke/[shot number]/background** directory.
7. If you had clean plates, you'd render the EXR sequences for each of them, setting the Output to the appropriate **Nuke/to Nuke/clean plates/[shot number]_description** directory.

Now that you have converted the shot's files to image sequences, you will composite the shot:

Create the shot's NukeX script

8. In your project's **project files/Nuke** directory, make a copy of the

TEMPLATE_composite.nknc Nuke script file.

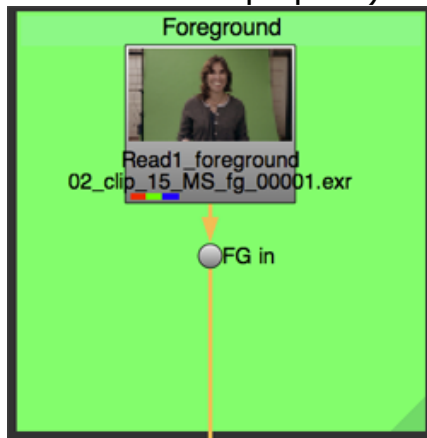
- a. Rename the copy **[shot number]_[description]_composite.nknc**. Make the description very short. In this example, name the script for this shot **02_MS_composite.nknc**.
- b. Move the project file to the **/Project Files/Nuke** directory.
- c. Open NukeX, then open the **02_MS_composite.nknc** script.



Nuke Studio has a visible timeline; Nuke and NukeX do not. Instead, each shot is placed on a virtual timeline. There is no dragging and dropping: each shot is read into the project with a Read node. It's an interface that only an engineer could love.

Because I've specified to retain each shot's original frame numbers when you convert its movie file to an image sequence (as opposed to starting them all at frame 1), you have to manually sync up each image sequence's "track" in Nuke's virtual timeline. It's done by specifying frame ranges:

9. In the **Foreground** section, add a Read node (press the **R** key while the cursor is in the Node Graph pane):



- a. In the Read node's properties, navigate to the

Let the first Read node control the script's frame range.

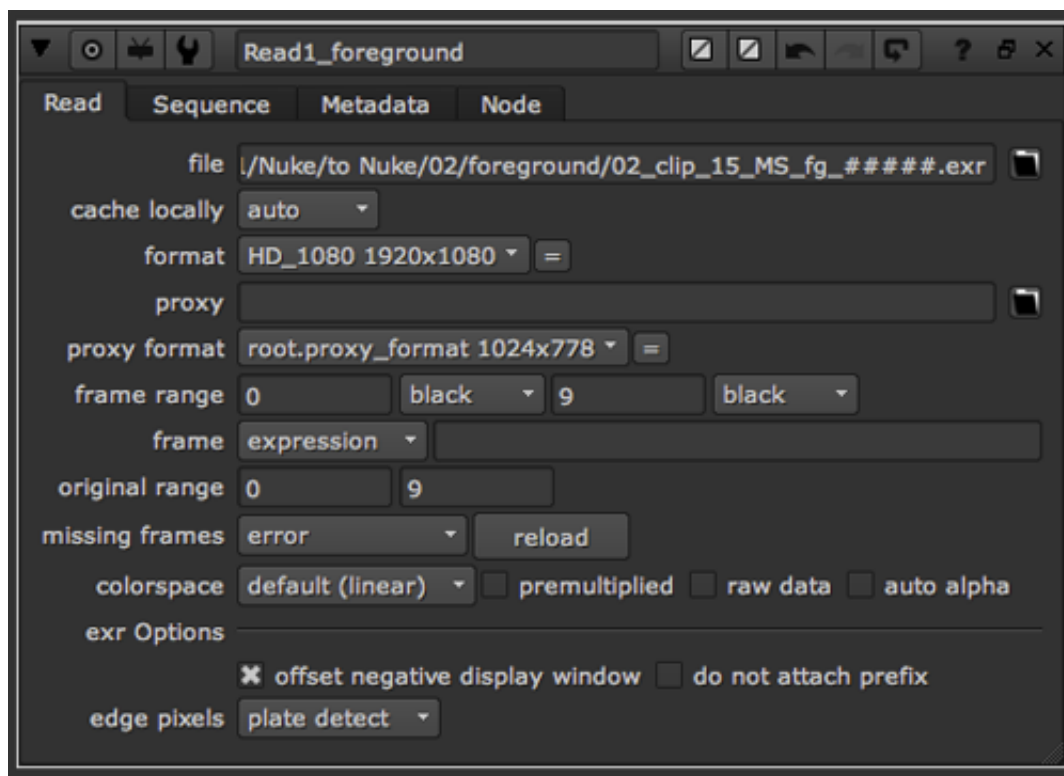
Nuke has a vast global virtual timeline, with frames numbered from, say, 1 to 1M. The project, Read and Write nodes and Viewer nodes all have to be set to use the same part of this timeline.

- Each Read node is like a "clip" on a "track" on this timeline. When the first Read node is read, Nuke places its frames on this timeline as they are numbered (for example, starting on frame 1491861 and ending on frame 1491960). As this "clip" is something that you want to work on, you can think of this frame range as a sort of "work area" for the script.
- When other Read nodes read into frame ranges outside of this range, they are outside of this "work area" and won't sync with the first "clip". The **frame.start at** setting tells the other Read nodes which frame in the global timeline to start playing at. For example, if the second Read node has frames numbered 289 to 388, the frame.start at.1491861 setting would tell this second "clip" to sync the first frame of its "track" with the 1491861 - 1491960 "work area" defined by the first Read node.

- Similarly, the project must be set to use this **frame range** in the **Project Settings**.
- Each Viewer node must be set to view this **frame range** in its properties.

/Nuke/to Nuke/[shot number]/foreground/directory to set the file to that shot's foreground image sequence.

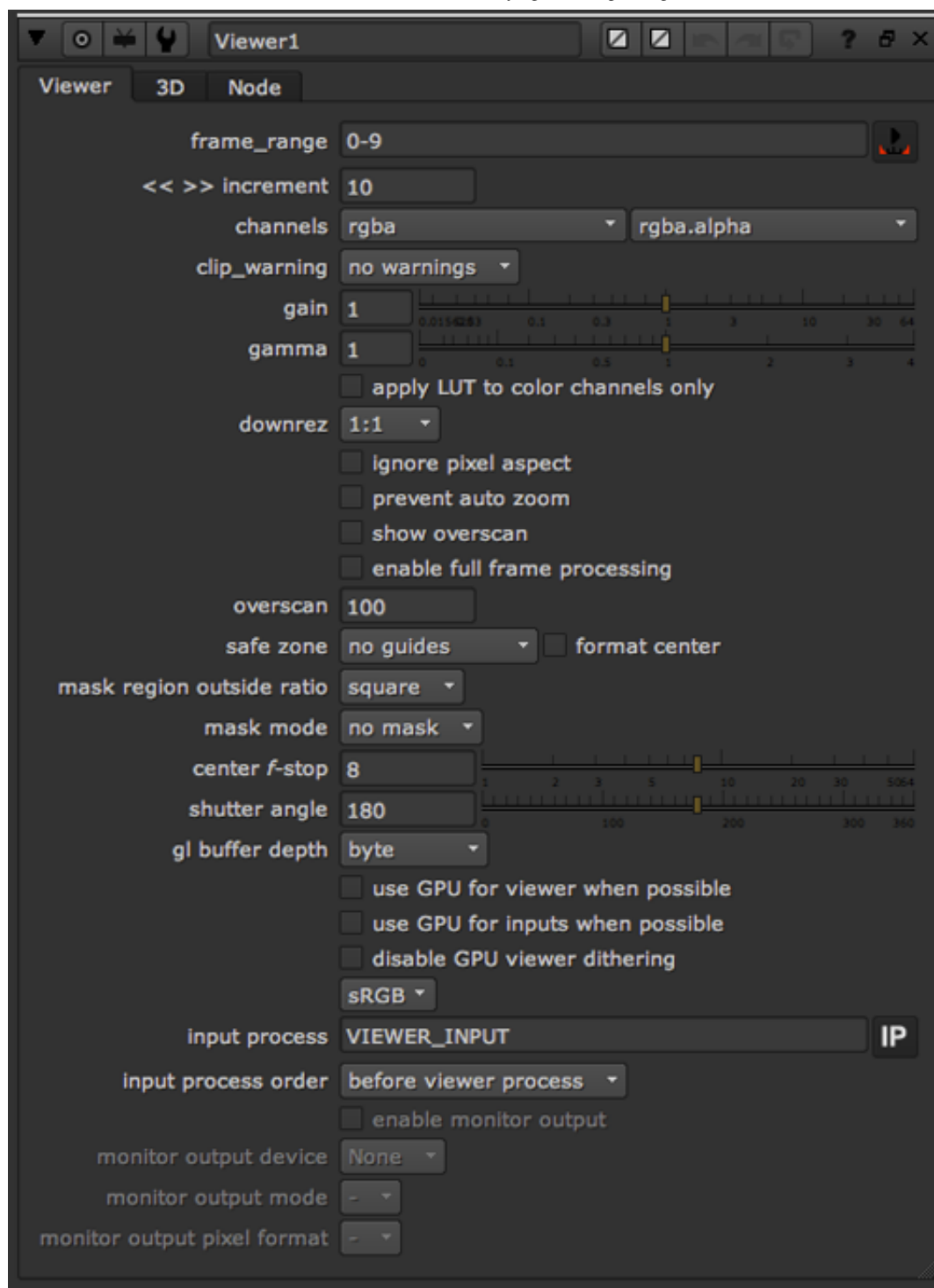
- Name the Read node **Read1_foreground**.
- Set the **frame range** to **black** for both the first and last frame. This specifies that Nuke will show a black frame when you view frames before and after the frame range.



- Open the project settings (press the **S** key while the cursor is in the Properties pane). Set the project's **frame range** to the first Read node's frame range:



11. Open the Viewer node's properties (press the **S** key while the cursor is in the Viewer pane). Set the viewer's **frame range** to the first Read node's frame range:



12. In the Viewer, click on the timeline to move the current time indicator and refresh the Foreground Read node's thumbnail. Return the CTI to the first frame.

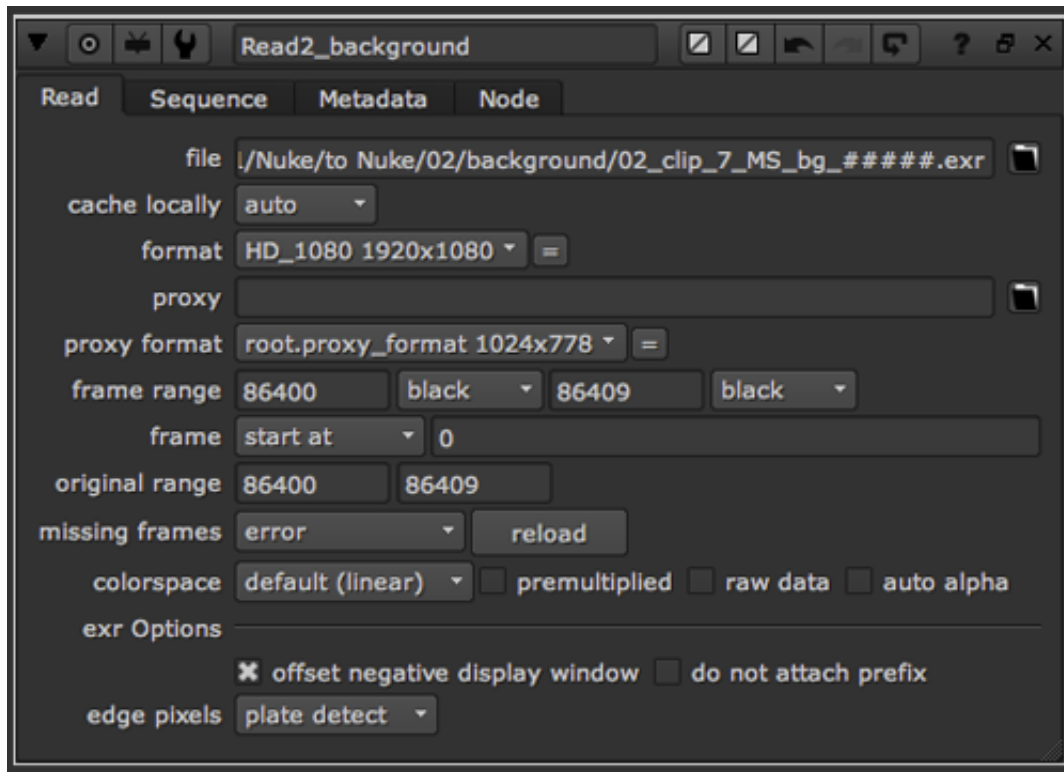


Be sure to return the CTI to the first frame, so that any mattes that you later draw have their first keyframe there.

13. In the **Background** section, add a Read node:

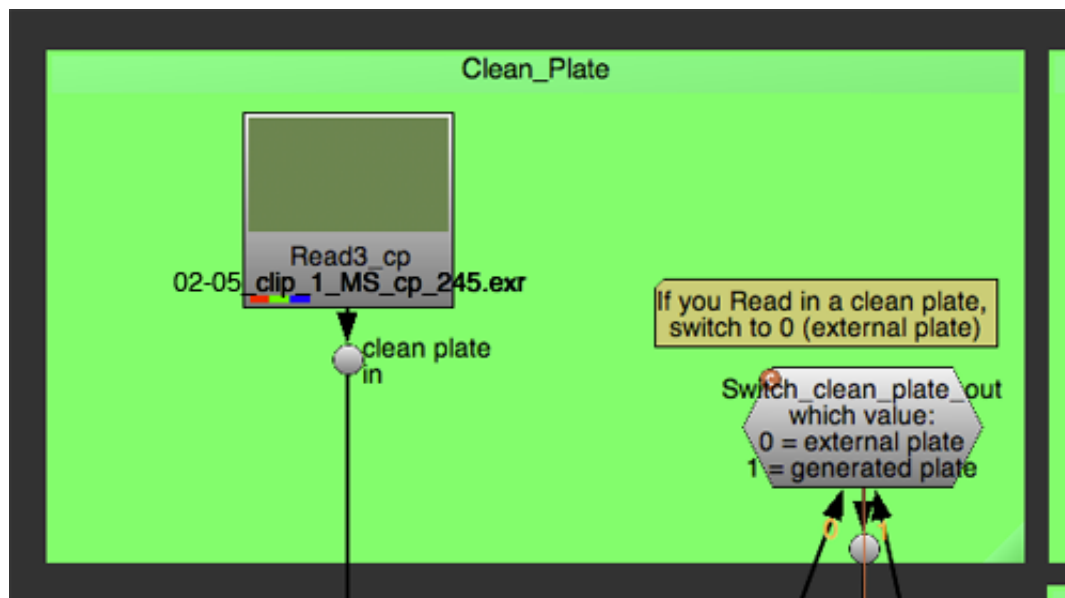


- a. Set the file to that shot's background image sequence.
- b. Name the Read node **Read2_background**.
- c. Set **frame.start at** to the frame number of the first frame of the Read1_foreground node.



