# NUKE 101

PROFESSIONAL COMPOSITING AND VISUAL EFFECTS

Second Edition

Ron Ganbar

# NUKE 101

## PROFESSIONAL COMPOSITING AND VISUAL EFFECTS

Second Edition

Ron Ganbar

Peachpit Press

**NUKE 101**

Professional Compositing and Visual Effects, Second Edition

**Ron Ganbar**

Peachpit Press
Find us on the Web at www.peachpit.com
To report errors, please send a note to errata@peachpit.com
Peachpit Press is a division of Pearson Education

**Senior Editor:** Karyn Johnson
**Senior Production Editor:** Lisa Brazieal
**Technical Editor:** Jonathan Mcfall
**Copyeditor:** Rebecca Rider
**Proofreader:** Emily K. Wolman
**Indexer:** Valerie Haynes Perry
**Interior Design:** Kim Scott, Bumpy Design
**Composition:** Danielle Foster
**Cover Design:** Aren Straiger
**Cover images:** *Goose* (2012) by Dor Shamir, Shai Halfon, and Oryan Medina

Footage from *This Is Christmas* and *Grade Zero* directed by Alex Norris, www.alexnorris.co.uk.

*Goose* footage (Chapter 3) by Dor Shamir, Shai Halfon, and Oryan Medina.

Keying footage (Chapter 7) from *Aya* by Divine Productions and Cassis Films. All rights reserved.

Chapter 9 footage by Or Kantor.

Pan and Tile panorama footage (Chapter 10) by Assaf Evron, www.assafevron.com.

Camera and geometry for Chapter 11 was solved by Michal Boico.

ISBN-13: 978-0-321-98412-8
ISBN-10: 0-321-98412-9

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

To The Foundry, who updates its products so often,
it makes me wish I hadn’t started down this road.

# CONTENTS

# INTRODUCTION

The Foundry's Nuke is fast becoming the industry leader in compositing software for film and TV. Virtually all the leading visual effects studios—ILM, Digital Domain, Weta Digital, MPC, Framestore, The Mill, and Sony Pictures Imageworks—now use Nuke as their main compositing tool. This is not surprising, as Nuke offers a flexible node-based approach to compositing, has a native multichannel workflow, and boasts a powerful integrated 3D compositing environment that delivers on the artist's needs.

Nuke was first developed as the in-house compositing tool at Digital Domain, the visual effects studio behind the *Terminator* series, *The Fifth Element*, *Tron: Legacy*, *The Curious Case of Benjamin Button*, and other major films. The software has been developed by artists for artists to meet the immediate needs of actual top-level productions. Nuke is now developed by The Foundry (www.thefoundry.co.uk), which remains committed to making Nuke the best tool for compositing artists working in the trenches.

## ABOUT THIS BOOK

Learning Nuke is a must for visual effects artists who want to master high-end compositing techniques and artistry. My goal with this book is to get you up and running with the program and give you the skills you need for doing your own compositing projects in Nuke.

### Whom this book is for

This book is for anyone interested in learning Nuke. Whether you're an artist experienced in using Adobe After Effects, Autodesk Flame, Apple Shake, or eyeon Fusion, or you have only a basic understanding of compositing and image manipulation concepts, this book will guide you through the necessary theory and practice you need to use Nuke—from a basic level to Nuke's more advanced toolset.

### How this book is organized

This book was written as a series of lessons, each focusing on a part of the interface, a tool, or a technique. Chapters 1 through 3 discuss Nuke basics, which are important for understanding where things are and how to create simple composites. Chapters 4 through 7 cover important tools and techniques. In Chapter 8 and onward, advanced tools and techniques are explained.

## What this book covers

This book teaches how to use Nuke from its very basic interface to its very advanced toolsets, including the 3D engine, Camera Projection, and Camera Tracking. Although the book teaches a fair amount of compositing theory, there is not enough space here to cover that topic in depth. Some of the theory discussed in this book may be new to you, but my intention is to cover just enough so you understand how to use Nuke. If you want to dive further into the theory, two of my favorite books are Ron Brinkmann's *The Art and Science of Digital Compositing* and Steve Wright's *Digital Compositing for Film and Video*.

## How to use this book

As you advance through the chapters in this book, the later lessons rely on knowledge you learned in the previous lessons. Chapter 2 relies on what you learned in Chapter 1, and so on. Because of this, I recommend completing the exercises in the chapters in order.

In the book you will find explanatory text and numbered steps. Ideally, you should **complete each numbered step exactly as it is written**—without doing anything else (such as adding your own steps). Following the steps exactly as written will give you a smooth experience. Not going through the steps as they are written might result in the next step not working properly, and could well lead to a frustrating experience. Each series of steps is also designed to introduce you to new concepts and techniques. As you perform the steps, pay attention to why you are clicking where you are clicking and doing what you are doing, as that will truly make your experience a worthwhile one.

You can use this book on your own through self-study or in a classroom.

- **Using the book for self-study:** If you're reading this book at your own pace, follow the instructions in the previous paragraph for your first read-through of the chapters. However, as you are not limited by any time frame, I recommend going through chapters a second time, and trying to do as much of the work as possible without reading the steps. Doing so can help you better understand the concepts and tools being taught. Also, the book leaves a lot of room for further experimentation. Feel free to use the tools you're learning to take your compositions further the second time you run through a chapter.
- **Using the book in a classroom setting:** You can use this book to teach Nuke in a classroom. As a course, the material is designed to run for roughly 40 hours, or five eight-hour days. I suggest that the trainer run through a chapter with the students listening and writing down notes; the trainer should explain the steps to the class as they are shown on screen while taking questions and expanding on the text where necessary. Once a chapter has been presented from start to finish, the instructor should give students time to run through the same chapter on their own in the classroom in front of a computer, using the book to read the

instructions and follow the steps. This second pass will reiterate everything the trainer has explained and, through actual experience, show the students how to use the software while the trainer is still there to answer questions and help when things go wrong.

## INSTALLING NUKE

While this book was originally written for Nuke version 8.0v1, The Foundry updates Nuke on a regular basis and the lessons can be followed using more recent updates. Small interface and behavior updates might slightly alter the Nuke interface from version to version, especially for so-called "point" updates (for example, if Nuke version 8.1 was released). I recommend using this book with Nuke version 8.0v1 if you haven't already downloaded the most current version and you want the exact results that are shown in the book.

You can download Nuke in a variety of versions from The Foundry's web site, www.thefoundry.co.uk, as discussed in the next sections.

### Different flavors of Nuke

Nuke comes in three different flavors with different features at different prices. There is only a single installation file for Nuke, but the license you purchase determines which type of Nuke you will be running. The Foundry offers a 15-day trial license, so you can try it before you purchase it (see "Getting a trial license," later in this section).

Here are the three flavors of Nuke.

1. **Nuke PLE (Personal Learning Edition):** This license (or lack of) is free—as in, you pay nothing. You can install Nuke on your computer and not purchase a license. With the PLE you can use Nuke as much as you want, although certain limitations apply. These include the placement of a watermark on the Viewer and on renders, and the disabling of WriteGeo, Primatte, FrameCycler, and Monitor Output. Keep in mind that Nuke project files saved with the PLE version can be opened only with the PLE version.

2. **Nuke:** This is regular Nuke—the flavor this book covers. Nuke requires a trial license or regular paid license, which should cost about $4,200.

3. **NukeX:** This license includes all the regular Nuke features with a few additional high-end tools. These tools include, among other things, the Camera Tracker, Particles System, PointCloudGenerator, LensDistortion, DepthGenerator, FurnaceCore plug-ins, and PRmanRender (allowing for RenderMan integration). NukeX costs around $8,150. Chapter 10 covers the Camera Tracker and shows how to use it under the NukeX license; however, the exercises in the chapter can also be done without a NukeX license.

### Downloading Nuke

To download Nuke, follow these steps.

1. Go to http://www.thefoundry.co.uk/products/nuke-product-family/nuke.
2. In the bar on the right, click Product Downloads.
3. Register with your email address and a password.
4. Select the latest copy of Nuke for your operating system (Mac, Windows, or Linux). You can also download older versions of Nuke if necessary.
5. Follow the instructions for installation on your specific operating system.

### Getting a trial license

After successfully installing Nuke, when you double-click the Nuke icon it explains that because you don't have a license yet, you can use Nuke under the PLE license. If you would like to use a fully licensed Nuke or NukeX, you will have to buy Nuke, rent Nuke (both available on The Foundry's web site shown below), or get a free 15-day trial license.

As there is no functional difference between getting a Nuke trial license or a NukeX trial license, I recommend getting a NukeX trial license. To get your free 15-day trial NukeX license, do the following:

1. Go to http://www.thefoundry.co.uk/products/nuke-product-family/nuke/trial.
2. In the Free Trial page, fill in the form and click Continue.

   The System ID, which is the first entry to fill in on this next page, is the unique code of your computer—the free license will be locked to that computer. The section below the entry field called Finding The System ID explains where to find this number on your computer.
3. After you complete the form and click Continue, follow the rest of the instructions on The Foundry's web site for how to install the license on your operating system.

## ADDITIONAL TECHNICAL REQUIREMENTS

Nuke is a very powerful piece of software, even though its system requirements are pretty low. If you bought your computer in the last couple of years, you are probably OK. The requirements are listed on The Foundry web site, but here are three things you should really check to make sure you have:

- Workstation-class graphics card, such as NVIDIA Quadro series, ATI FireGL series, R3D Rocket, or newer. Driver support for OpenGL 2.0.
- Display with at least 1280×1024 pixel resolution and 24-bit color.

- Three-button mouse. This kind of mouse is really a must as Nuke uses the middle mouse button extensively. A scroll wheel, by the way, can serve as the middle mouse button.

For a full list of Nuke’s system requirements, visit http://www.thefoundry.co.uk/products/nuke-product-family/nuke/sys-reqs.

## ABOUT THE MEDIA FILES

You can follow along with the exercises throughout this book by using the files that are available online. The files are a mix of real productions that my colleagues or I created in recent years and some shot elements I intended for use with this specific book.