- d. Set the **frame range** to **black** for both the first and last frame.

14. The script has an input for a clean plate:



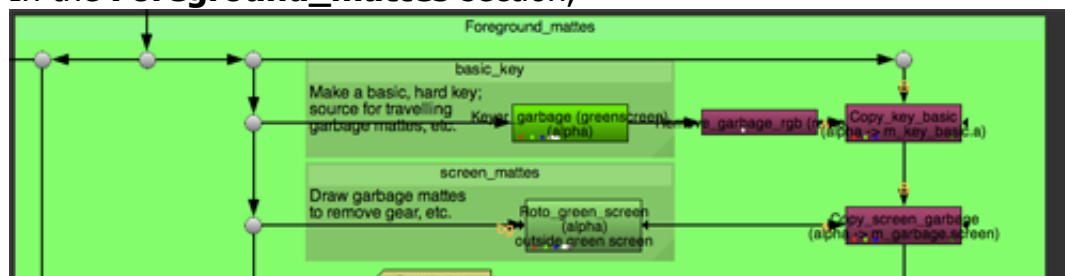
If you don't add an external clean plate, the script will generate a fairly decent one for you. This tutorial doesn't have a clean plate, but this is how you'd add one if it did:

- In the **Clean_Plate** section, add a Read node.
- Set the file to that shot's clean plate image sequence.
- Name the Read node **Read3_cp**.
- Set **frame.start at** to the frame number of the first frame of the Read1_foreground node.
- Set the **frame range** to **black** for both the first and last frame.
- In the properties for the **Switch_clean_plate** node, set the **value** to **0** (to switch the script from the generated clean plate to the external clean plate).

Congratulations! You've imported the footage into Nuke. Yes, it was unreasonably difficult.

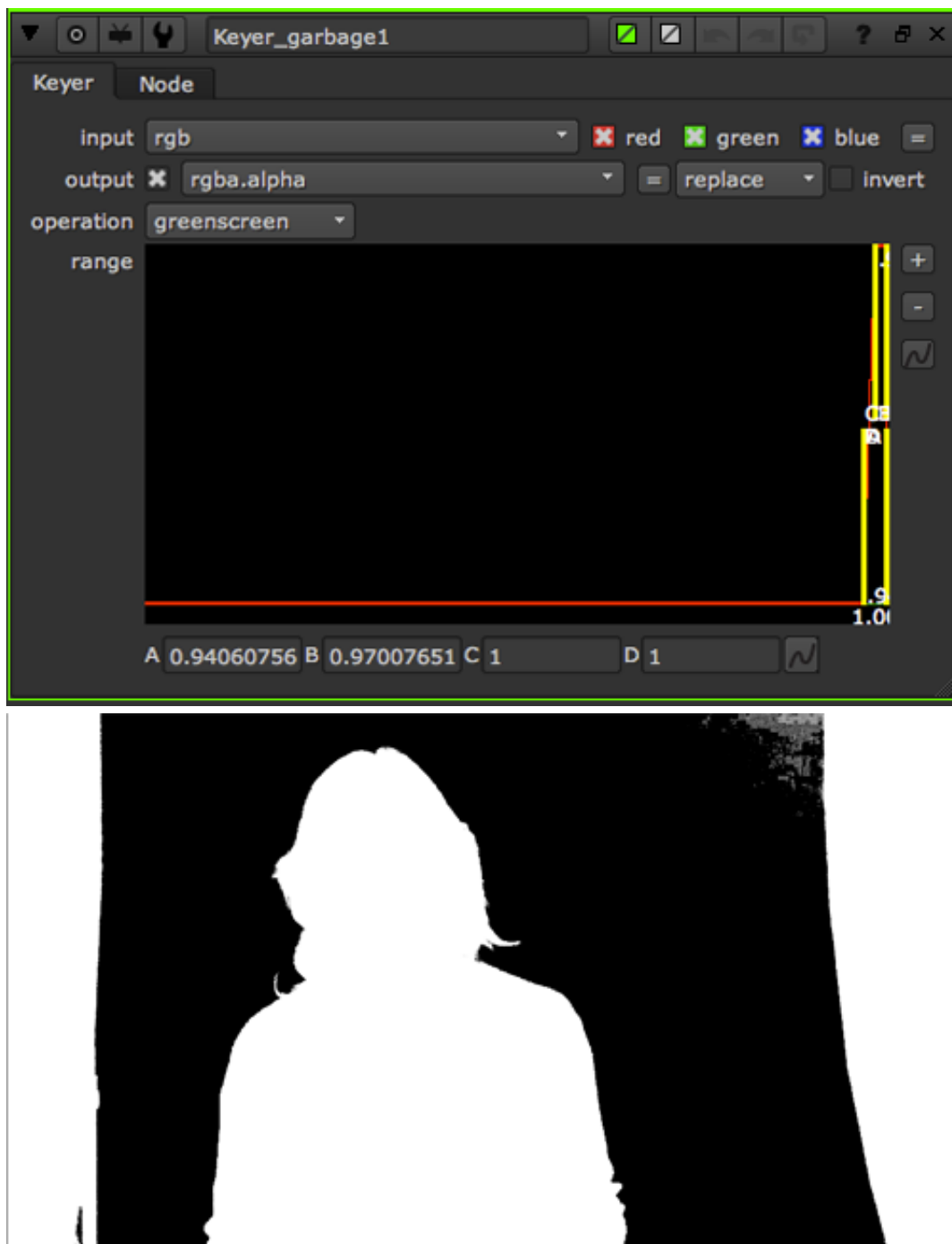
Adjust the default mattes

15. In the **Foreground_mattes** section,



adjust the basic key (a rough key used to make traveling garbage mattes, etc.):

- Set the viewer to the **basic_key** section's **keyer_garbage1** node. Set the viewer to show the Alpha channel.
- Adjust the keyer so that it makes a hard matte:



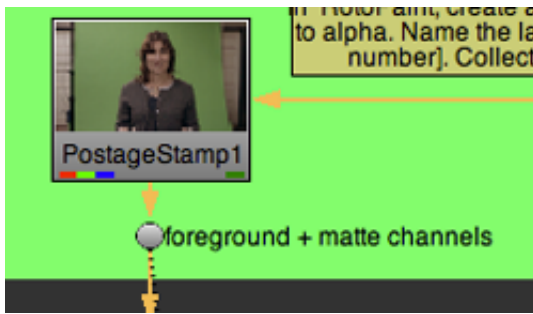
In addition to this keyer's hard matte, the script is designed to use two basic and essential Roto node mattes:

- a green screen garbage matte, which reveals the green screen but masks areas off of the set, lighting equipment, etc.
- a body matte which reveals the foreground (through the m_body.a matte channel)

You have to draw each of these mattes.

16. Add the screen matte:

- a. Set the viewer to the **foreground + matte channels** dot:



Add the green screen garbage matte:

- b. In the **screen_mattes** section's **Roto_green_screen** node, draw one or more garbage mattes:



- c. Simply include the entire foreground image in the matte if there is no equipment, etc., to exclude with a garbage matte.
- d. Use a Tracker or animate the matte if necessary.

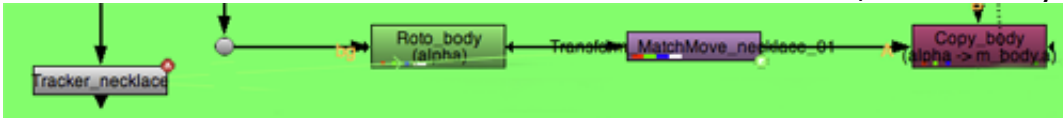
17. Add the body matte:

- a. In the **screen_mattes** section's **Roto_body** node, draw one or more mattes that reveal the foreground:



At this time, be sure to include all of the foreground in the Roto_body node's matte(s).

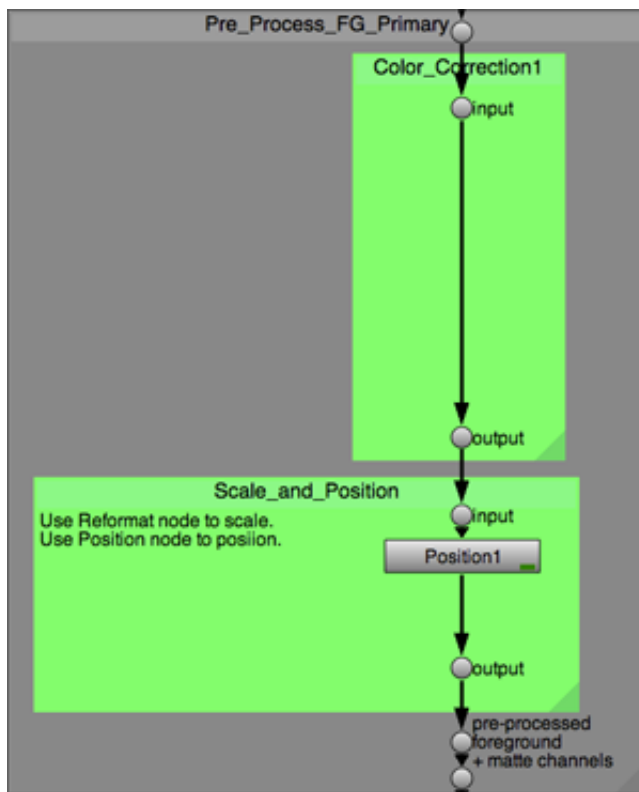
- b. Add a Tracker node and a MatchMove node to the matte, if necessary:



- c. Animate the matte if necessary.

Adjust the foreground pre-processing

18. Adjust the script's foreground position, if necessary:
 - a. In the **Pre_Process_FG_Primary** section's **Scale_and_Position** section, add a Reformat or Transform node to scale the image and / or a Position node to move it:



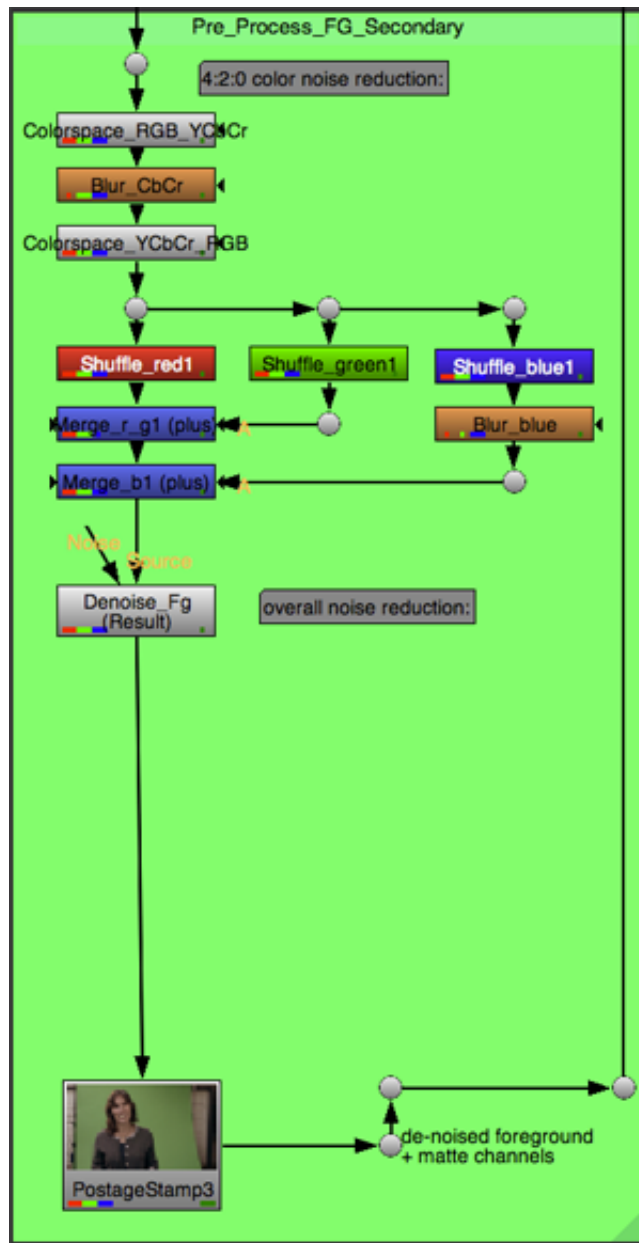
19. Adjust the script's foreground noise reduction analysis region:

- a. In the **Pre_Process_FG_Secondary** section, set the Viewer to view the **Denoise_Fg** node. Open the Denoise_Fg node's properties panel. The node's Analyze Region box will appear in the Viewer.



- b. Adjust the Analyze Region box so that it covers a smooth section of the image, with no image details or shadows. The green screen is a good choice. The Denoise node analyses this one frame.

20. Evaluate the script's noise reduction:

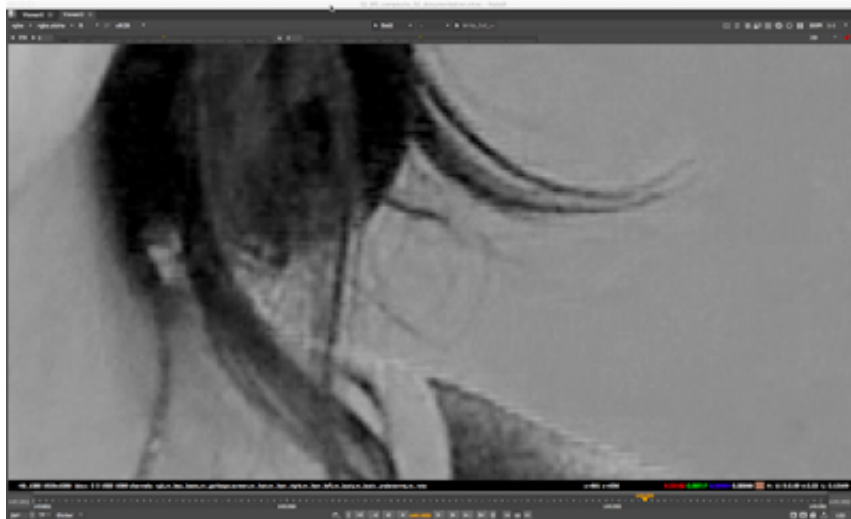


The section has two areas:

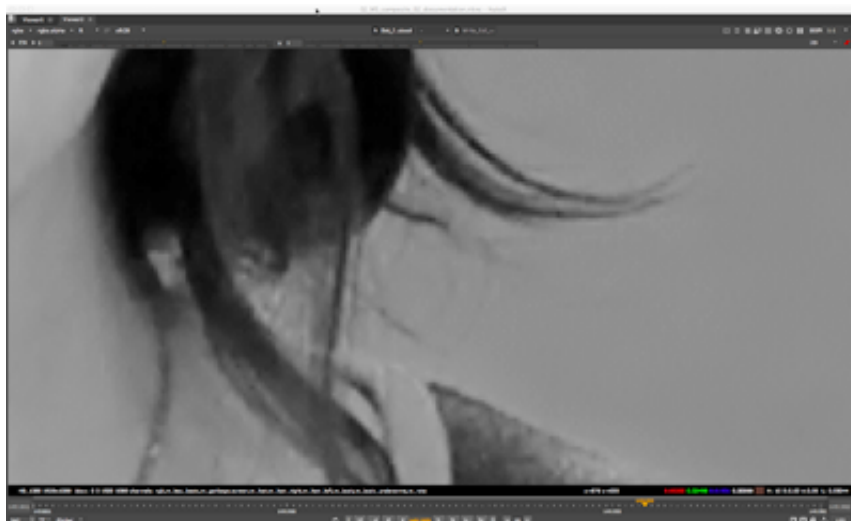
- 4:2:0 color noise reduction (for color compression noise caused by low-resolution video codecs)
- overall noise reduction (adjusted with the the Denoise_Fg node)



The Pre_Process_FG_Secondary section is only used by the Edge section; the **Pre_Process_FG_Primary section's** foreground image is used for the Core and other sections.