To access the download and install the files on your computer, follow these steps:

1. Register your book at . If you don’t already have a Peachpit account, you will be prompted to create one.
2. Once you are registered at the Peachpit web site, click the Account link, select the Registered Products tab, and click the Access Bonus Content link that appears next to the Nuke 101 book image.

   A new page opens with the download files listed as individual zip files ordered by chapter number.
3. Download each zip file and copy them to your hard drive.
4. Unzip the lesson files for each chapter. Each chapter has its own directory. Some chapters use files from other chapters, so you need to unzip all the files.
5. Create a directory on your hard drive and name it **NukeChapters**.
6. Drag the chapter folders you unzipped into the NukeChapter directory (after you have done so, you can delete the zip files from your system).

## ACKNOWLEDGMENTS

I’ve been teaching compositing since 2001. When Nuke started becoming the tool of choice for a lot of the studios around me, I decided to write a course that focused on it. I started writing the course in the spring of 2009 with help from The Foundry, whose staff was very kind and forthcoming. I would specifically like to thank Vikki Hamilton, Ben Minall, Lucy Cooper, John Wadelton, and Matt Plec.

I finished writing the original course in the autumn of 2009. I taught it several times at Soho Editors Training in London; they were kind enough to let me try out the new course at their training facility. The course was well received, so between sessions I updated, corrected, and expanded on the original course.

About a year after that, I approached Peachpit Press with the idea of turning the course into a book. Karyn Johnson, the book's senior editor, took on the project and after a long digestion period I sat down and started adapting the course into a book. Karyn made sure I had the best support I could possibly have, and with the help of the wonderful team at Peachpit, including Corbin Collins and Kelly Kordes Anton, I managed to complete the book to the high standard that Peachpit expects of their writers. Thanks also go out to the kind friends and colleagues who gave me materials to use for the book: Alex Orrelle, Alex Norris, Hector Berrebi, Dror Revach, Assaf Evron, Menashe Morobuse, and Michal Boico.

For the second edition, started at the end of 2013, Karyn again stepped up and pushed me to make the book even more than it was before—striving for perfection. Rebecca Rider taught me a lot about how badly I write, fixing my English as much as I would let her. I also got help from a lot of friends and colleagues again (to add to the previous list): Oded Binnun, Dor Shamir, Or Kantor, Jonathan Carmona, Itay Greenberg, Paul Wolf, and Shani Hermoni. Wow. I owe a lot to a lot of people.

The book uses the following footage:

Alex Norris gave permission to use footage from two of his short film productions: *This Is Christmas* and *Grade Zero*. This is used in Chapters 1, 6, and 8.

The footage in Chapter 3 is taken from a personal short film called *Goose* by Dor Shamir (dorshamir.blogspot.com), Shai Halfon, and Oryan Medina (oryanmedina.com). Special thanks to Geronimo Post&Design. Jonathan Carmona did the 3D rendering specifically for this book.

The footage in Chapter 7 is taken from a short film called *Aya* directed by Mihal Brezis and Oded Binnun (who was also DP), starring Sarah Adler and Ulrich Thomsen, with thanks to production companies Divine Productions and Cassis Films.

The bullet in Chapter 8 was rendered especially for this book by Dror Revach.

The footage in Chapter 9 is taken from a yet unnamed short film by Or Kantor who wrote, designed, modeled, animated, and composited the whole thing. Thanks again to Jonathan Carmona, who helped take Or's files and render usable elements just for this book.

The footage for the panorama in Chapter 10 is by Assaf Evron.

The camera and geometry for Chapter 11 was solved by Michal Boico.

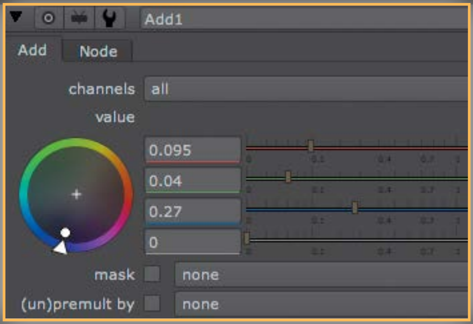
And finally: The second edition was a breeze to write compared to the first one. Maybe that was because I knew what I was getting into—but more important, my family knew what I was getting them into, and so things were a lot more relaxed. So thank you to my wife, Maya, and my two sons, Jonathan and Lenny, who had to bear with my long days and long nights of writing, gave me the quiet and solitude I needed, and believed (and prayed) that I would finish this second edition quickly so life could get back to normal. And so I have.

4

# COLOR CORRECTION

Wow. This is a bit naive, calling a lesson “Color Correction.” It should be a whole course on its own. But this book is about more than that, and limited space reduces color correction to a single chapter. So let me start by explaining what it means.

*Color correction* refers to any change to the perceived color of an image. Making an image lighter, more saturated, changing the contrast, making it bluer—all of this is color correction. You can make an image look different as the result of a stylistic decision. But you can also color correct to combine two images so they feel like part of the same scene. This task is performed often in compositing when the foreground and the background need to have colors that work well together. However, there are plenty more reasons you may change the color of an image. An image might be a mask or an alpha channel that needs to have a different color—to lose softness and give it more contrast, for example.



Add1

Add | Node

channels: all

value: 0.095, 0.04, 0.27, 0

mask: none

(un)premult by: none

Whatever reason you have for color correcting an image, the process will work according to the way Nuke handles color. Nuke is a very advanced system that uses cutting-edge technology and theory to work with color. Therefore, it is important to understand Nuke's approach to color so you understand color correcting within Nuke.

## UNDERSTANDING NUKE'S APPROACH TO COLOR

Nuke is a 32-bit float linear color compositing application. This is a bit of a fancy description, with potentially new words. I explain this bit by bit.

- **32 bit:** That's the number of bits used to hold colors. Most compositing and image-manipulation programs are 8-bit, allowing for 256 variations of color per channel (resulting in what's referred to as "millions of colors" when the three color channels are combined). 8-bit is normally fine for displaying color but is not good enough for some calculations of operations and may produce unwanted results such as *banding*—an inaccurate display of gradients where changes in color happen abruptly instead of smoothly. 32-bit allows for a whopping 4,294,967,296 variations of color per channel. That's a staggering number that results in *much* more accurate display of images and calculations of operations. 8- or 16-bit images brought into Nuke will be bumped up to 32-bit, although that doesn't add any detail, it just enables better calculations from that point onward.

- **Float:** Normally the color of an image is represented between black and white. In 8-bit images, for example, the 256 color variations are split evenly between black and white—so the value 1 is black, the value 256 is white, and the value 128 is a middle gray. But what about colors that are brighter than white? Surely the whiteness in the middle of a lit lightbulb is brighter than a white piece of paper? For that reason, colors that are brighter than white are called super-whites. Also, colors that are darker than black are called sub-blacks (but I can't think of a real-world analogy that I can use here short of black holes). Using 8 bits to describe an image simply doesn't allow enough room to describe colors beyond black and white. These colors get *clipped* and are simply represented as black or white. However, in 32-bit color, there is plenty of room and these colors become representable. As mentioned before, 8-bit color is normally enough to display images on screen. Furthermore, the computer monitor can still display only white—and nothing brighter. However, it is still very important to have access to those colors beyond white, especially when you're color correcting. When you're darkening an image that has both a piece of white paper and a lightbulb in it, you can leave the lightbulb white, while darkening the paper to a darker gray color, resulting in an image that mimics real-world behavior and looks good and believable. Doing the same with a nonfloating image results in the white paper and the lightbulb appearing the same gray color—which is unconvincing.

- **Linear:** Linear can mean lots of things, but here, in terms of color, I mean *linear color space*. A computer monitor doesn't show an image as the image appears in reality, because the monitor is not a *linear* display device. It has a mathematical curve called *gamma* that it uses to display images. Different monitors can have different curves, but most often, they have a gamma curve called *sRGB*. Because the monitor is not showing the image as it appears in reality, images need to be "corrected" for this. This is usually done automatically because most image capture devices are applying an sRGB curve too, in the opposite direction. Displaying a middle gray pixel on a monitor shows you only middle gray as it's being affected by the gamma curve. Because your scanner, camera, and image processing applications all know this, they color correct by applying the *reverse* gamma curve on this gray pixel that negates the monitor's effect. This process represents basic *color management*. However, if your image's middle gray value isn't middle gray because a gamma curve has been applied to it, it will react differently to color correction and might produce odd results. Most applications work in this way, and most people dealing with color have become accustomed to this. This is primarily because computer graphics is a relatively new industry that relies on computers that, until recently, were very slow. The correct way to manipulate imagery—in whatever way—is *before* the gamma curve has been applied to an image. The correct way is to take a linear image, color correct it, composite it, transform it, and then apply a reverse gamma curve to the image to view it correctly (as the monitor is applying gamma correction as well and negating the correction you just applied). Luckily, this is how Nuke works by default.

Still confused? Here's a recap: Nuke creates very accurate representations of color and can store colors that are brighter than white and darker than black. It also calculates all the compositing operations in linear color space, resulting in more realistic and more mathematically correct results.

**NOTE** Nuke color values are displayed and calculated in what's called *normalized values*. This means that instead of defining black at a value of 0 and white at a value of 255, black is still 0, but white is 1. It's a very easy thing to remember that makes understanding the math easier.

Nuke has many color correction nodes, but they are all built out of basic mathematical building blocks, which are the same in every software application. The next section looks at those building blocks.

## COLOR MANIPULATION BUILDING BLOCKS

Color correction is a somewhat intuitive process. Often compositors just try something until they get it right. Understanding the math behind color correction can help you pick the right tool for the job when you're attempting to reach a specific result—which is better than trial and error. **TABLE 4.1** explains most of these building blocks.