Red channel, before noise reduction

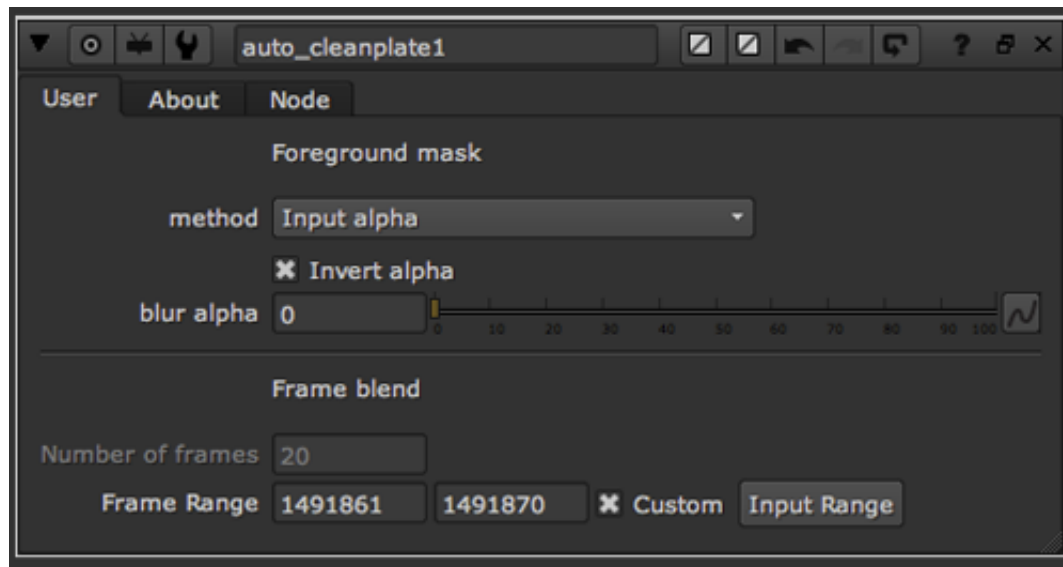


Red channel, after noise reduction

- a. Inspect the individual color channels; evaluate the noise.
- b. If your video doesn't use color compression, turn off the color noise reduction by disabling the two **Blur** nodes in this section.
- c. If you feel that the overall noise reduction is incorrect for this foreground, disable or adjust the **Denoise_Fg** node.

The **Clean_Plate_Generator** section generates a clean plate from the foreground footage. It uses a gizmo called [auto_cleanplate](#).

21. Set the **auto_cleanplate** node to use the project's entire frame range by clicking on the Input Range button:



22. In the Edge section, adjust the **Screen Colour** in the edge_body section's **Keylight_body** node. Use the color picker to select a range of green values near the body.



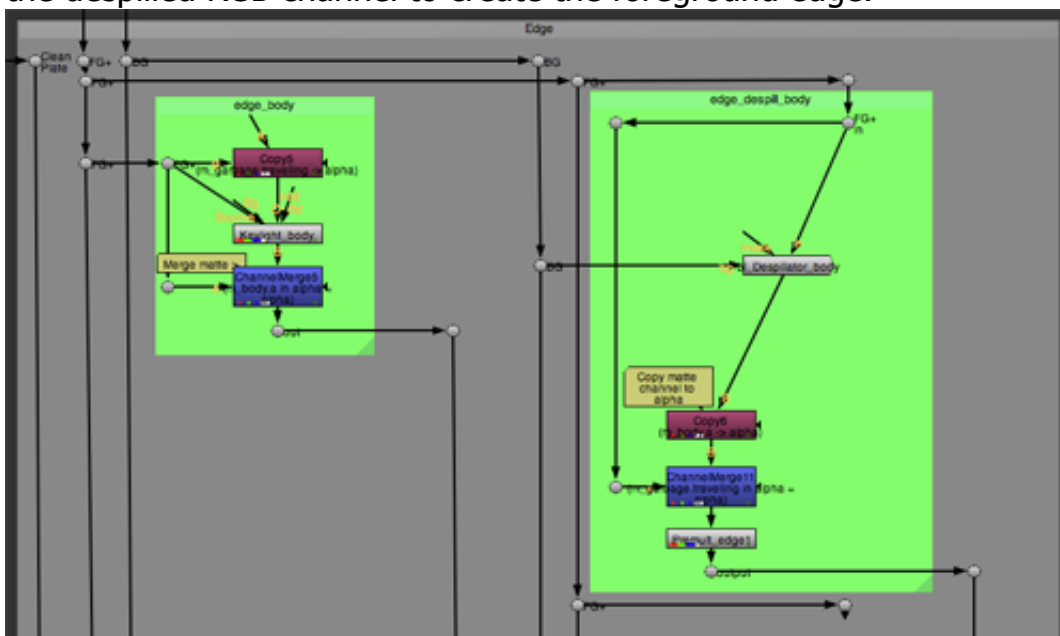
23. Set the Viewer node to the **Final Composite** dot (at the bottom of the script) to view a rough composite:



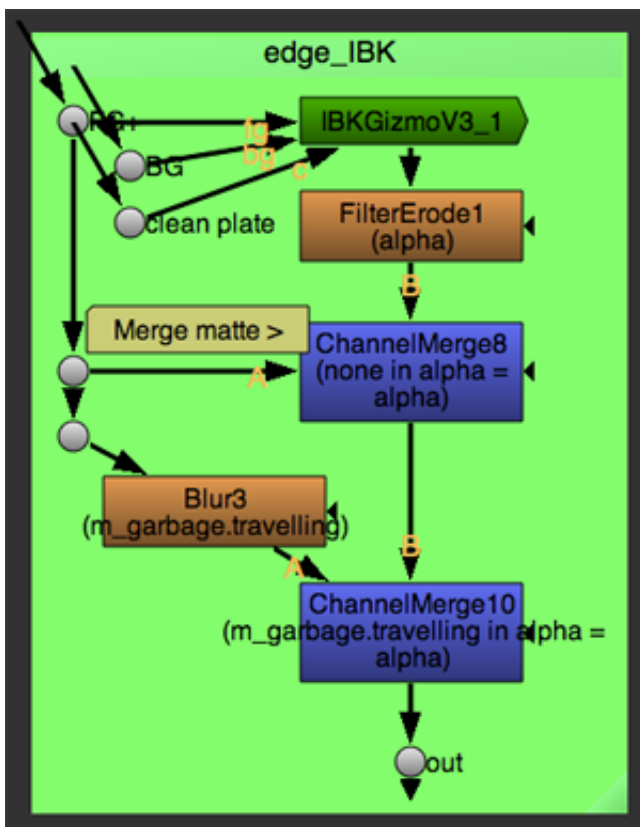
You've adjusted the script template for the foreground shape and green screen values of your imported footage. Next, let's make a better edge for the foreground.

Adjust the edge

The Edge section defaults to one keyer (the **edge_body** section), which creates an alpha matte, and one despiller (the **edge_despill_body** section), which despsills the green from the green screen and replaces it with background pixels. It premultiplies the edge alpha matte with the despilled RGB channel to create the foreground edge.

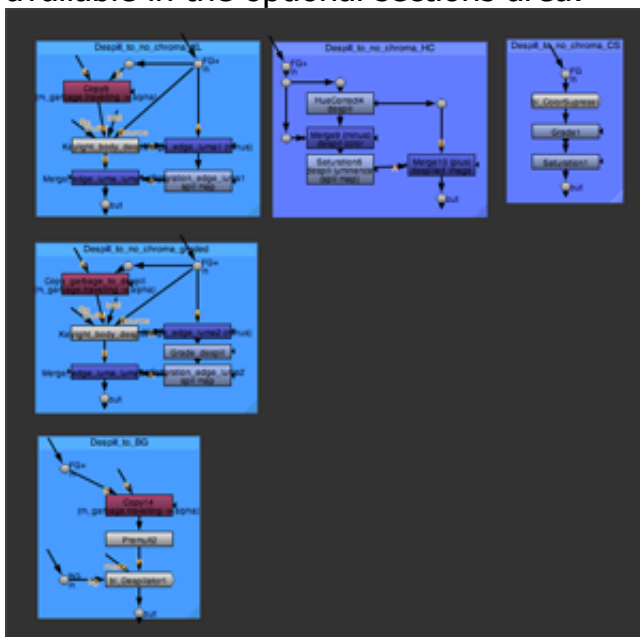


The **edge_body** section uses the Keylight node to make an alpha matte. Keylight is a good, all-around keyer. In the optional sections area of the script, you will find the **edge_IBK** section:



which uses the IBKGizmo node to make an alpha matte. The IBKGizmo is a difference keyer, which can be good with fine details.

The edge_despill_body section uses the [bl Despillator](#) gizmo. Other despiller sections are available in the optional sections area:



You could also use any other keyer or despiller available in Nuke instead of these, as long as you use the **m_body.a** matte channel to matte the foreground's body.

The first thing to do is adjust the edge so that it works well with the simplest (or largest or least-detailed) part of the image, which, in this example, is the body. To adjust the edge:

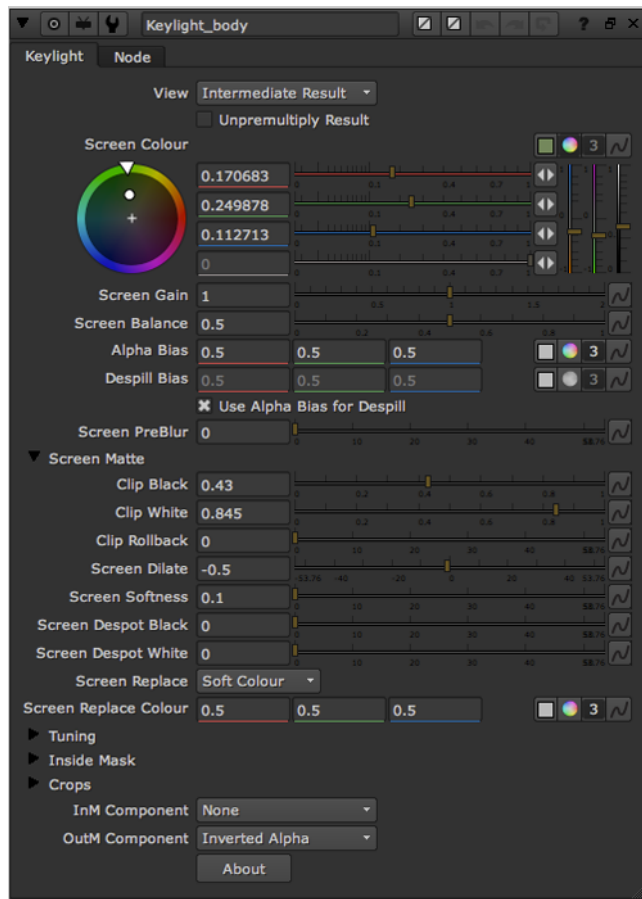
24. Confirm that the merge nodes for the Shadows, Core and Light_Wrap sections are disabled (effectively turning off these layers):



25. Adjust the edge matte (using, in this example, the edge_body section's Keylight node):



Don't worry about the foreground image quality; focus on the edge. This is just a source for the edge alpha channel.



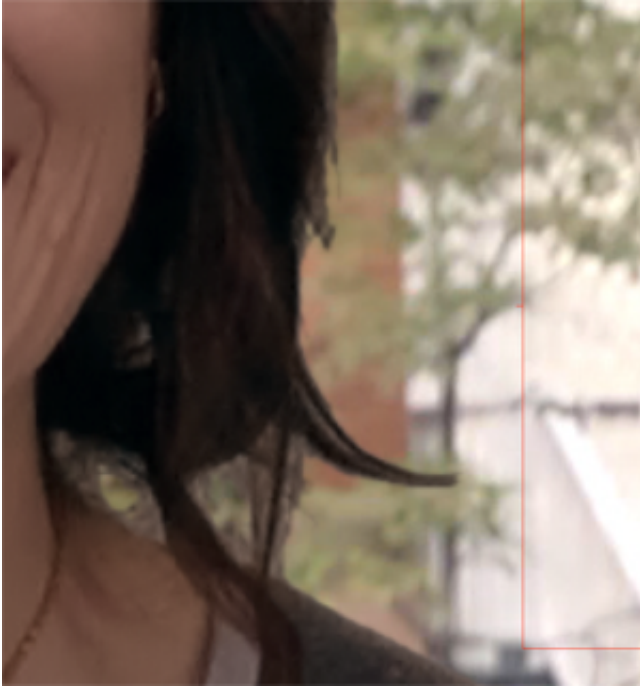
Keylight's Status view

- The traveling garbage matte will mask the majority of the background. The core layer (which we'll make later) will fill in the center.**
 - Keep the largest possible difference between the white and black settings;** this difference is where semi-transparent alpha pixels live.
 - In Keylight, set the view to **Intermediate Result** to disable any spill suppression.
26. Play the shot and look carefully for problems with the edge that can be solved by adjusting the keyer. Remember, you are working on the simplest (or largest or least-detailed) part of the image.

The goal is to make a clean alpha channel on the edge, where solids are solid and transparencies are transparent, preserving the most edge detail possible.

It is OK to preserve fine detail with a somewhat grey edge matte: the despiller section(s) can merge this somewhat with the background.

Here's the best that Keylight could do:



The edge matte still needs work, notably in the hair. You'll need to separate this problem part of the image from the body using a matte and adjust it individually.

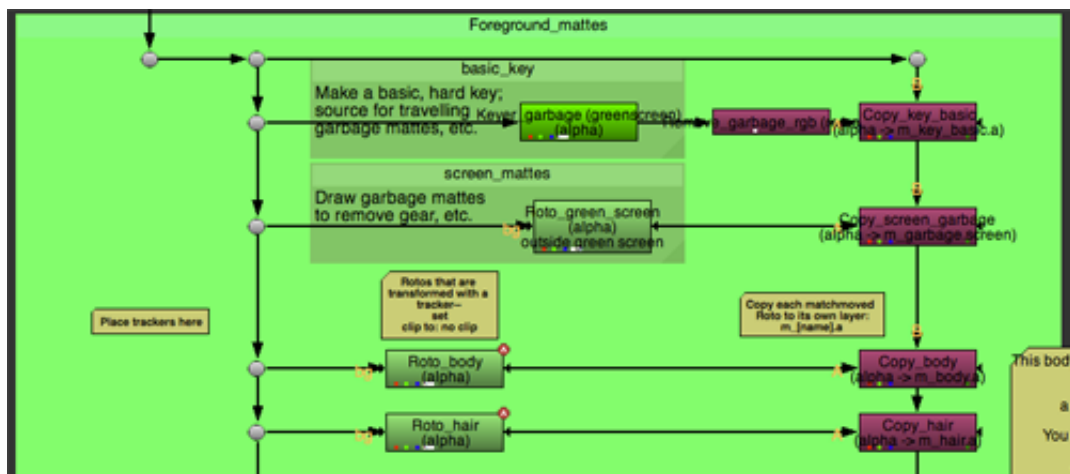
Adding mattes

27. In the Foreground_mattes section,
 - a. adjust the Roto_body matte so that it excludes the hair portion of the edge_body key. Feather the matte:



Next, you'll draw mattes to separate the image into optimally workable parts:

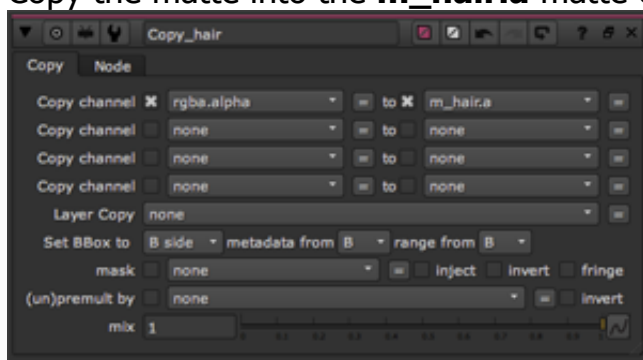
28. Draw a matte that includes the hair and overlaps with the top part of the body matte:
 - a. Add a Roto node-Copy node combination.



b. Draw a matte for the hair edge.



c. Copy the matte into the **m_hair.a** matte channel.



The script includes several other channels that are used by the edge, core and light wrap sections to fine tune the various parts of the image. The names of these channels begin with **m_** (for matte) and end in **.a** (for alpha):

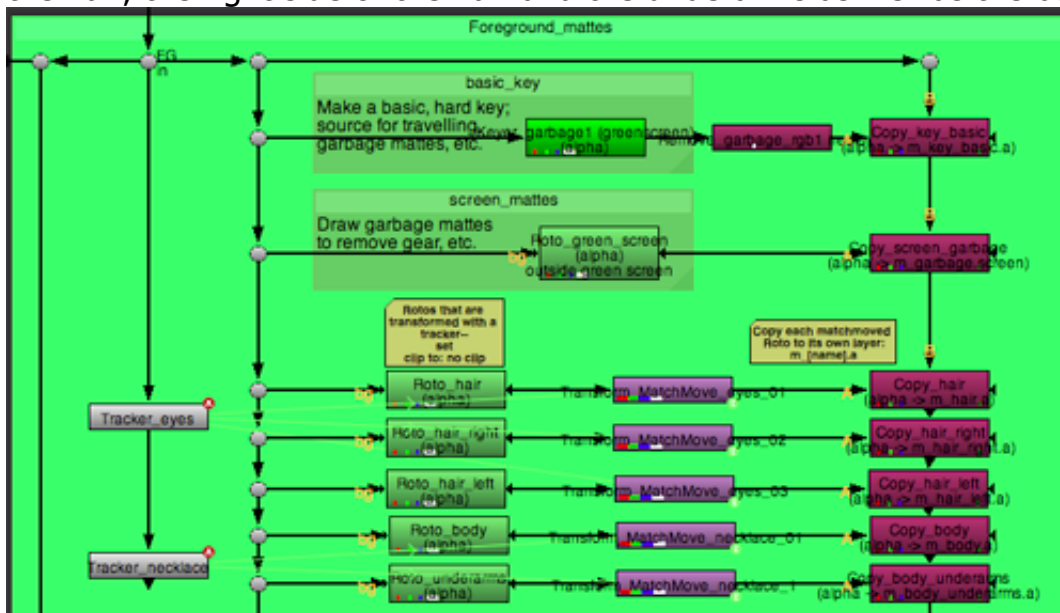
- m_body.a
- m_body_underarms.a
- m_hair_left.a
- m_hair_right.a
- m_shoes.a
- m_torso.a

and so on.

You can create your own channel if necessary by selecting **new** from the Copy node's channels list.

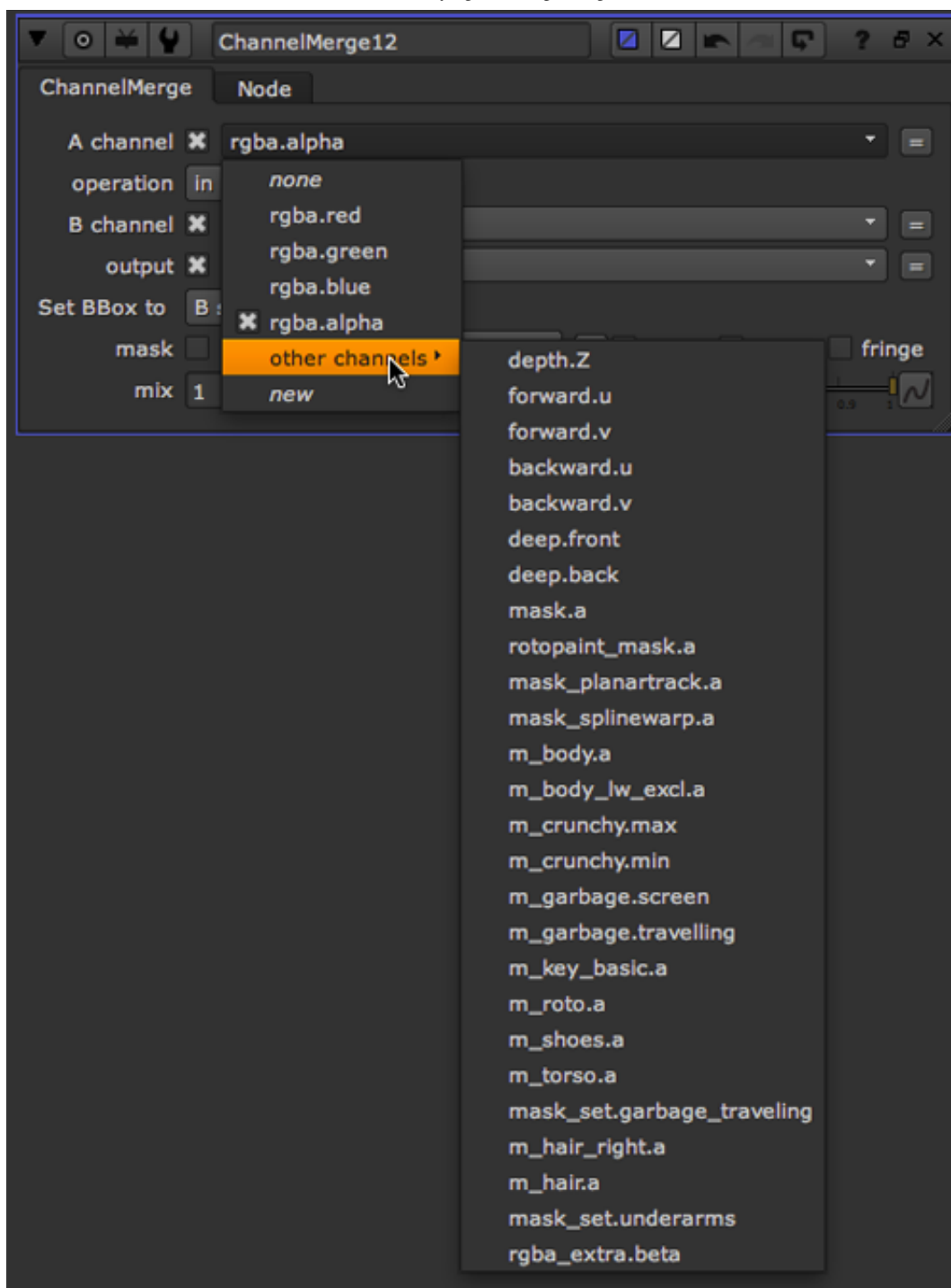
- d. The tutorial foreground and background sample clips are only 10 frames long: there isn't much movement, so it isn't necessary to animate the mattes. You'd add Tracker nodes to any footage with much movement and apply a MatchMove node to the mattes. Name any new Tracker nodes to specify what item in the image is being tracked (for example, Tracker_necklace).
- e. In Roto nodes that are transformed with a tracker, set **clip to:** to **no clip**.

The body and hair mattes will do for this tutorial. In the much longer shot that I used in my movie *#LookUp*, I made separate mattes for the top of the hair, the left side of the hair, the right side of the hair and the underarms as well as the body:

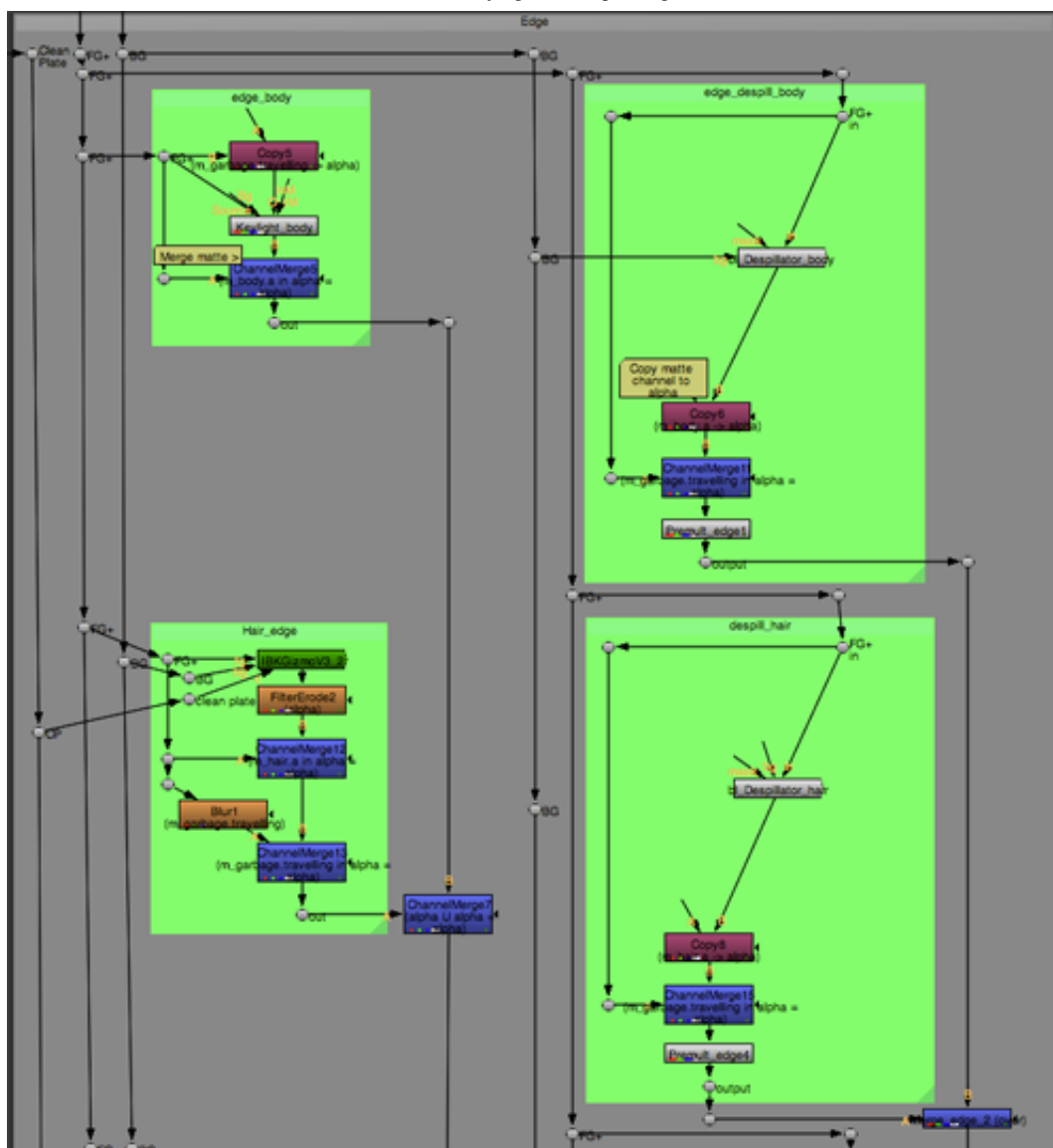


Refining the edge

29. In the edge section, add a keyer and despiller for the hair portion of the image.
 - a. Make a copy of an edge_IBK section (which is loitering in the optional sections portion of the script). Name the copy **Hair_edge**.
 - b. Similarly, make a copy of an edge_despill section. Name the copy **despill_hair**.
 - c. Set the hair's keyer and despiller to use the **m_hair.a** matte channel:



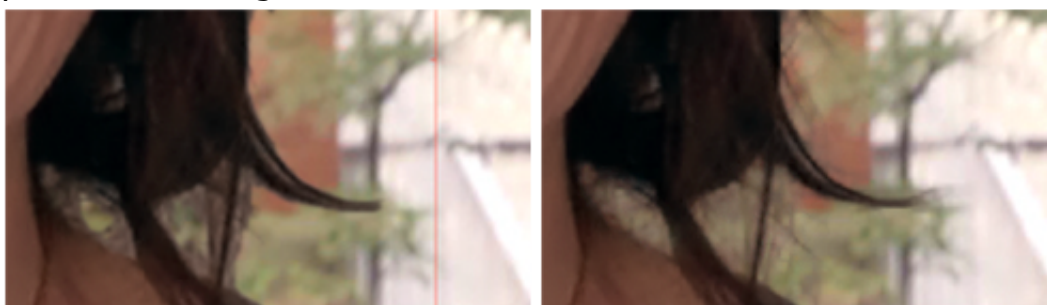
- d. Connect the keyer and despiller sections down the edge tree, merging each with the previous one:



30. Adjust the hair edge:

- Fine tune the keyer until it creates a good edge matte for the hair.
- Fine tune the despiller until it creates a well blended edge.