**TABLE 4.1** Basic Color Correction Functions

| Math Function | Node | Explanation | Names in Nuke | Other Known Names |
|---|---|---|---|---|
| Add | Add, Grade, ColorCorrect | Adds a constant value to a channel. | Add, Offset | |
| Multiply | Multiply, Grade, ColorCorrect | Multiplies the channel by a constant value. | Gain, Multiply | Brightness, Contrast, Exposure, Input/ Output White |
| Gamma | Gamma, Grade, ColorCorrect | Applies a gamma curve to a channel. | Gamma | |
| Contrast | RolloffContrast, ColorCorrect | Applies a contrast curve to a channel. This is also a form of multiplication. | Contrast, RolloffContrast | |
| Lift | Grade | This function is similar to Multiply and Contrast. It's a contrast curve with a center at white. More later in this chapter. | Lift | Pedestal, Input/Output Black |
| Lookup | ColorLookup | Applies a user-defined curve to a channel. | ColorLookup | Curves |
| Saturation | | Adjusts the color intensity by reducing the differences between the RGB channels. | Saturation | |

## Dynamic range

When dealing with color correction, I usually talk about dynamic range and its parts. *Dynamic range* means all the colors that exist in your image, from the darkest to the brightest color. The dynamic range changes from image to image, but usually you are working with an image that has black and white and everything in between. The parts of the dynamic range, as mentioned, are split according to their brightness value as follows:

- The shadows or lowlights, meaning the darkest colors in the image
- The midtones, meaning the colors in the image that are neither dark nor bright
- The highlights, meaning the brightest colors

In Nuke, and in other applications that support colors beyond white and black (float), there are two more potential parts to the dynamic range: the super-whites and the sub-blacks.

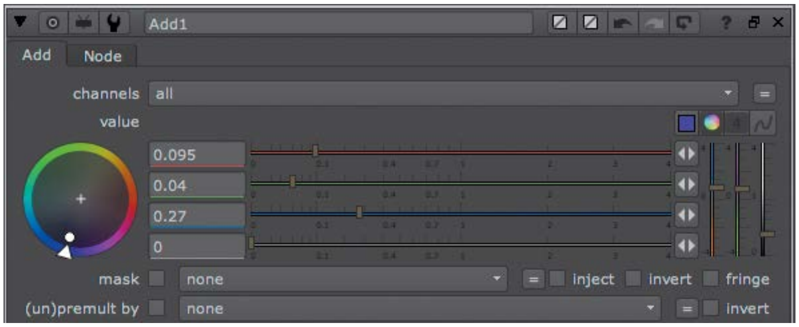
Let's look at these building blocks in several scenarios to really understand what they do and why you might choose one over another.

1. Launch Nuke.

2. Bring in a clip called car.png by pressing R and navigating to the chapter04 directory.
3. Click Read1, then press 1 on the keyboard to view it.

   It's an image of a car. Did that catch you by surprise?
4. With Read1 selected, go to the Color toolbox and click Add in the Math folder.

   You have now inserted a basic color-correcting node after the car image. Let's use it to change the color of the image and see its effect.
5. In Add1's Properties panel, click the Color Picker button to display the In-panel Color Picker. Play with the R, G, and B colors to see the changes (**FIGURE 4.1**).



channels all

value 0.095 0.04 0.27 0

mask none inject invert fringe

(un)premult by none invert

**FIGURE 4.1** Using the In-panel Color Picker

You can see that everything changes when you're playing with an Add node—the highlights, midtones, and even blacks (**FIGURE 4.2**). An Add operation adds color to everything uniformly—the whole dynamic range. Every part of the image gets brighter or darker.

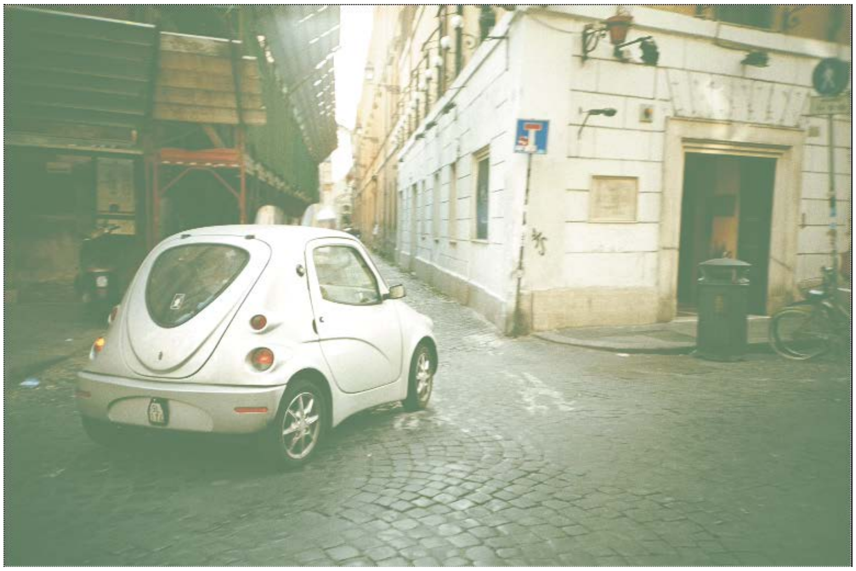


**FIGURE 4.2** The whole image is becoming brighter.

6. When you're finished, close the In-panel Color Picker.
7. Select Read1 again and branch out by holding the Shift key and clicking a Multiply node from the Math folder in the Color toolbox.
8. While Multiply1 is selected, press 1 on the keyboard to view it.
9. In Multiply1's Properties panel, click the Color Picker button to display the In-panel Color Picker and experiment with the colors (**FIGURE 4.3**).

   You can see very different results here. The highlights get a strong boost very quickly whereas the blacks are virtually untouched.

10. Repeat the previous process for the Gamma node. Remember to branch from Read1 (**FIGURE 4.4**).

   You can see that gamma deals mainly with midtones. The bright areas remain untouched and so do the dark areas.

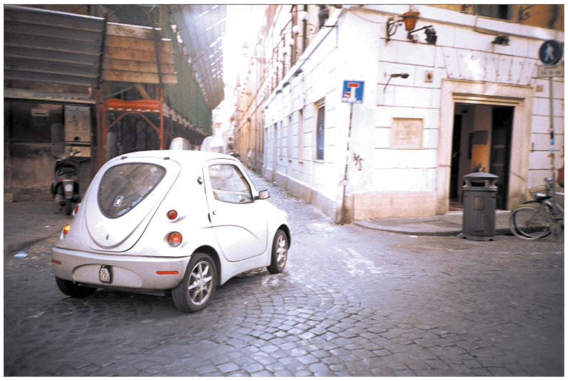


**FIGURE 4.3** The changes affect the highlights more than the rest of the image.

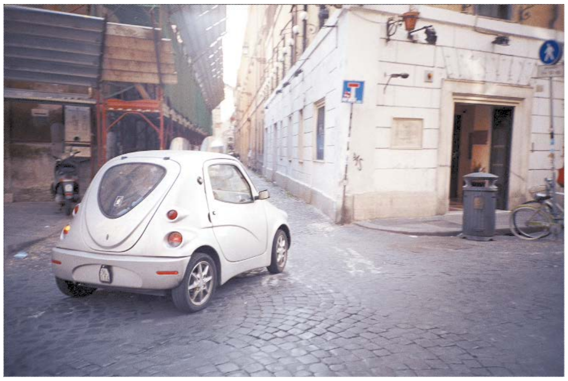


**FIGURE 4.4** The midtones change the most when you're changing gamma.

You should now have three different, basic, math-based color correctors in your Node Graph that produce three very different results, as shown in **FIGURE 4.5**.

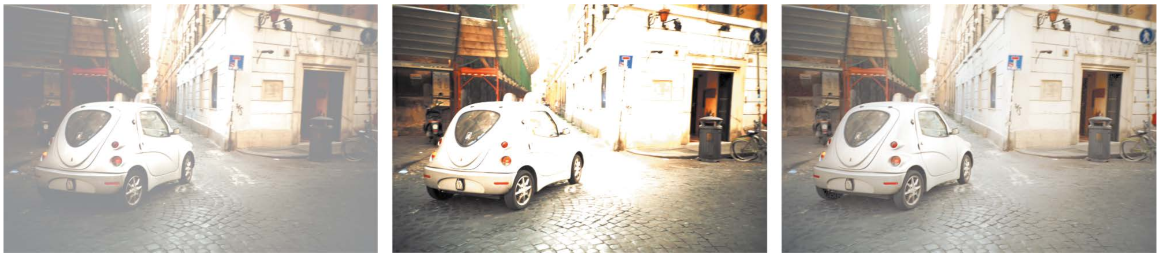


**FIGURE 4.5** The results from changing Add, Multiply, and Gamma

Your DAG should look a little like **FIGURE 4.6**.

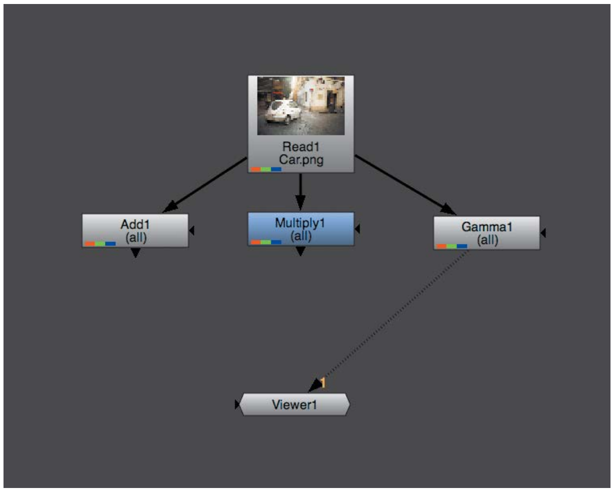


**FIGURE 4.6 Branching three color correctors from a node**

**NOTE** I find it really annoying that they chose to call the Contrast node RolloffContrast, especially since it makes opening it via the Tab key so much harder because typing "contrast" won't display this node.

Let's try some more color correction nodes.

11. Select Read1 and then Shift-click RolloffContrast in the Color toolbox to create another branch.
12. While viewing RolloffContrast1, open its Properties panel and play with the Contrast value (**FIGURE 4.7**).

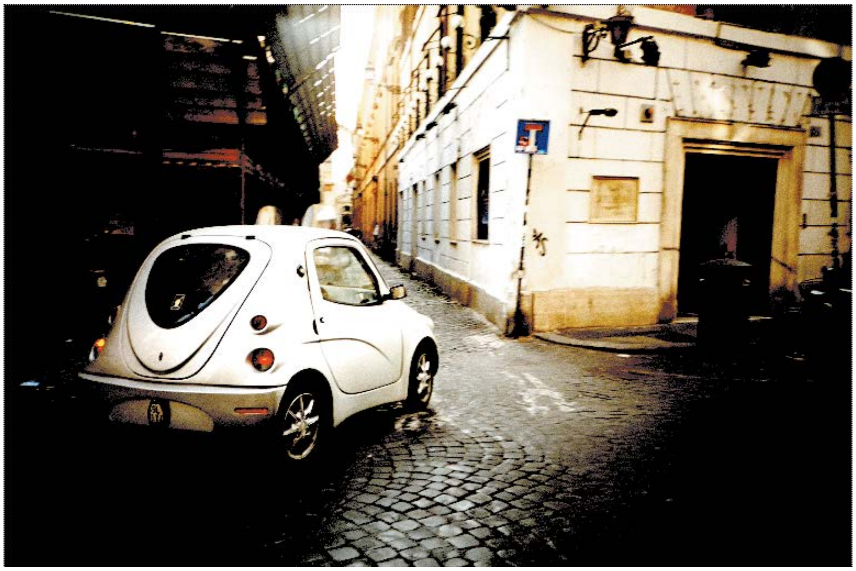


**FIGURE 4.7 A high contrast value produces a high contrast image.**

You can see how, when you increase the contrast above 1, the lowlights get pushed down and the highlights are pushed up.

13. Keep the Contrast property above 1 and bring the Center value down to 0.

    The Center property changes what is considered to be the highlight or lowlight. Colors above the Center value are considered bright and are pushed up, and colors below the Center value are considered dark and are pushed down.

    Now you can see that the result of the RolloffContrast operation is very similar to that of the Multiply node. In fact, they are virtually identical. When setting the center value at 0, you lock that value in place. The value 0 is locked in place when you're multiplying as well.

14. Bring the Center value up to 1.

You haven't gone through an operation called Lift yet, but the RolloffContrast operation is virtually the same as that operation. With Lift, the value 1 is locked in place, and the further the values are away from 1, the bigger the effect. You will go through Lift when you learn about the Grade node later in this chapter.

To wrap up this part of the color introduction, here's an overall explanation:

- When dealing with color, usually you need to control the lowlights, midtones, and highlights separately.
- The Add operation adds the same amount of color to every part of the dynamic range.
- The Multiply operation multiplies the dynamic range by a value. This means that a perfect black doesn't change, lowlights are barely touched, midtones are affected by some degree, and highlights are affected the most. It is good to mention that a Multiply operation is virtually the same as changing the exposure in a camera or increasing light. It is the most commonly used color operation.
- The Gamma control is a specific curve designed to manipulate the part of the dynamic range between 0 and 1 (black and white, remember?), without touching 0 or 1.
- Contrast is actually very similar to a Multiply, but has a center control. If you place the center point at 0, you get a Multiply node.

## USING AN I/O GRAPH TO VISUALIZE COLOR OPERATIONS

Studying an *I/O graph* (input versus output graph) is a great way to understand color operations. The X axis represents the color coming in, and the Y axis represents the color going out. Therefore a perfectly diagonal line represents no color correction. The graph shows what the color operation is doing and the changes to the dynamic range.

To view an I/O graph like this, you can bring in a premade script I made.

1. Choose File > Import Script to load another script from the disk and merge it with the script you have been building.
2. In the File Browser that opens, navigate to chapter04 and click IO_graph.nk to import it into your current script.

   Notice that when you imported the script (which is only four nodes), all of its nodes were selected. This is very convenient as you can immediately move the newly imported tree to a suitable place in your Node Graph.
3. Make sure the imported tree is not sitting on top of your existing tree. Move it aside to somewhere suitable, as in **FIGURE 4.8**.

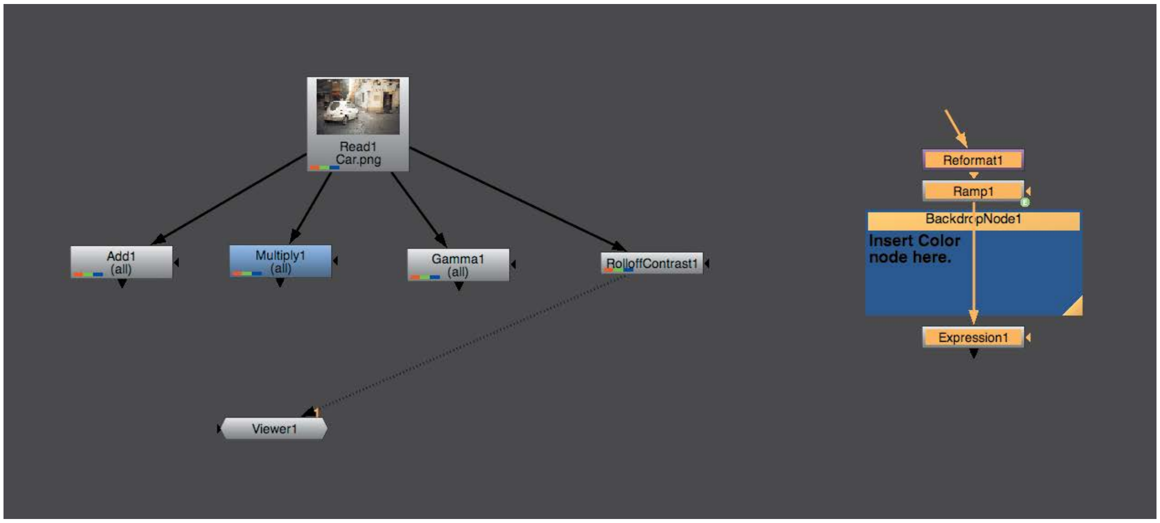


**FIGURE 4.8** **You now have two trees in your DAG.**

4. Make sure you are viewing the output of Expression1.

   Here is a quick explanation of the script you imported, node by node:

   - The first node is a Reformat node, which defines the resolution of your image—in this case, 256×256 pixels. Notice that its input isn't connected to anything. This is a good way to set a resolution for your tree.
   - The second node is a Ramp. This can be created from the Draw toolbox. This node generates ramps—in this case, a black to white horizontal ramp from edge to edge.
   - The third node is a Backdrop node used to highlight areas in the tree. You can find it in the toolbox called Other. It indicates where to add your color correction nodes in the next step.

- The fourth and last node is an Expression node, a very powerful node. It can be found in the Color > Math toolbox. It lets the user write an expression with which to draw or manipulate an image. You can do a lot of things with this node. From simple color operations (such as adding or multiplying, though this is wasteful) to complex warps or redrawing of different kinds of images all together. In this case, you use this node to draw on screen values of a horizontal black to white ramp (you have the ramp from above) as white pixels in the corresponding height in the image. A gray value of 0.5 in the ramp will generate a white pixel halfway up the Y resolution in the output of the Expression node. The leftmost pixel is black in the ramp and shows as a white pixel at the bottom of your screen. The middle pixel is a value of 0.5 and so it shows as a white pixel in the middle of the screen. The rightmost pixel has a value of 1 and so it draws a white pixel at the top of the screen. All these white pixels together form a diagonal line (**FIGURE 4.9**). Changing the color of the ramp will change the line. This happens on each of the three color channels individually.

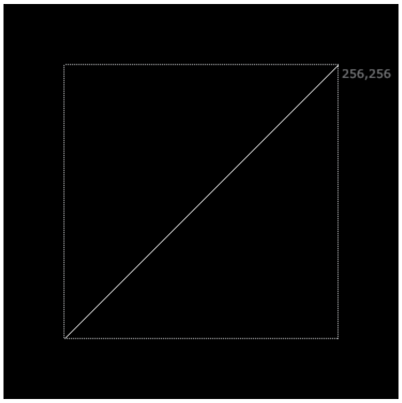


**FIGURE 4.9** The I/O graph in its default state

Let's start using this I/O Graph tree. You will insert a Color node in between Ramp1 and Expression1 and look at the resulting I/O graph.

5. Insert an Add node from the Color > Math toolbox after Ramp1, as shown in **FIGURE 4.10**.

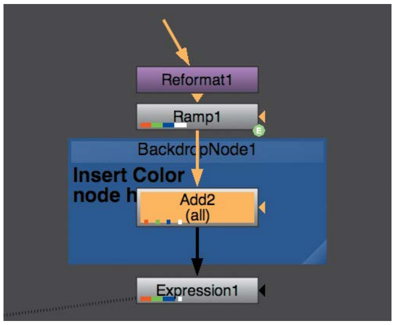


**FIGURE 4.10** Add2 has been inserted after Ramp1 and will change your I/O graph.

6. Bring the value of Add2's value property to around 0.1.

   You can see, as in **FIGURE 4.11**, that the Add operation changes the whole dynamic range of your graph and, therefore, for any image.

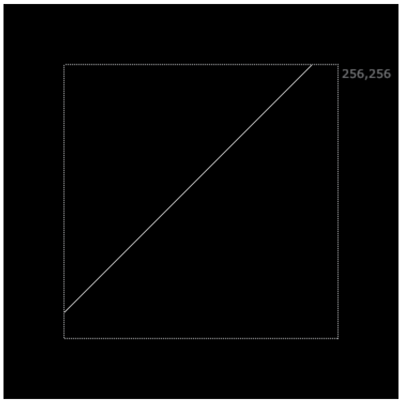


**FIGURE 4.11** **The whole graph is raised or lowered as a unit.**

   Let's replace your Add node with a Multiply node. You've never done this before, so pay attention.

7. With Add2 selected, Ctrl/Cmd-click the Multiply node in the Color > Math toolbox to replace the selected node with the newly created one.
8. Increase and decrease Multiply2's value.
9. You can also open the In-panel Color Picker and change the RGB channels individually (**FIGURE 4.12**).