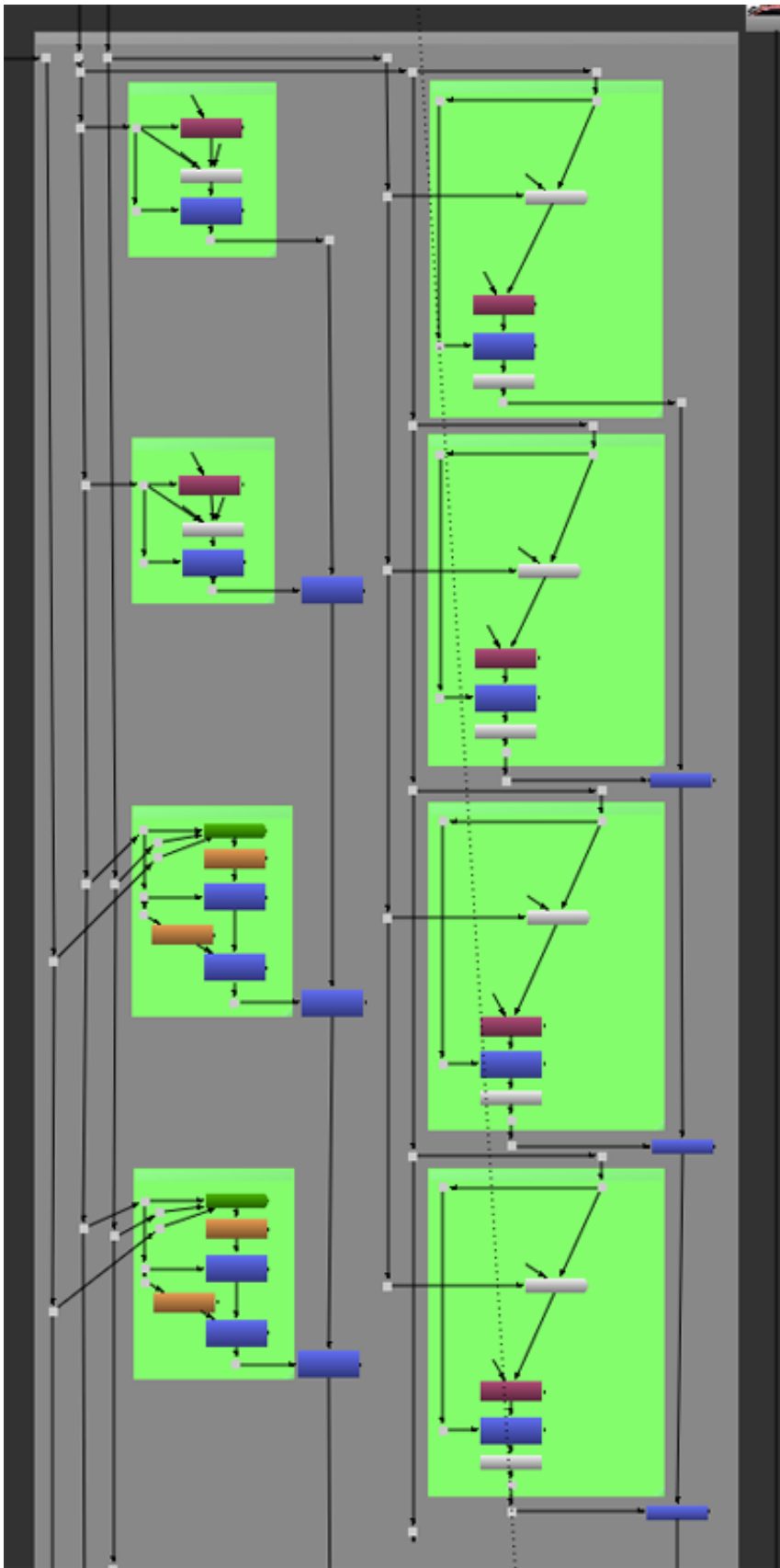
Creating a separate matte for the hair and keying it differently improved the hair portion of the edge:



body matte with Keylight

separate hair matte with IBKGlzmo

When I composited the longer shot for my movie, I used several more mattes, keying and despilling sections:



The completed edge matte and despilled image looked like this:



31. Rotoscope any stubborn flaws in the edge composite.

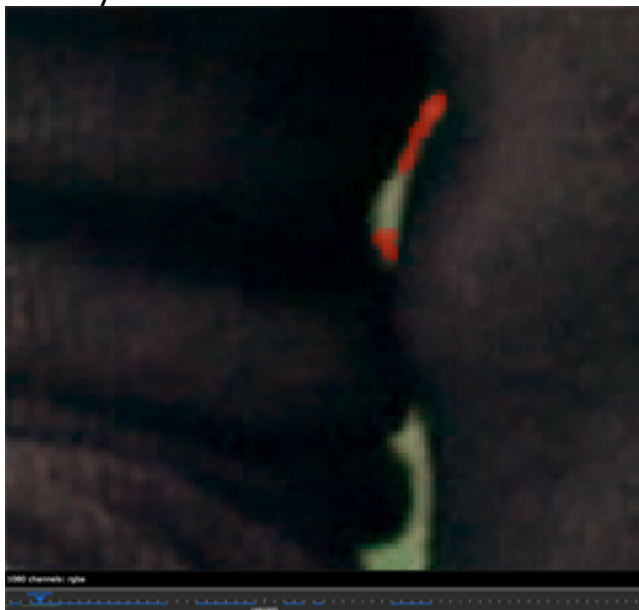
Even with great keying and compositing, it is possible to have webbed appendages and other flaws. For example, diffraction from fingers or arms being close together de-saturates the green screen, keeping those spots in the RGB and out of the alpha channel. Despilling doesn't affect these parts because they aren't actually very green. The only way to deal with these is by rotokeying them out. Painting them works best.

- a. Set the Viewer node's input to be from the the Roto_touchups section's RotoPaint_roto node.



Press the M key in the viewer to view the matte overlay.

- b. Set a second Viewer node's input to the final composite (from the **Final Composite** dot at the bottom of the script).
- c. Open the RotoPaint_roto node's properties panel.
- d. Select the paint brush tool. Paint pixels that you want to erase with white paint. The brush strokes create a matte in the alpha. You can see the matte in the matte overlay in Viewer 1



and see the result of the matte on the composite in Viewer 2.



RotoPaint requires a lot of processing power. This can slow the responsiveness of the script to a crawl. Once you've finished rotoscoping these touchups, **disable the RotoPaint_roto, Invert_roto and Copy_roto nodes** (thereby turning off the rotoscoped touchups) as often as possible **while you work on the rest of the script.**

When completed, the edge will look something like this:



finished edge layer



finished edge layer, composited over the background layer

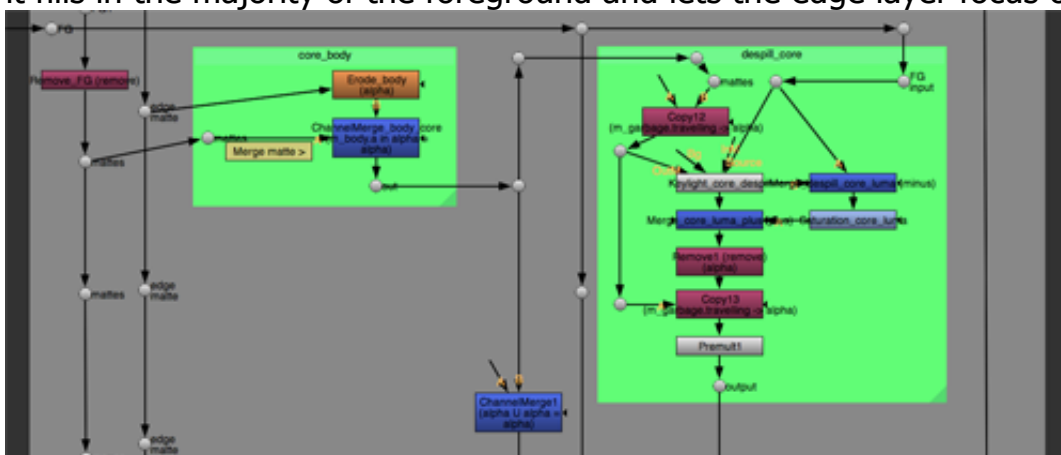
The edge is good. However, in an effort to make the edge of the hair more nuanced, the middle of the image may look a little thin:



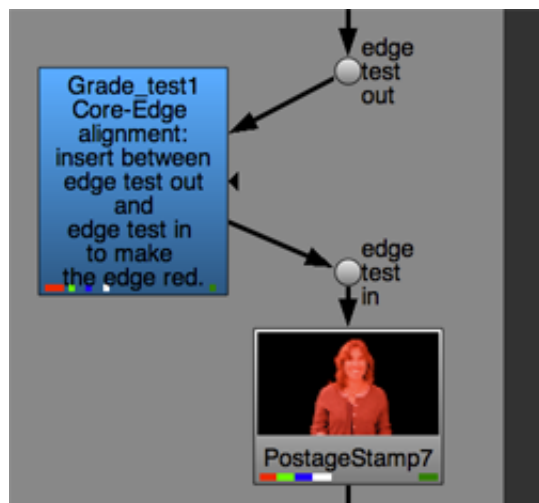
Next, we'll fill in the middle of the foreground with the core layer.

Adjust the core

The Core section defaults to one matte (in the **core_body** section), which creates a core alpha matte. It uses one despiller (the **despill_core** section), which despill the green from the green screen and replaces it with neutral grey pixels. It premultiplies the core alpha matte with the despill RGB channel to create the core. This core layer is set inside the foreground edge; it fills in the majority of the foreground and lets the edge layer focus on the edges.

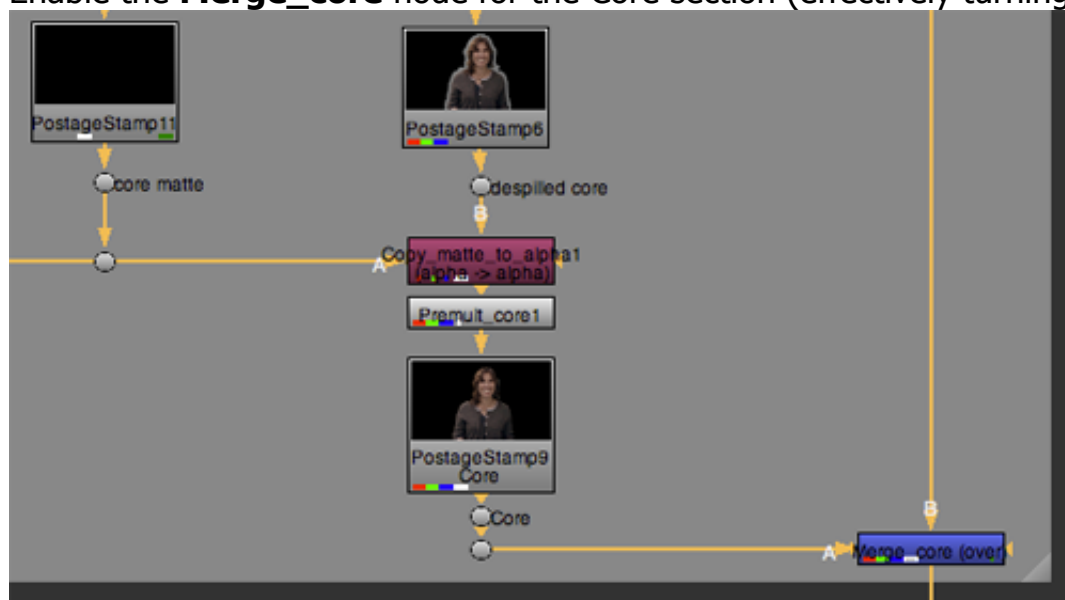


32. In the **Core** section's **despill_core** section, adjust the **Keylight_core_despill** node's **Screen Colour** to select the most accurate green screen color (so that the section can properly despill the core).
33. View the final composite (from the **Final Composite** dot at the bottom of the script).
34. Connect the **Grade_test** node between **edge test out** and **edge test in**:

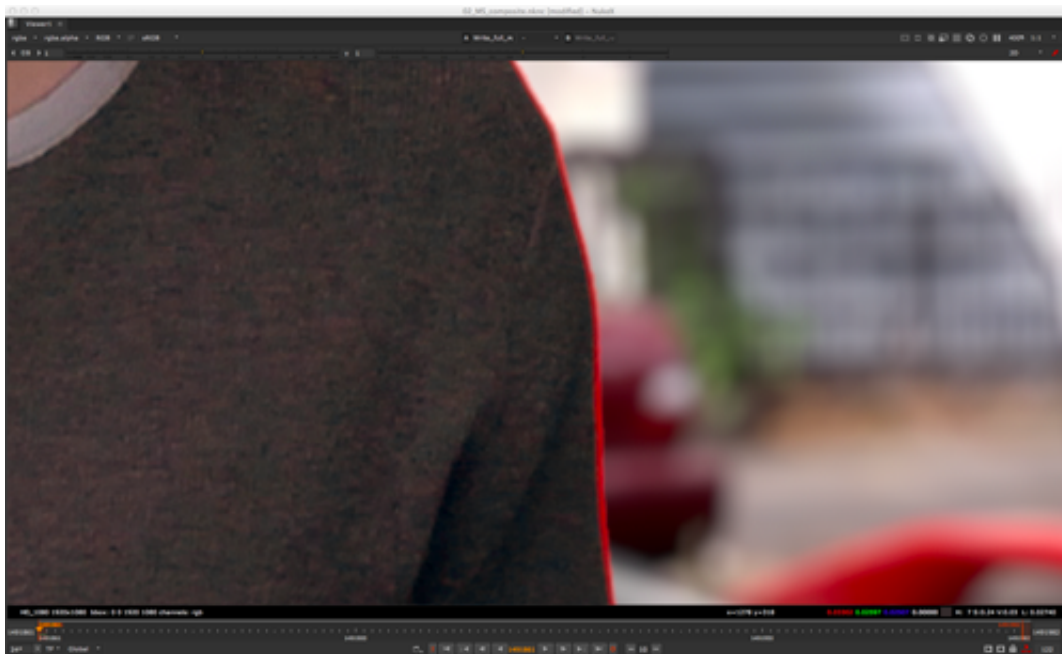


This will turn the edge layer red.

35. Enable the **Merge_core** node for the Core section (effectively turning on the core layer)



36. In the core_body section, adjust the Erode_body node until the core layer is just inside the edge layer:

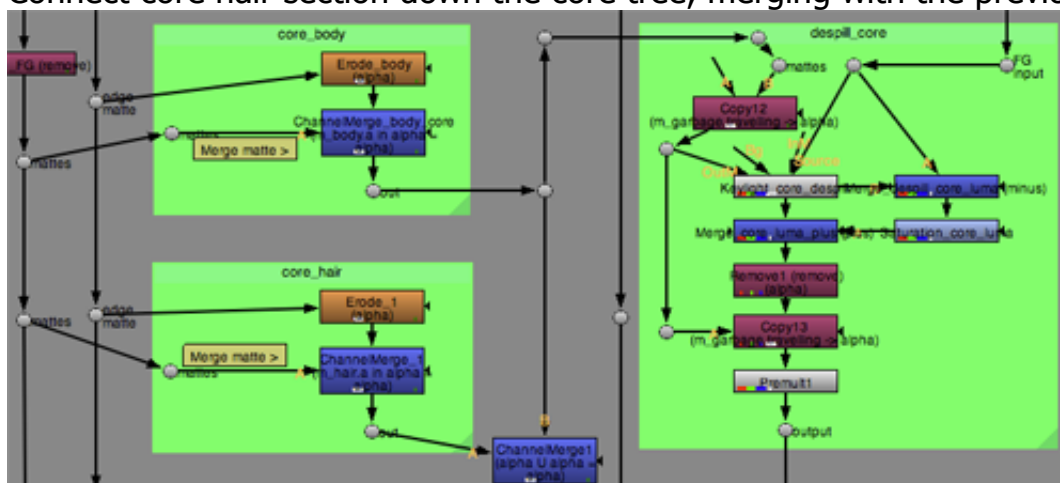


37. Play the shot and look carefully for problems with the core that can be solved by adjusting the Erode_body node. The goal is to have the core extend as close to the edge of the foreground image as possible without obscuring any of the edge.

Because you used multiple mattes in the Edge section to adjust the hair separately, you will need to add core sections that composite this part of the image in the core layer.

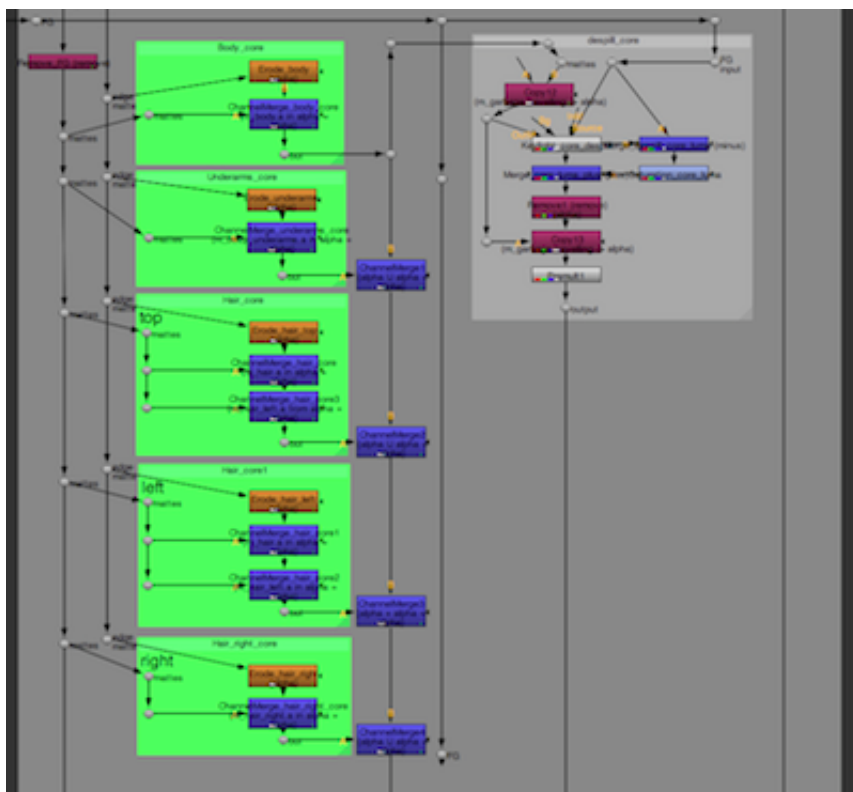
Refining the core

38. In the Core section, add a copy of the core Erode section for the hair.
- Make a copy of a core_section (in the optional sections area). Name the copy **core_hair**.
 - Set the hair's ChannelMerge node to use the **m_hair.a** matte channel.
 - Connect core hair section down the core tree, merging with the previous one:



39. Fine-tune the hair core Erode section until it creates a good core matte.

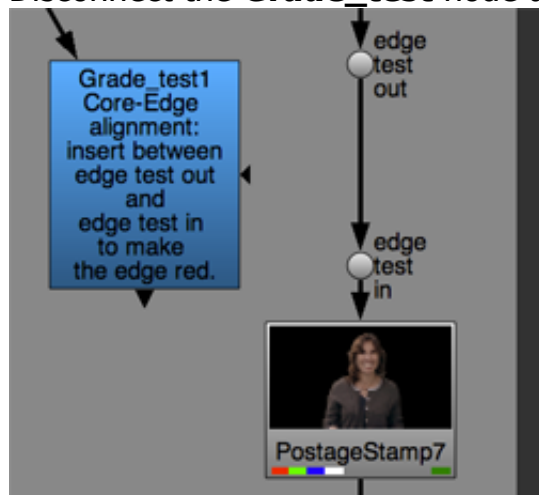
In my movie, this shot needed several more core mattes:



The completed core matte and despilled image looked like this:



40. Disconnect the **Grade_test** node and reconnect the **edge test out** and **edge test in**:



When completed, the core layer will look something like this:



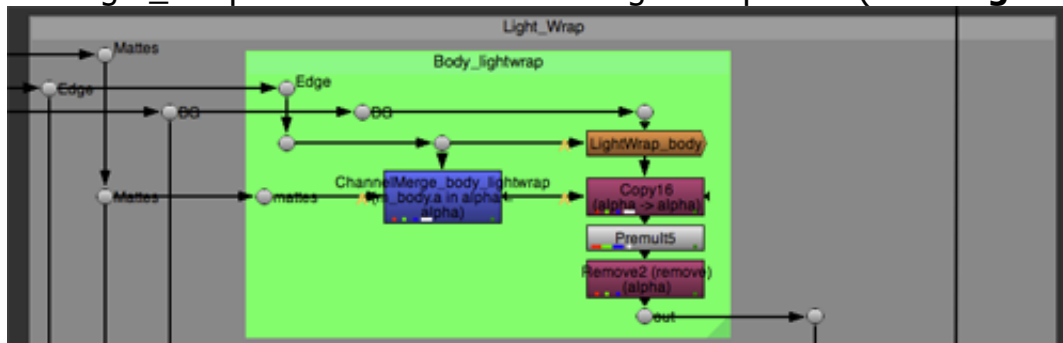
finished core layer



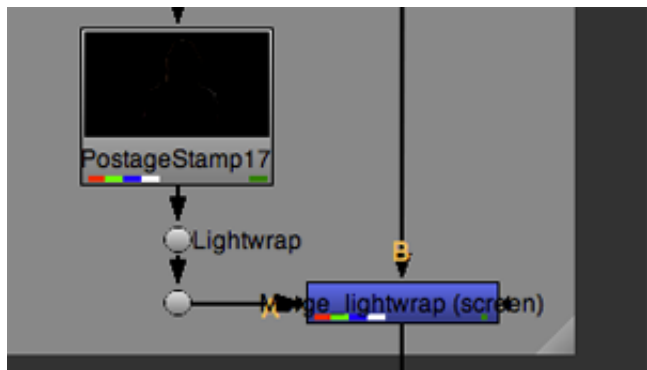
core layer composited over the edge and background layers

Adjust the Light Wrap

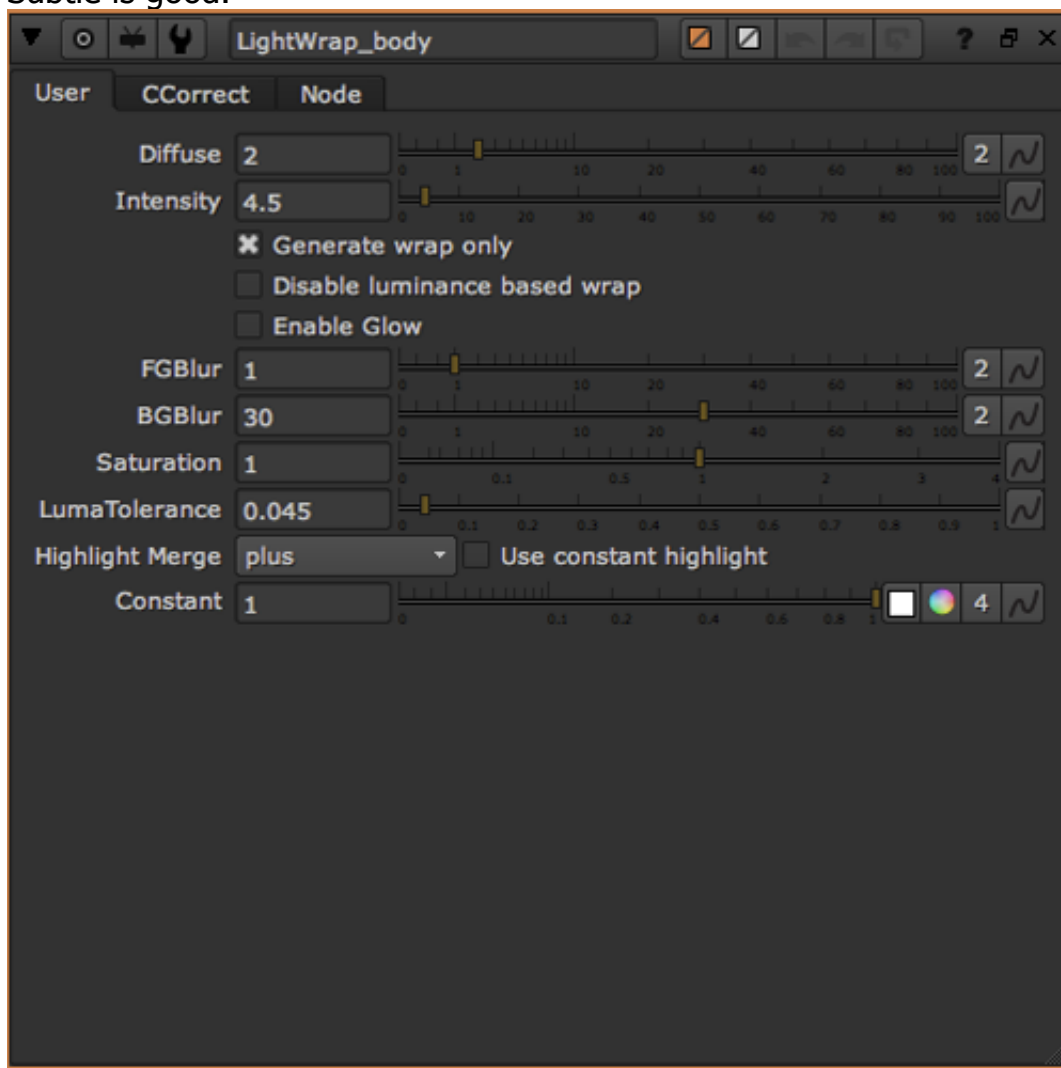
The Light_Wrap section defaults to one light wrap node (in the **lightwrap_body** section).



41. In the the **Light_Wrap** section, enable the **Merge_lightwrap** node (effectively turning on the light wrap layer):



42. View the LightWrap_body node in one Viewer input and the final composite (from the **Final Composite** dot at the bottom of the script) in another.
43. In the LightWrap_body node's properties, adjust the light wrap on the body portion. Subtle is good.

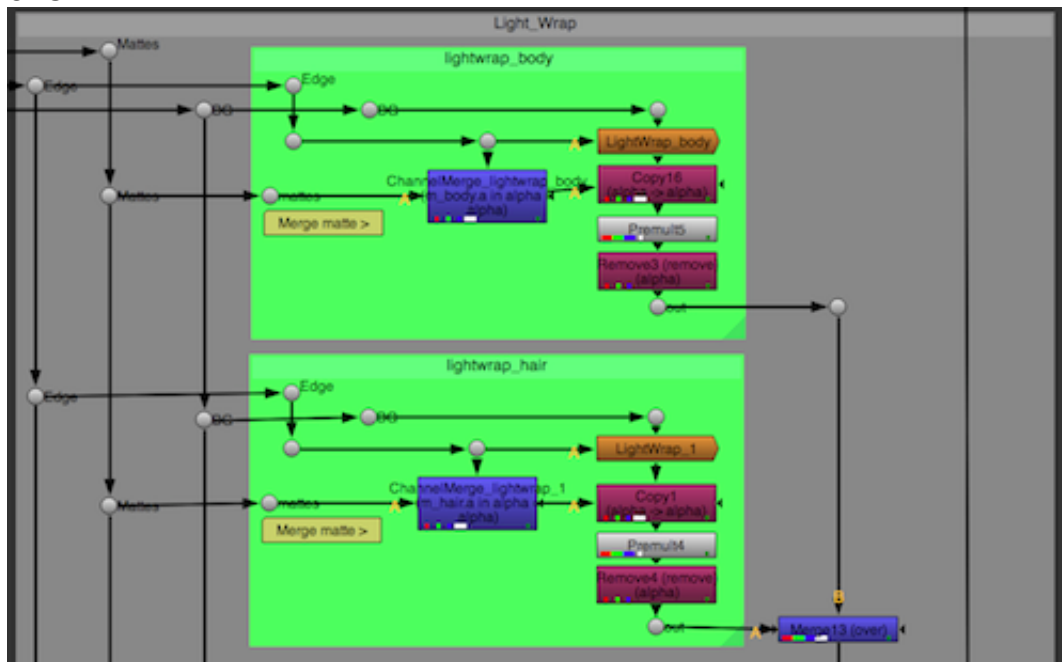


44. Play the shot and look carefully for problems with the light wrap that can be solved by adjusting the LightWrap_body node. The goal is to very slightly blend the foreground image with the background, the way diffraction does in real life, in a naturalistic way. Less is more.

Because you used multiple mattes in the Edge section to adjust the hair separately, you will need to add light wrap sections that composite this part of the image in the light wrap layer.