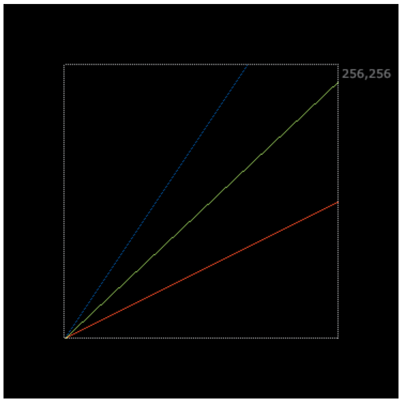


**FIGURE 4.12** **The graph changes more the further away it is from 0.**

   The Multiply operation has more effect on the highlights than the lowlights. When you are moving the slider, you can see that the 0 point stays put, and the further away you go from 0, the stronger the effect becomes.

Let's try gamma. Maybe you don't know what a gamma curve looks like. Well, here's your chance to learn.

10. Replace Multiply2 with a Gamma node from the Color or Math toolbox by holding down Ctrl/Cmd and clicking Gamma from the Color > Math toolbox.
11. Load Gamma2's In-panel Color Picker and play with the sliders for R, G, and B.

You should now get a similar result to **FIGURE 4.13**.

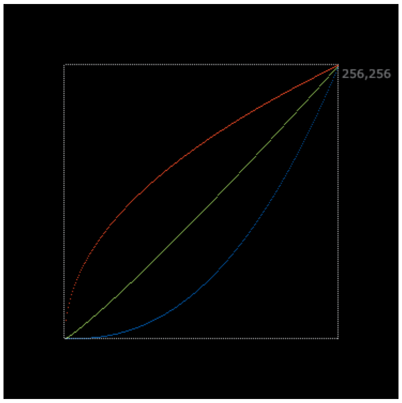


**FIGURE 4.13 Notice that only the middle part of the graph moves.**

The Gamma operation changes the midtones without changing the blacks or whites. You can tell that the points at the furthest left and at the furthest right are not moving.

Contrast is next.

12. Replace Gamma2 with a RolloffContrast node in the Color toolbox.
13. Bring RolloffContrast2's contrast value to 1.5.

The contrast operation pushes the two parts of the dynamic range away from one another (**FIGURE 4.14**).

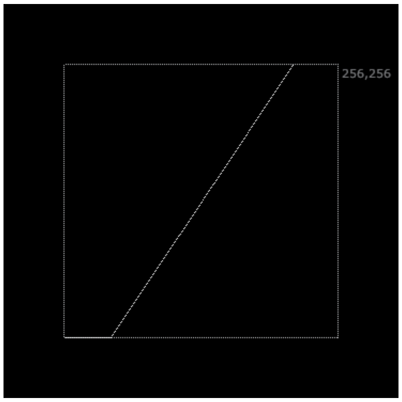


**FIGURE 4.14 A basic contrast curve. Though it's not curvy, it's still called a curve.**

14. Play around with RolloffContrast2's center property. When you are finished, set the value to 0.

    Here you can see what actually happens when you play with the center slider. It moves the point that defines where the lowlights and highlights are. When leaving the center at 0, you can see that the curve is identical to a Multiply curve (**FIGURE 4.15**).

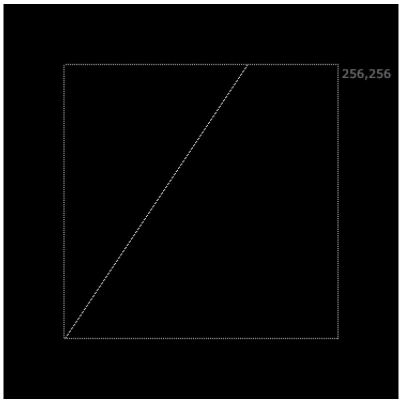


**FIGURE 4.15** A center value of 0 makes Contrast behave like Multiply.

15. Move the Center slider up to 1 (**FIGURE 4.16**).

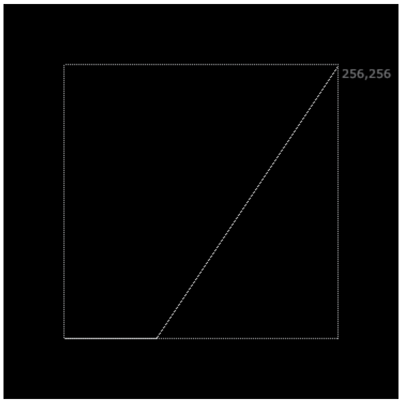


**FIGURE 4.16** Moving the slider up to 1 is actually a Lift operation.

This is a Lift operation, which is covered later in this chapter. Your white point is locked, while everything else changes—the opposite of Multiply.

RolloffContrast has one other property you can see in the I/O graph. This property, called Soft Clip, is the property that gives this node its name. This property smooths out the edges of the curve so that colors don't all of a sudden turn to black or white and result in a harsh transition.

16. Move the center slider to 0.5 and start to increase the Soft Clip slider. I stopped at 0.55.

    **FIGURE 4.17** shows what happens when you increase the soft clip. This creates a much more appealing result, which is unique to this node.

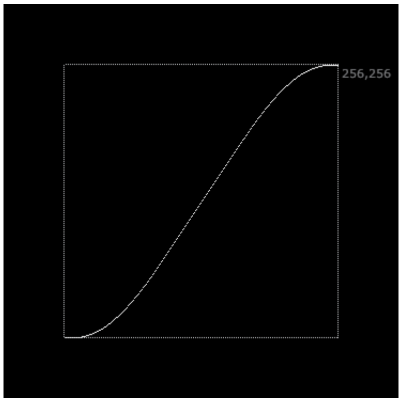


**FIGURE 4.17 This smooth edge to the curve is what gives RolloffContrast its name.**

If you have a fair amount of experience, you must have noticed that the I/O graph looks a lot like a tool you may have used in the past—something applications such as Adobe After Effects call Curves. In Nuke, this is called ColorLookup, and it is discussed in the next section.

## CREATING CURVES WITH COLORLOOKUP

The ColorLookup node mentioned at the beginning of this lesson is actually an I/O graph you can control directly. This makes it the operation with the most amount of control. However, it's actually the hardest to control and keyframe due to its more complicated user interface. After all, it's easier to set a slider and keyframe it than to move points on a graph.

Let's try this node on both the image and the I/O graph itself.

1. Replace RolloffContrast2 with a ColorLookup node in the Color toolbox (**FIGURE 4.18**).

   The interface for this node has the narrow curves list on the left and the curve area on the right. Choosing a curve at left displays that curve at right, which enables you to manipulate it. There are five curves. The first controls all the channels, and the next four control the R, G, B, and alpha separately. You can have more than one curve appear in the graph window on the right by Shift-clicking or Ctrl/Cmd-clicking them in the list.

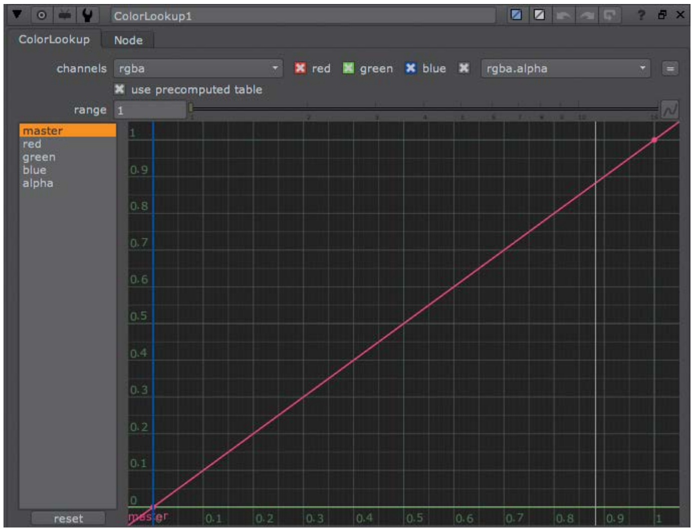


**FIGURE 4.18** The ColorLookup interface

2. Click the Master curve in the list at left.

   In the graph (Figure 4.18) you can now see a curve (a linear one at the moment). It has two points that define it, one at the bottom left and one at the top right. Moving them will change the color. For example, moving the top one will create a Multiply operation.

   The ColorLookup's strength lies in making curves that you can't create using regular math functions. However, to do this, you need to create more points.

3. To create more points on the curve, Ctrl/Cmd-Alt/Option-click the curve in the graph window. It doesn't matter where on the curve you click.

   You've just created another point. You can move it around and play with its handles. If you look at the I/O graph on the Viewer, you can see that it mimics what you did in the ColorLookup node. They are exactly the same (**FIGURE 4.19**).

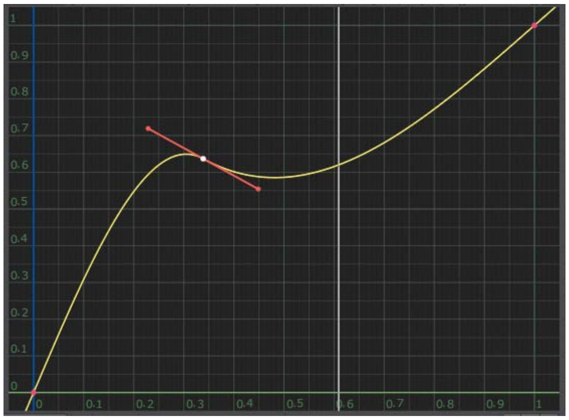


**FIGURE 4.19** Changing the curve is just like working with an I/O graph.

Now let's use ColorLookup on the car image.

4. Select Read1 and Shift-click the ColorLookup node in the Color toolbox to branch another output.
5. Click ColorLookup2 and press the 1 key to view it in the Viewer.
6. Play around with ColorLookup2's curves. You can play with the separate RGB curves as well.

I ended up with **FIGURE 4.20**—pretty drastic. But that's the level of control you have with ColorLookup. The Reset button at bottom left allows me to reset this mess.

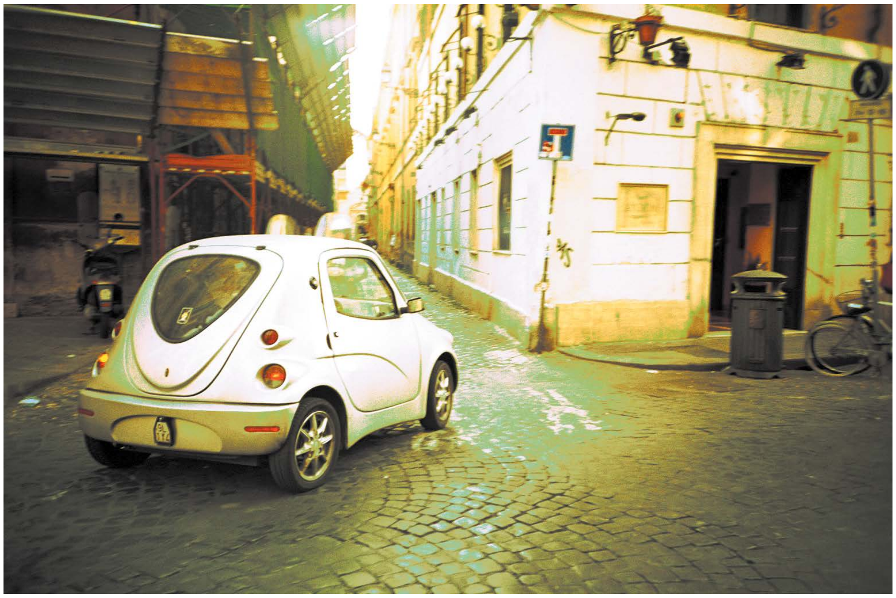


**FIGURE 4.20** Extreme color correction courtesy of ColorLookup

## COLOR MATCHING WITH THE GRADE NODE

The Grade node is built specifically to make some color correction operations easier. One of these operations is matching colors from one image to another.

When matching colors, the normal operation is to match black and white points between the foreground and background (only changing the foreground), then match the level of the gray midtones, and finally match the midtone hue and saturation.

## Using the Grade Node

The Grade node is made out of a few of the building blocks mentioned earlier. **TABLE 4.2** shows a list of its seven properties.

**TABLE 4.2 Grade Node Properties**

| Property | Definition |
|---|---|
| Blackpoint | This is the reverse operation of Lift. It works in the same way, but a higher number will result in stronger blacks instead of lighter blacks. Basically, the color chosen here becomes black. |
| Whitepoint | This is the reverse operation of Multiply. It works in the same way, but higher numbers will result in lower highlights instead of stronger highlights. Basically, the color chosen here becomes white. |
| Lift | A Lift operation. |
| Gain | A Multiply operation. |
| Multiply | Another Multiply operation. |
| Offset | An Add operation. |
| Gamma | A Gamma operation. |

By using Blackpoint and Whitepoint to set a perfect black and a perfect white, you can stretch the image to a full dynamic range. When you have a full dynamic range, then you can easily set the black point and white point to match those of the background using Lift and Gain. You then have Multiply, Offset, and Gamma to match midtones and for final tweaking.

Let's practice color matching, starting with a fresh script.

1. If you want, you can save your script. When you are finished, press Ctrl/Cmd-W to close the script and leave Nuke open with an empty script.

**NOTE** If Nuke quits altogether, just start it again.

2. From your chapter04 folder, bring in two images: CarAlpha.png and IcyRoad.png.
3. Make sure that CarAlpha.png is called Read1 and IcyRoad.png is Read2. You can change the name of a node in the topmost property.

   You will quickly composite these images together and then take your time in color matching the foreground image to the background.
4. Select Read1 and press the M key to insert a Merge node after it.

5. Connect Merge1's B input to Read2 and view Merge1 in the Viewer (**FIGURE 4.21**).

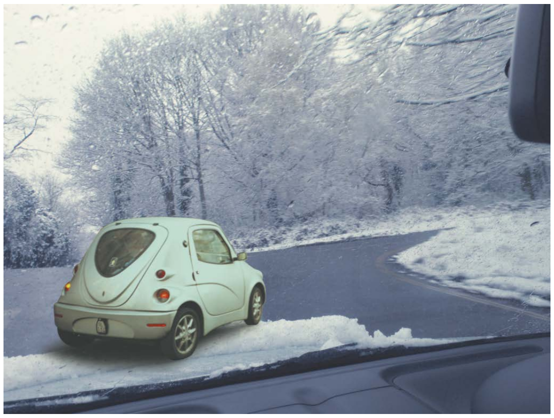


**FIGURE 4.21** The car is over the dashboard—this is wrong.

The composite is almost ready. You just need to punch a hole in the foreground car so it appears to be behind the snow that's piling up on the windshield. For that, you'll bring in another image (you will learn how to creates mattes yourself in Chapter 6).

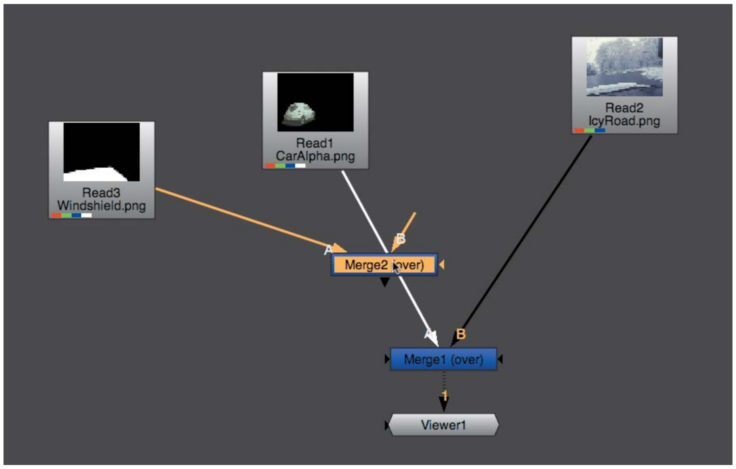
6. From your chapter04 folder, bring in Windshield.png and display it in the Viewer.

   Here you can see this is a matte of the snow. It is a four-channel image with the same image in the R, G, B, and alpha. You need to use this image to punch a hole in your foreground branch. To do this, you need another Merge node.

7. Select Read3 and insert a Merge node after it.

8. Drag Merge2 on the pipe between Read1 and Merge1 until the pipe highlights. When it does, release the mouse button to insert Merge2 on that pipe (**FIGURE 4.22**).

**FIGURE 4.22** Inserting a node on an existing pipe



9. View Merge1 (**FIGURE 4.23**).

   You can see here that this is not the desired result. You still need to change the Merge2 operation to something that will cut the B image with the A image. This operation is called Stencil. Stencil is used often to combine mattes in the same way we're using it now. The reverse of this operation, which is just as important, is called Mask, which masks the B image inside the A image's alpha channel. Mask holds image B inside the alpha channel of image A, and Stencil holds image B outside image A.

10. In Merge2's Properties panel, choose Stencil from the Operation drop-down menu (you can see the result in **FIGURE 4.24**).

    Looking at your comp, you can see that it now works—short of a color difference between the foreground and background. Let's use a Grade node to fix this shift.

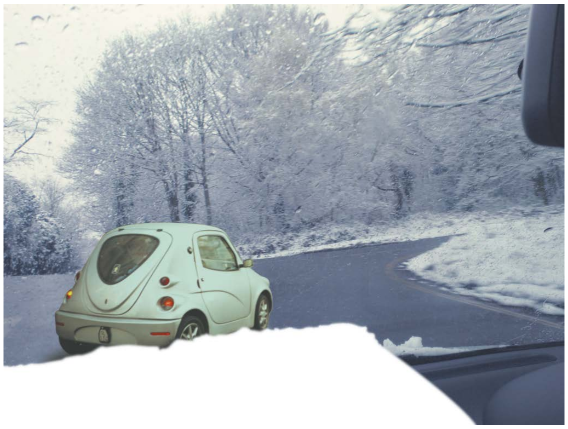


**FIGURE 4.23 All that white on the dashboard shouldn't be there.**

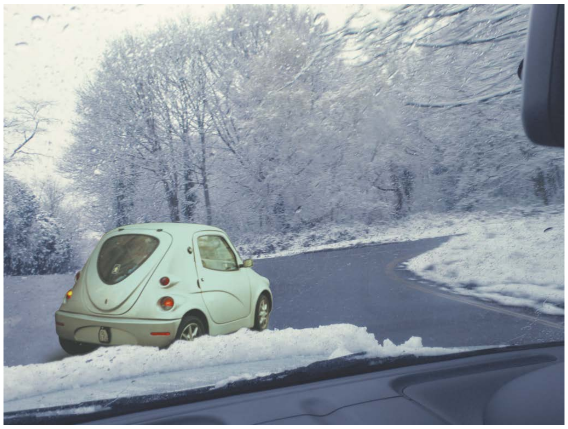


**FIGURE 4.24 The car is now correctly located behind the dashboard.**

11. Select Read1 and press the G key to insert a Grade node after it.

    As you know from Chapter 2, you are not allowed to color correct premultiplied images. It is often hard to tell if an image is premultiplied or not, but in this case it is. You can also look at the RGB versus the alpha channels and see that the areas that are black in the alpha are also black in the RGB.

    Since you can't color correct premultiplied images, you have to unpremult them. You can do this in one of two ways: using an Unpremult node before the color correction (in this case, Grade1) and then a Premult node after it, or using the (Un)premult By Switch in your Color nodes. Let's practice both.

12. Bring the Grade1's Offset property up to around 0.4 (**FIGURE 4.25**).

    You can see that the whole image, except the dashboard area, turned brighter, even though you are correcting only the car image. This is due to the lack of proper premultiplication. Let's do the two-node method first.

13. Click Read1 and, from the Merge toolbox, add an Unpremult node.

14. Click Grade1 and, from the Merge toolbox, add a Premult node and look at the Viewer (**FIGURE 4.26**).

    The problem has been fixed. This is one way to use proper premultiplication. Let's look at another.