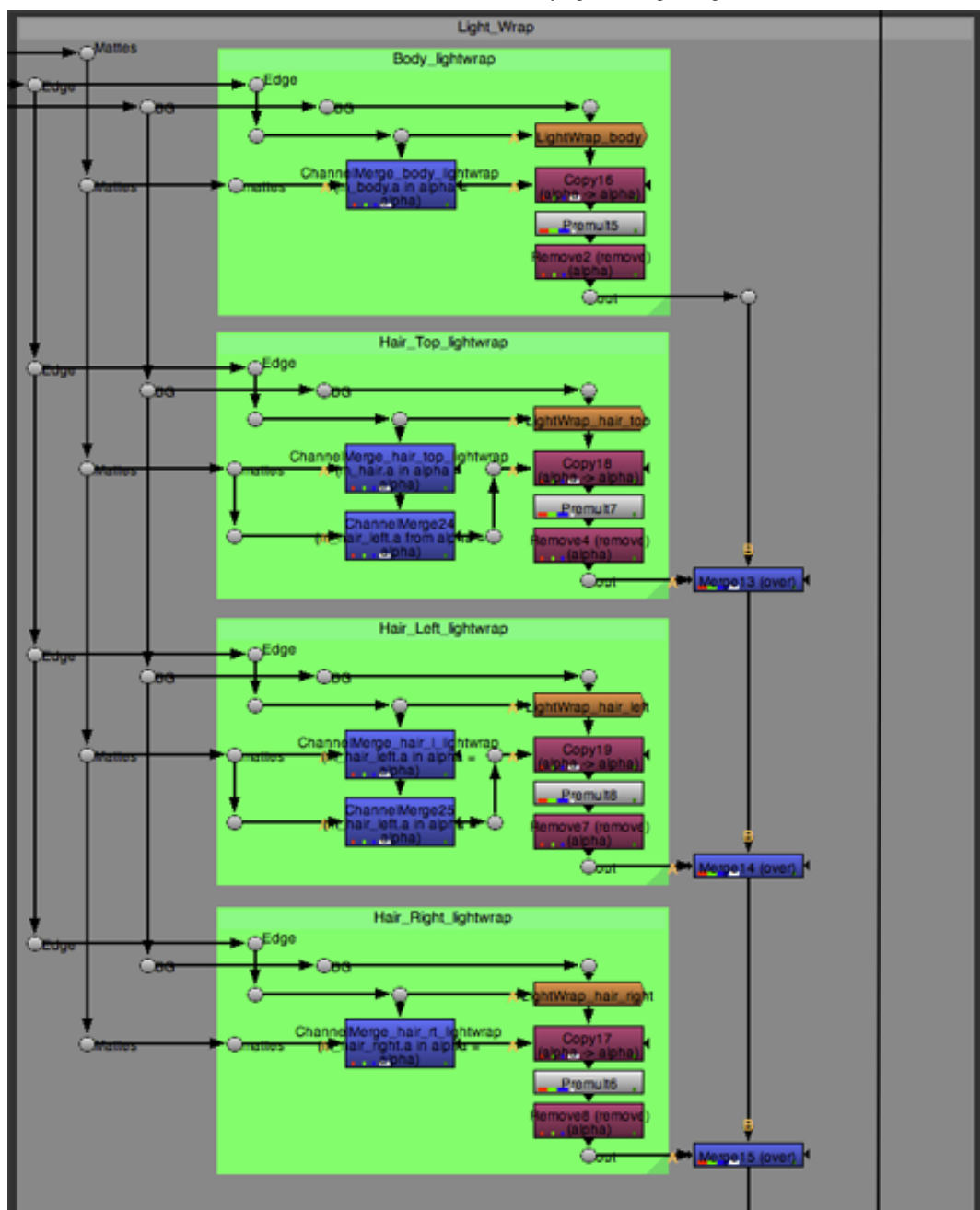
Refining the light wrap

45. In the Light_Wrap section, add a copy of the light wrap node section for the hair.
- Make a copy of a lightwrap_ section (in the optional sections area). Name the copy **lightwrap_hair**.
 - Set the hair's ChannelMerge node to use the **m_hair.a** matte channel.
 - Connect the core hair section down the light wrap tree, merging with the previous one:

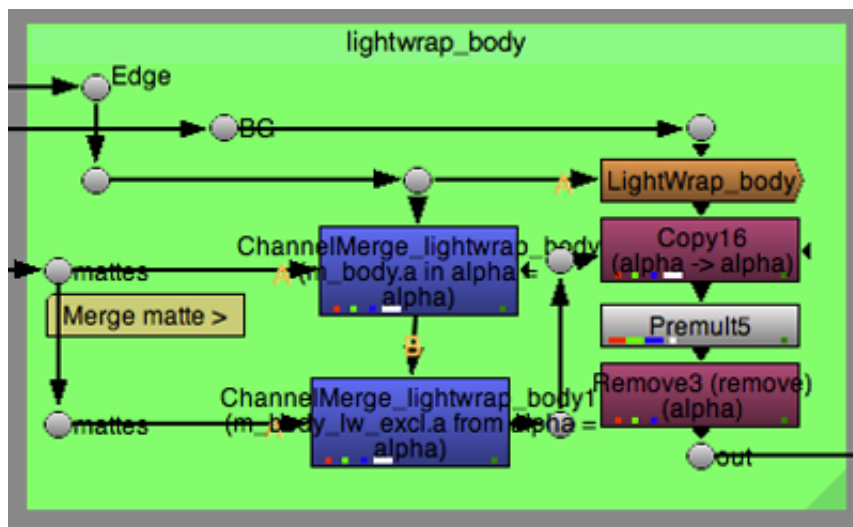


46. Fine-tune the hair LightWrap node until it creates a good (read: subtle) light wrap.

In my movie, this shot needed several light wrap sections:



Do not use a light wrap on parts of the image where it doesn't look right or make sense. For example, the bottoms of people's feet shouldn't have a light wrap. You can make a matte (in the Foreground_mattes section) to exclude part of the image from a light wrap:



When completed, the light wrap layer will look something like this:



finished light wrap layer,
shown against magenta background and
exaggerated in the Viewer for illustration



finished edge, core, light wrap and background layers composited together

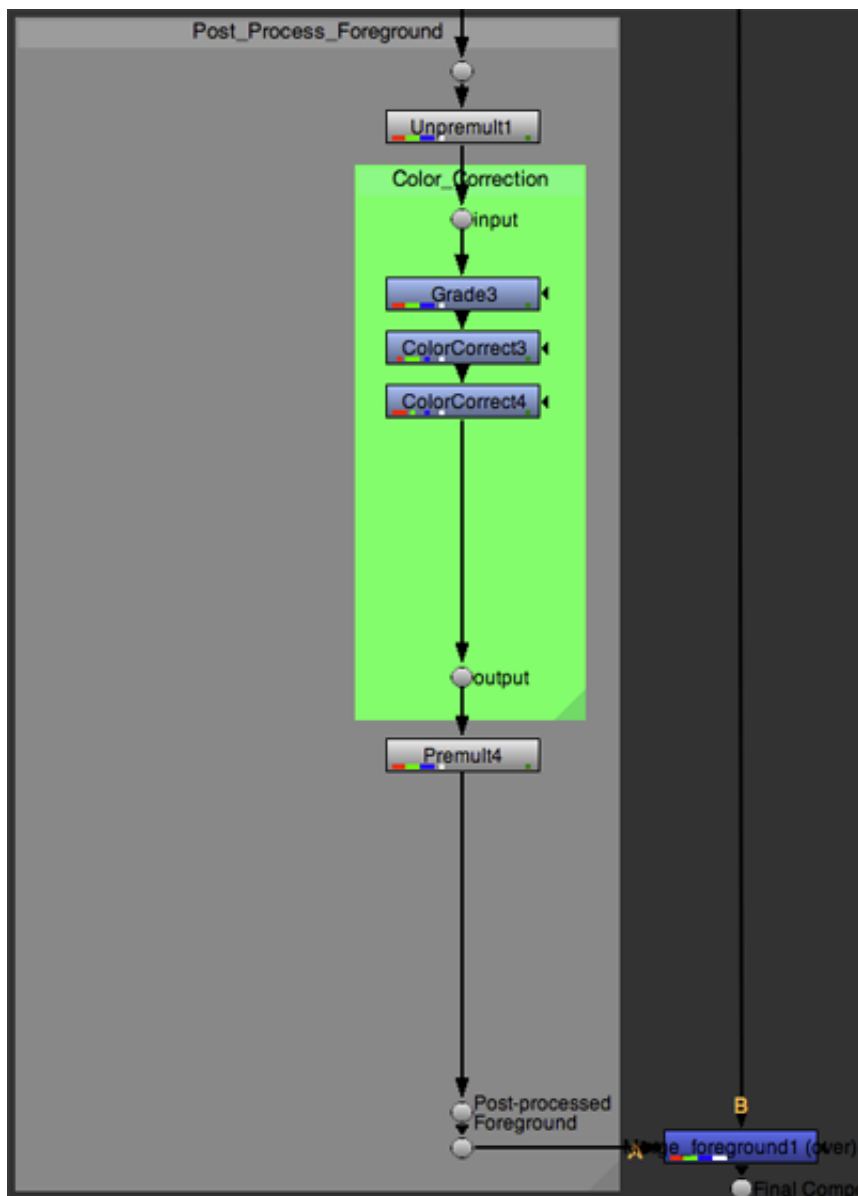
Matching

You can adjust the foreground's scale and position in the the Pre_Process_FG_Primary section; use a **Reformat** or **Transform** node to scale the image and the **Position** node to position the image.

There are two places to adjust the foreground's color:

Pre_Process_FG_Primary This section comes before the Edge and Core sections; changes may affect their keys.

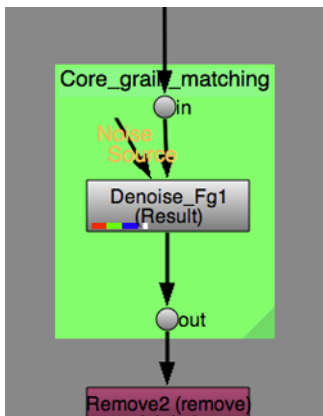
Post_Process_Foreground This section comes after the Edge, Core and Light Wrap foreground elements have been composited. This could alter how the despilling and light wrap blend with the background.



There are two places to adjust the foreground's grain (or noise):

Core_grain_matching This section only adjusts the core; as a noisy edge is usually a bad thing, this is probably the best place.

Post_Process_Foreground This section adjusts the composited edge, core and light wrap foreground elements.



You can adjust the background's color, scale and position in the **Background** section.

47. In the **foreground's Pre_ and / or Post_ processing sections** and in the **Background** section, grade, scale and position the foreground and / or background to match each other.
48. Match the foreground's grain (or noise) to the background's in the **Core** section's **Core_grain_matching** section or in the **Post_Process_Foreground** section.

Shadows

Now that you have the hang of compositing the foreground and background in shot 2, we'll cover adding shadows using shot 1. Its foreground and background shots are in the tutorials **/~Nuke template and tutorial/shadows/01/** folder:



01_LS_fg.mov



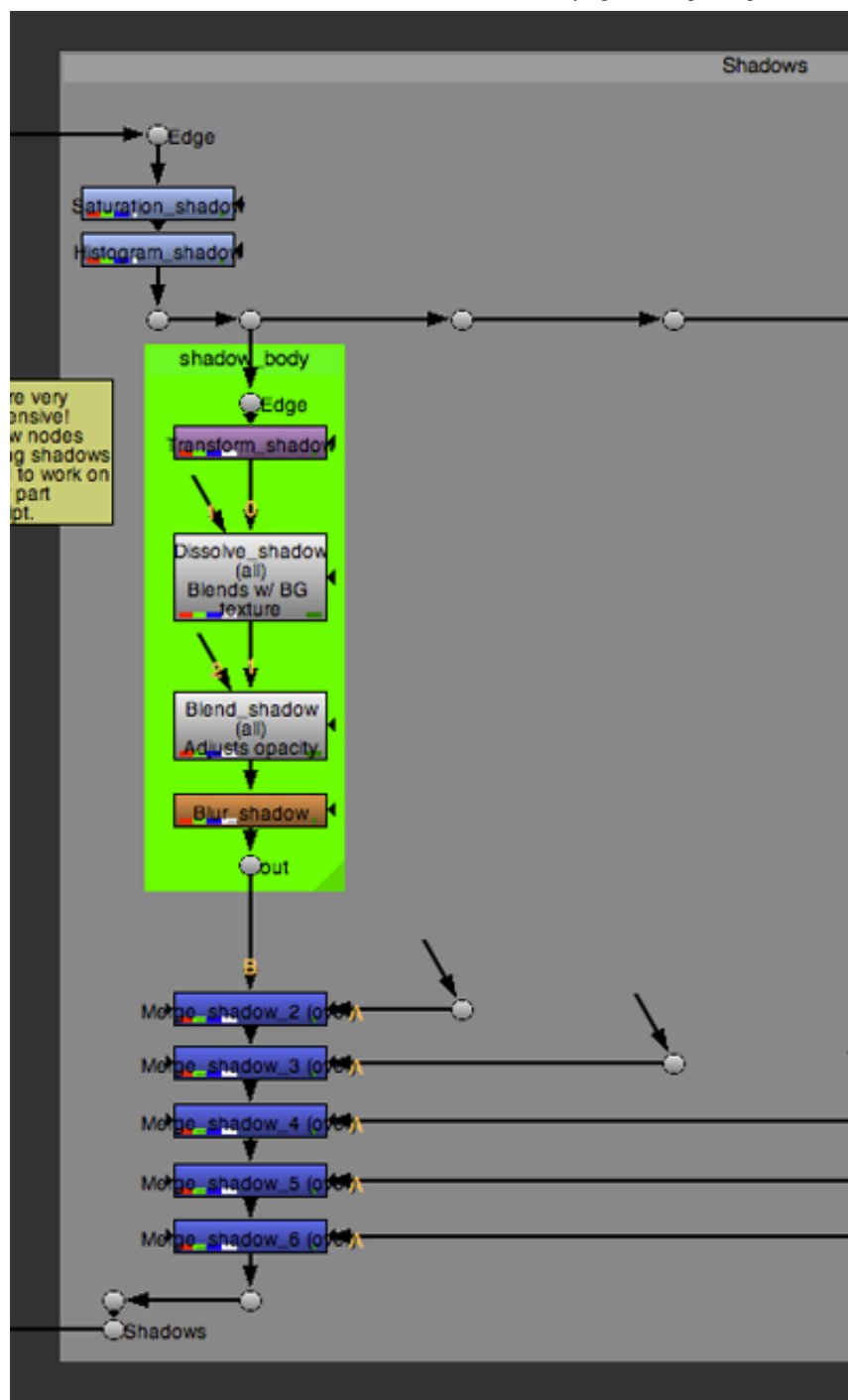
01_LS_bg.mov

49. Convert each shot's media into EXR image sequences in After Effects.
50. Move the **/~Nuke template and tutorial/shadows/01/01_LS_composite.nknc** script file to the **/Project Files/Nuke** directory.
51. Open the script file in NukeX.
52. Read in the foreground and background image sequences. Confirm that the project, viewer and background node's **frame.start at** frame ranges are all correct.
53. Set the Viewer node to the **Final Composite** dot (at the bottom of the script) to view a rough composite:

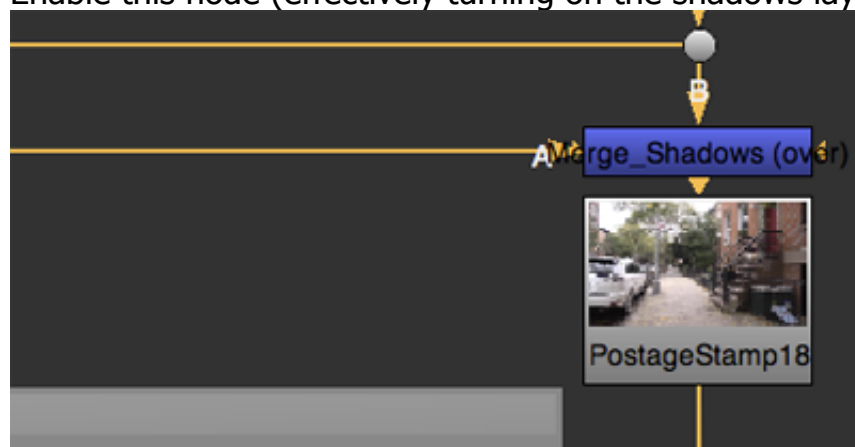


the foreground is comped in but has no shadows

54. Copy the **shadow_** section; rename the copy **shadow_body** and add it to the **Shadows** section.



55. The **Merge_shadows** node is near the top of the script, just before the Edge section. Enable this node (effectively turning on the shadows layer):