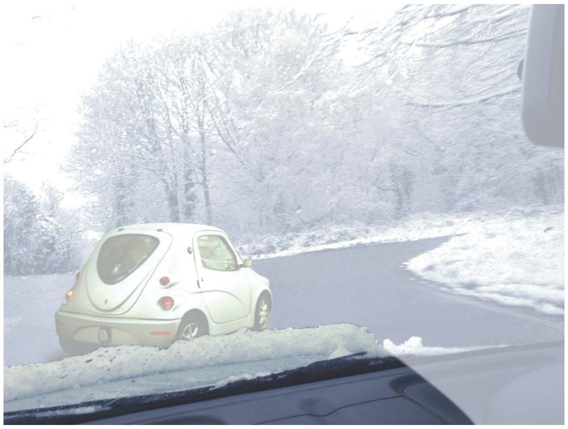


**FIGURE 4.25** The whole image turned brighter.

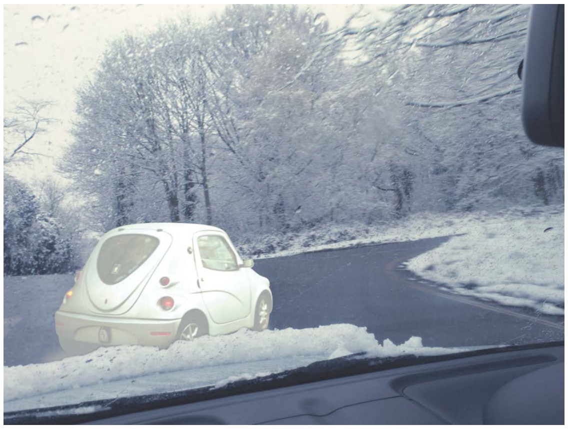


**FIGURE 4.26** The proper premultiplication fixed the problem.

15. Select Unpremult1 and Premult1, and press the Delete key.

16. In Grade1's Properties panel, choose rgba.alph from the (Un)premult By menu; this automatically selects the associated check box (**FIGURE 4.27**).

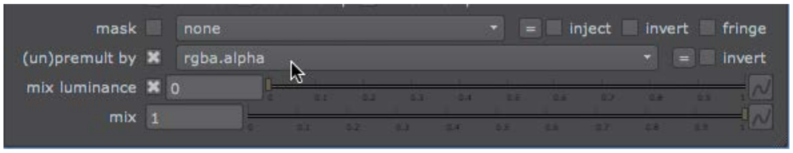

mask none = inject invert fringe
(un)premult by ✓ rgba.alpha = invert
mix luminance ✓ 0
mix 1

**FIGURE 4.27** Using the (Un)premult By property does the same thing as the Unpremult and Premult nodes workflow.

The resulting image looks exactly as it did before (in Figure 4.26). This technique does exactly the same thing as the first method, just without using other nodes. I usually prefer the first method, as it shows clearly in the DAG that the premultiplication issues are handled. However, if you look at Grade1 in the DAG now, you will see that, although the change is not as noticeable, Grade1 is showing that it is dividing the RGB channels with the alpha channel. The label now says "rgb/alpha" (**FIGURE 4.28**).

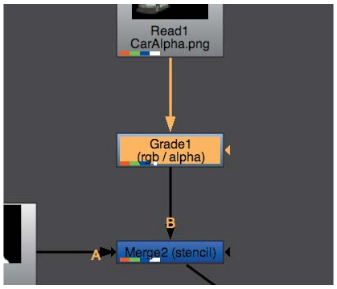


**FIGURE 4.28** The node's label changes to show the Unpremult and Premult operations are happening inside the node.

Let's use the second method you have set up already. You will now be color correcting an unpremultiplied image but outputting a premultiplied image.

After a little rearranging, the tree should look like the one in **FIGURE 4.29**.

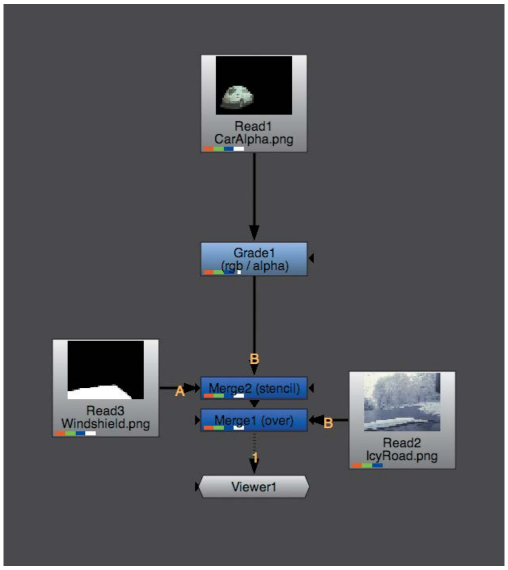


**FIGURE 4.29** Your tree should look like this at this point.

17. Bring the Offset property back to 0.

## Using CurveTool and Pixel Analyzer to match black and white points

Think back to the introduction of this section; how are you going to find the darkest and lightest points in these two images so you can match them together?

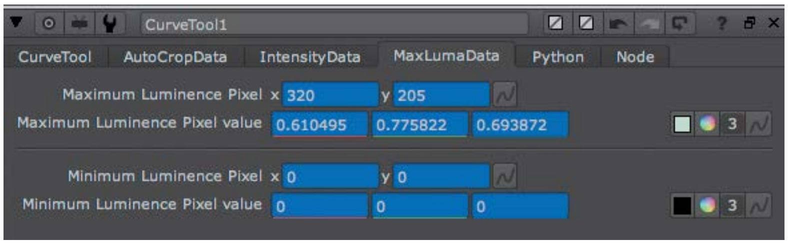
One way, which is valid and happens often, is by using your eyes to gauge which are the darkest and brightest pixels. However, the computer is so much better at these kinds of things, and it doesn't have to contend with light reflections on the screen and other such distractions.

The node to use for this is the CurveTool node, which you used in Chapter 3 to find the edges of the pilot element. You can also use this node to find out other color-related stuff about your image. Let's bring in a CurveTool node to gauge the darkest and brightest point in the foreground and use that data to stretch the foreground image to a full dynamic range.

1. Select Read1 and branch out by Shift-clicking a CurveTool node in the Image toolbox.

   This time you are going to use the Max Luma Pixel Curve Type. This finds the brightest and darkest pixels in the image.

2. In CurveTool1's Properties panel, switch the Curve Type drop-down menu to Max Luma Pixel.
3. While viewing CurveTool1 in the Viewer, click the Go! button.
4. In the dialog box that opens, click OK since you want to process only one frame.
5. Switch to the MaxLumaData tab and view CurveTool1 in the Viewer (**FIGURE 4.30**).



Maximum Luminence Pixel x 320 y 205
Maximum Luminence Pixel value 0.610495 0.775822 0.693872 3
Minimum Luminence Pixel x 0 y 0
Minimum Luminence Pixel value 0 0 0 3

**FIGURE 4.30** The MaxLumaData tab's two sections

The purpose of this operation is to find the darkest and lightest pixels in the image. When switching to this tab you see two sections, the one showing the lightest pixel (Maximum) and the darkest pixel (Minimum). For each, the X and Y location and the RGB values are displayed.

Looking closely, you can see that the value of the minimum pixel is 0 in every property. This is because this image is a premultiplied image, and as far as CurveTool is concerned, all that black in the image is as much a part of the image as any other part of it. You need to find a way to disregard that black area. Let's do the following.

6. From the Image toolbox, create a Constant node.

   A Constant node creates a solid color with a chosen resolution.

7. Change Constant1's Color value to 0.5.
8. Select Read1 and branch a Merge node from it by pressing Shift-M.

9. Connect Merge3's B input to Constant1, and then view Merge3 in the Viewer (**FIGURE 4.31**).

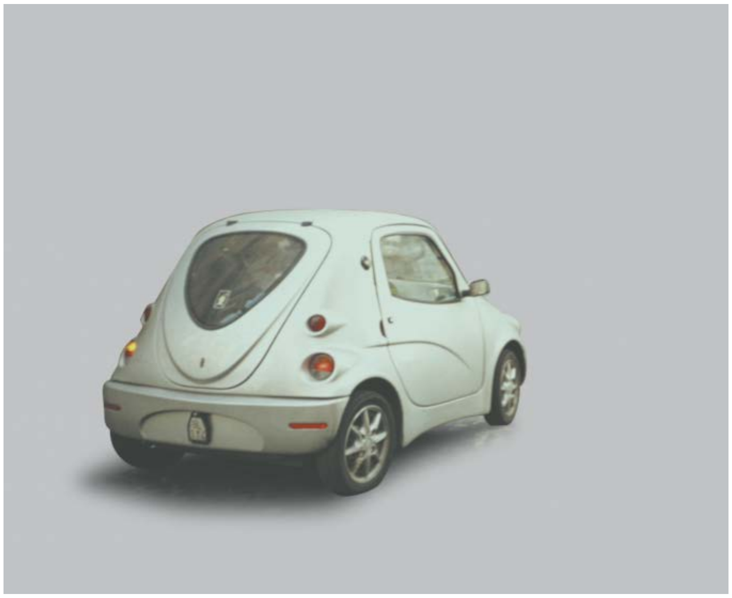


**FIGURE 4.31 The car is on a gray background.**

What you did here was replace, momentarily, the black background with a middle gray background. This way, you get rid of the black and replace it with a color that is not the darkest nor the lightest in the image. This new image is the image you want to gauge using the CurveTool. You'll need to move the pipe coming in to CurveTool1 (**FIGURE 4.32**).

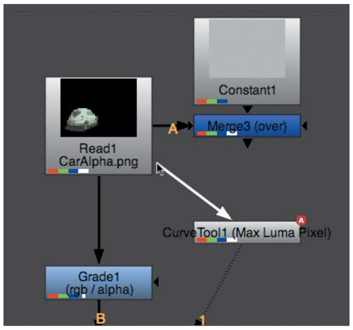


**FIGURE 4.32 Moving the pipe from Read1's output to Merge3's output**

10. Click the top half of the pipe going into CurveTool1, which will enable you to move it to the output of Merge3.

11. Double-click CurveTool1 to display its Properties panel in the Properties Bin. Switch to the CurveTool tab (the first one), click Go! again, and click OK.