56. Adjust the shadows:

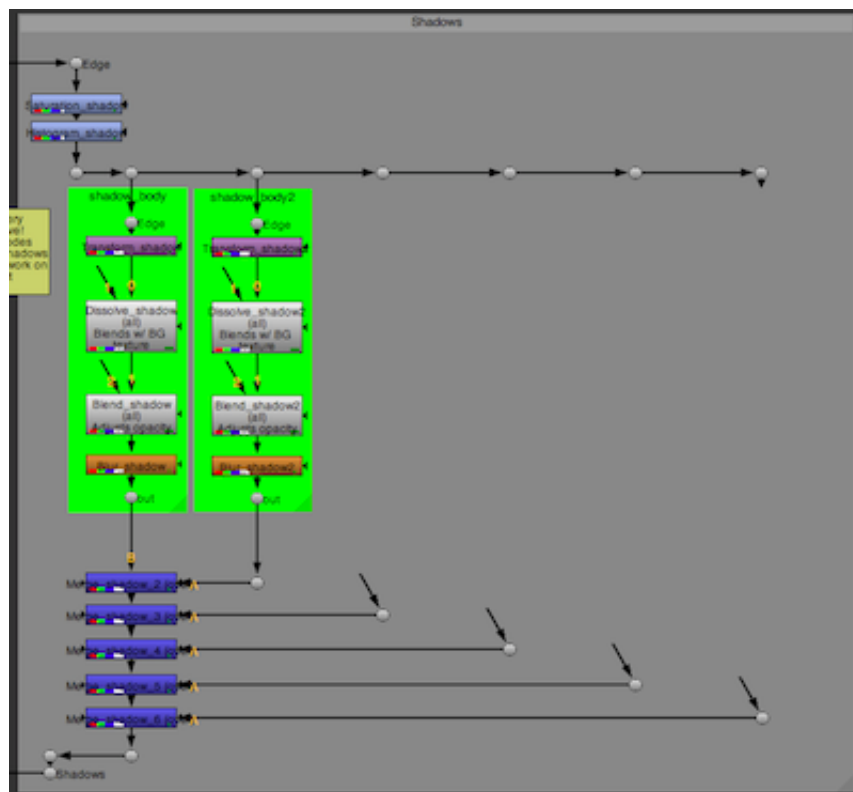
- a. In the **Shadows** section's **shadow_body** section, open the **Transform_shadow** node's properties panel. Using the transformation overlay in the Viewer, adjust the shadow so that it lays in the proper direction and is the proper size to appear as a shadow.



- b. Adjust the **Dissolve_shadow** node to blend the shadow with the background texture.
- c. Adjust the **Blend_shadow** node to adjust the shadow's opacity.
- d. Adjust the **Blur_shadow** node.



57. Add more **shadow_** sections for more shadows, if necessary. Name each section **shadow_[description]**. For example, the second body shadow section would be named shadow_body2. Merge each shadow with the previous one:



When completed, the shadows layer will look something like this:

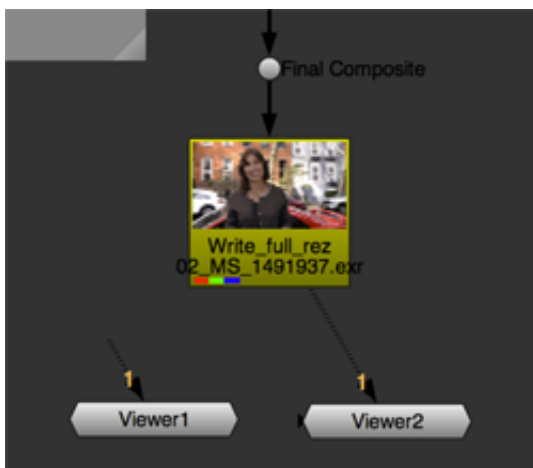


shadows layer composited on the background and behind the foreground layers

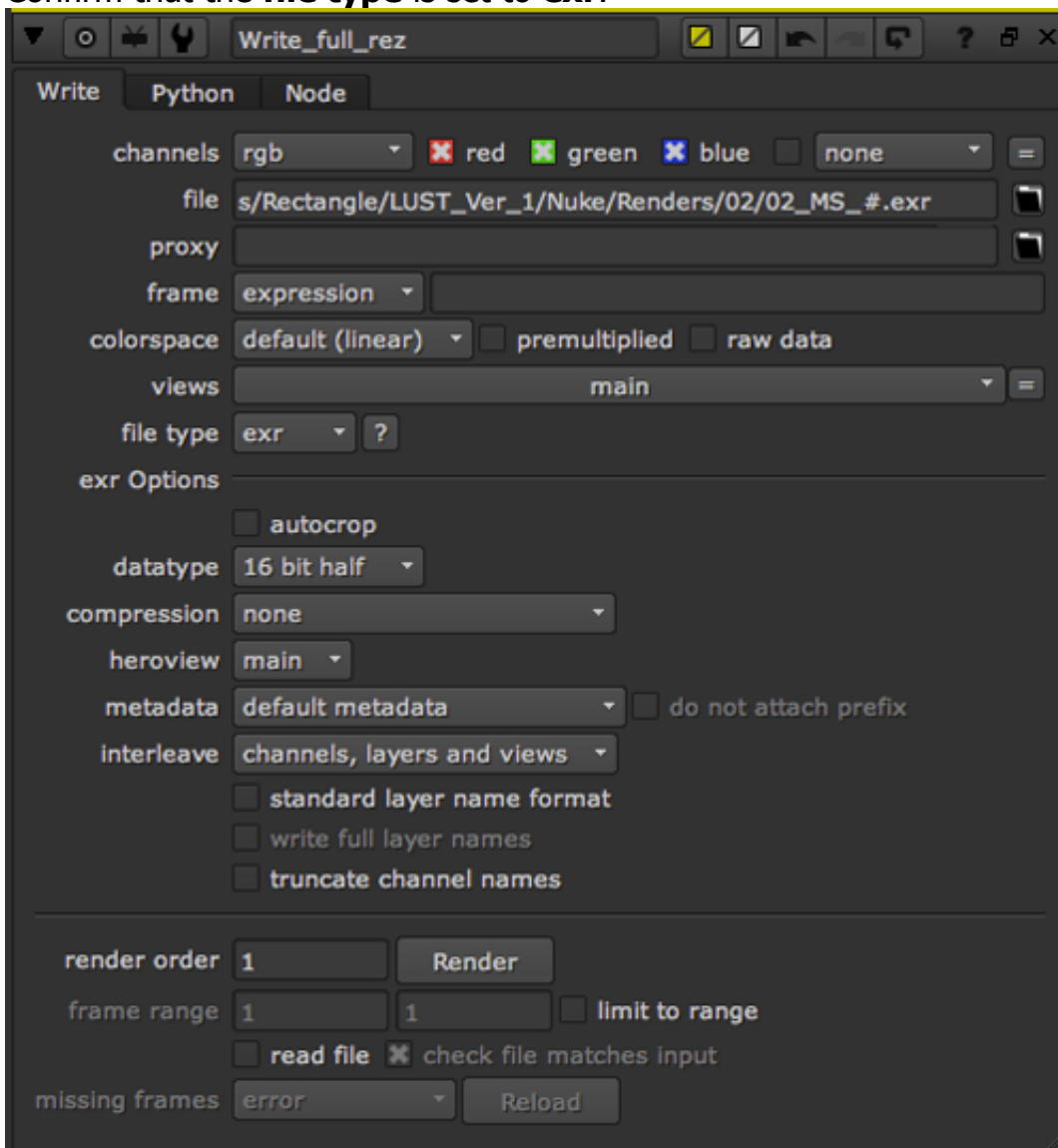
Rendering

58. When each shot is composited:

- a. Save the script.
- b. Add a **Write** node after the **Final Composite** dot at the bottom of the script.



- c. In the Write node's properties, navigate to the **/Nuke/Renders/[shot number]/**directory and set the file to **[shot number]_[description]_#.exr**. The # character is a frame number variable that inserts the frame numbers.
- d. Confirm that the **file type** is set to **exr**.

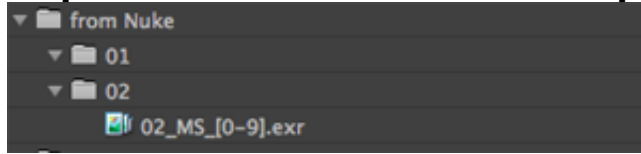


- e. Click the **Render** button; accept the defaults in the Render dialog box. The Progress bar or Progress panel will show the Render queue's progress.

Convert the EXR sequences to Quicktime

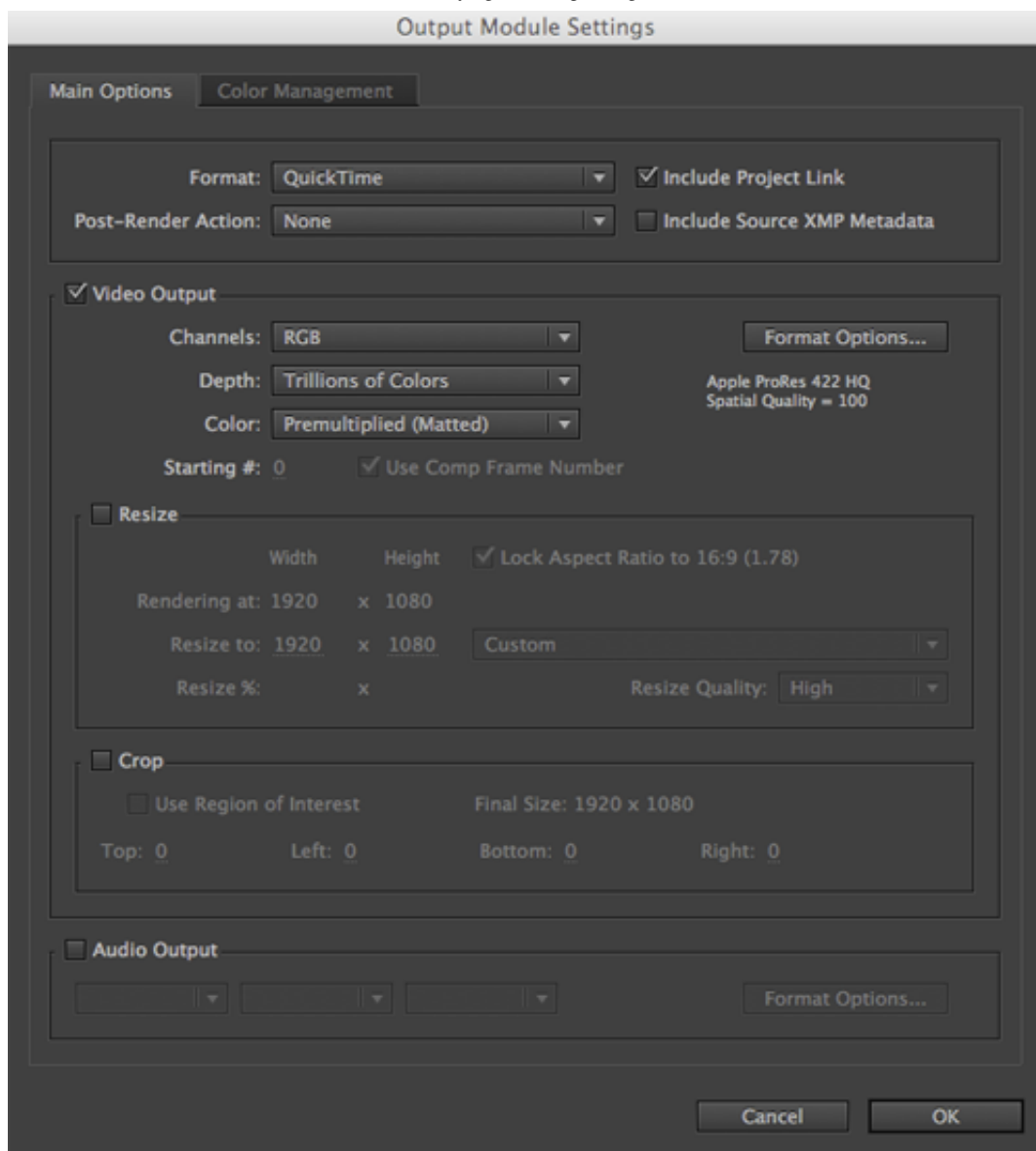
59. In After Effects, import the EXR sequence:

- a. Select the **from Nuke/[shot number]** folder.
- b. Select **File.Import.File**, navigate to the **/Nuke/Renders/[shot number]/** directory and select the first .exr file. Ensure that the **OpenEXR Sequence** checkbox is checked. Click **Open** to import the image sequence.



60. Render the image sequence:

- a. In Render Settings, confirm that the frame rate is correct.
- b. In Output Module settings, set the
 - Format: **Quicktime**
 - Format Options.Video Codec: **Apple PreRes 422 HQ**
 - Depth: **Trillions of Colors**
(this is the maximum; if the image sequence has only millions of colors, the Quicktime file will have the same)
 - Color: **Premultiplied**



- c. Set the Output to the project's **After Effects / Renders/EXR to QT conversions/** directory.
- d. Render the Quicktime movie.

61. Import the shot back into Final Cut Pro.

A plausible composite is not for the faint of heart

This is a long and very complicated process. You may be thinking "why in the world would I want to do this?" The best question is whether you need to have convincing, invisible compositing (for, say, a narrative film) or whether a collage look (which looks composited) is best for your project. Both are valid choices. However, if you absolutely, positively need to fool every single viewer in the room, I recommend this method and NukeX.

Don Starnes directs and photographs movies and videos of all kinds and is based in Los Angeles and the San Francisco Bay Area. He prefers real scenes, but shoots a lot green and blue screens.

www.donstarnes.com

Also by Don Starnes:



[Monitors and filmmaking](#)

The difference between a movie and a video is that a movie is created in the filmmakers' minds and a video is created on a monitor.



[Money](#)

It's the Producer's Invitational Pancake Breakfast. Just as the producers are cutting into short stacks with their plastic forks, all of the doors are locked...



[Blocking before coffee](#)

Feature film: first day. The first-time director is 45 minutes late. Finally, he shows up: harried, stubbly, preoccupied and exhausted



[Ask me no questions and I'll speak in no fragments](#)

A sign taped to the door says "Quiet-- filming." This only makes you more nervous.



[How to get trained](#)

Hint: it isn't by reading this.



[Preview: the Mini XTC 9250-XL](#)

Just in time for NAB, the 9250-XL is everything that a producer could want in a camera. The revolution has begun...



[DIY DCP](#)

How to make your own digital theatrical 'print' using Final Cut Pro, After Effects, guerrilla DCP software, pluck and maybe a little help from your friends.

[G+](#)