12. Switch to the MaxLumaPixel tab again and have a look (**FIGURE 4.33**).

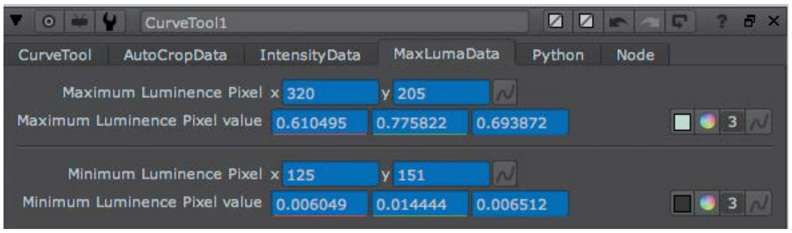


**FIGURE 4.33** The updated CurveTool1's MaxLumaData tab

CurveTool1

CurveTool | AutoCropData | IntensityData | MaxLumaData | Python | Node

Maximum Luminence Pixel x 320 y 205
Maximum Luminence Pixel value 0.610495 0.775822 0.693872 3

Minimum Luminence Pixel x 125 y 151
Minimum Luminence Pixel value 0.006049 0.014444 0.006512 3

Now you can see that the minimum values are far from being all 0. You are getting a true result that shows the lightest and darkest pixels. Let's make use of them.

13. Close all Properties panels in the Properties Bin to clear some room.

14. Double-click CurveTool1 and then double-click Grade1.

15. View Merge1 in the Viewer.

16. Click the 4 icon next to Grade1's Blackpoint, Whitepoint, Lift, and Gain to enable the four fields.

17. Ctrl/Cmd-drag from CurveTool1's Minimum Luminance Pixel value's Animation menu to Grade1's Blackpoint Animation menu and release the mouse button to create an Expression link between them.

18. Do the same from Maximum Luminance Pixel value to Whitepoint (**FIGURE 4.34**).

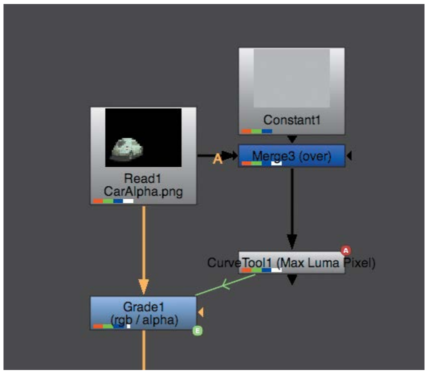


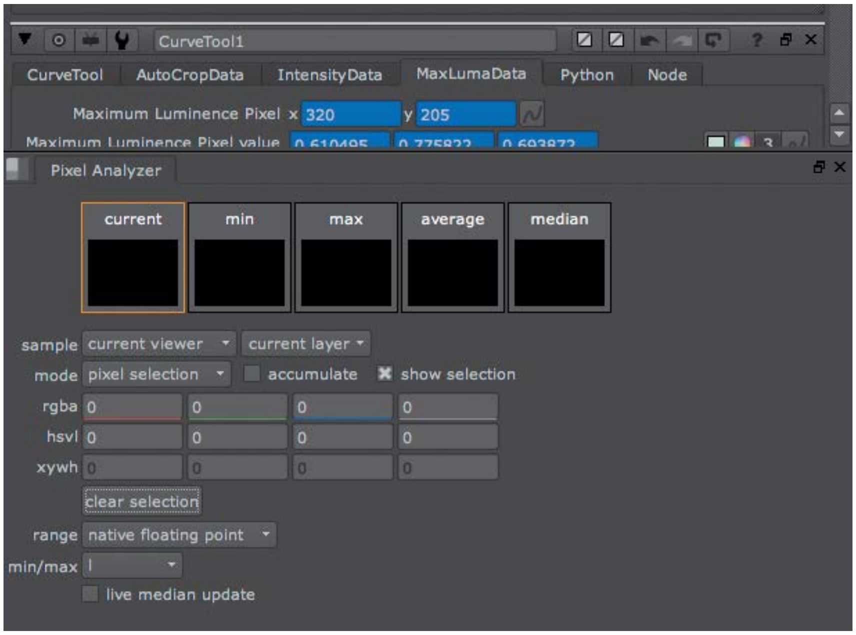
**FIGURE 4.34** The green arrow shows the Expression link between the two nodes.

The foreground image's dynamic range now spans from a perfect black to a perfect white. This enables you to push those colors to new black and white points to match these points to the background image. You can use another CurveTool to find those points in the background image, but just for fun, let's use the Pixel Analyzer for that this time.

The Pixel Analyzer is a new panel in Nuke 8.0. It helps you analyze the pixel values in your image.

19. From the Properties Bin’s Content menu, choose “Split Vertical”.

20. From the newly created pane’s Content menu, choose Pixel Analyzer (**FIGURE 4.35**).



Maximum Luminence Pixel x 320 y 205
Maximum Luminence Pixel value 0.610495 0.775822 0.693872

sample current viewer current layer
mode pixel selection accumulate show selection
rgba 0 0 0 0
hsvl 0 0 0 0
xywh 0 0 0 0
range native floating point
min/max l
live median update

**FIGURE 4.35** The Pixel Analyzer now lives at the bottom right of the interface.

This is the aforementioned Pixel Analyzer.

21. While holding Ctrl/Cmd, drag on the screen (**FIGURE 4.36**).

Notice this time around that a line of red dots appears on the screen? All those points accumulate to fill the five color boxes in the Pixel Analyzer with values.

The five boxes represent:

- Current: the last pixel dragged
- Min: the lowest or darkest value dragged
- Max: the highest or brightest value dragged
- Average: the average pixel value of all pixels dragged
- Median: out of all values dragged, the middle value

Clicking any of the boxes shows the values of that color below—RGBA and HSVL (Hue, Saturation, Value, Luminance).

Dragging on the screen is all well and good, but the whole frame is what you need to know about. There’s a feature for that too.

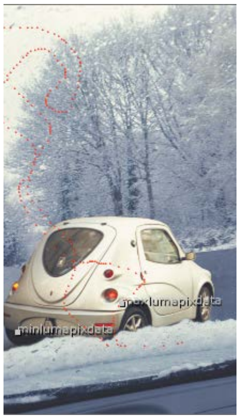


**FIGURE 4.36** Dragging on the screen creates this dotted line if the Pixel Analyzer is open.

22. View Read2 in the Viewer.

23. From the Mode menu, choose Full Frame.

    This option checks every pixel of the frame that's currently in the Viewer and returns the corresponding values in the five boxes. This is a really quick way to find your black point and white point in a given frame.

    You now have two sets of data to match to: new black points and white points. Let's copy them to your Grade node.

24. Close all Properties panels in the Properties Bin to clear some room.

25. Double-click Grade1.

    Because the Pixel Analyzer is a panel and not a node, you can't link to it, but you can very easily copy the values across from the Pixel Analyzer to the property where the values are needed by dragging.

26. Drag from the Pixel Analyzer's Min box to Grade1's Lift Color swatch to copy the values across (**FIGURE 4.37**).

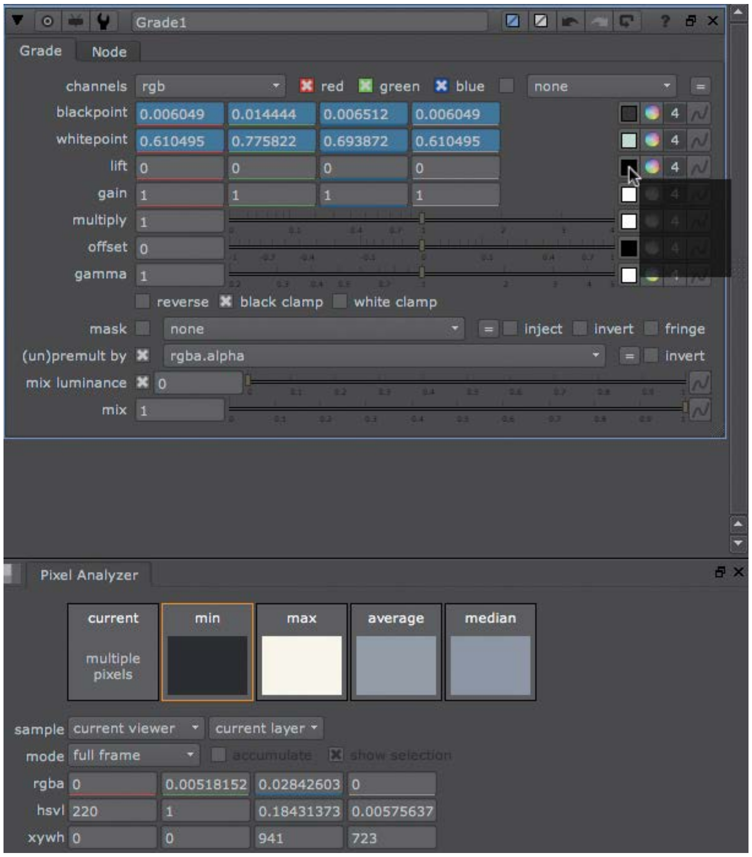


**FIGURE 4.37** Dragging from the Pixel Analyzer

Grade1

Grade | Node

channels rgb ✗ red ✗ green ✗ blue none

| | | | | |
|---|---|---|---|---|
| blackpoint | 0.006049 | 0.014444 | 0.006512 | 0.006049 |
| whitepoint | 0.610495 | 0.775822 | 0.693872 | 0.610495 |
| lift | 0 | 0 | 0 | 0 |
| gain | 1 | 1 | 1 | 1 |
| multiply | 1 | | | |
| offset | 0 | | | |
| gamma | 1 | | | |

reverse ✗ black clamp white clamp

mask none inject invert fringe

(un)premult by ✗ rgba.alpha invert

mix luminance ✗ 0

mix 1

Pixel Analyzer

current (multiple pixels) | min | max | average | median

sample current viewer current layer

mode full frame accumulate ✗ show selection

| | | | | |
|---|---|---|---|---|
| rgba | 0 | 0.00518152 | 0.02842603 | 0 |
| hsvl | 220 | 1 | 0.18431373 | 0.00575637 |
| xywh | 0 | 0 | 941 | 723 |

27. Do the same from the Max box to the Gain Color swatch.

28. You don't need the Pixel Analyzer anymore, so from its Content menu choose Close Pane.

29. View Merge1 in the Viewer.

    You have now matched the foreground's shadows and highlights to those of the background (**FIGURE 4.38**).

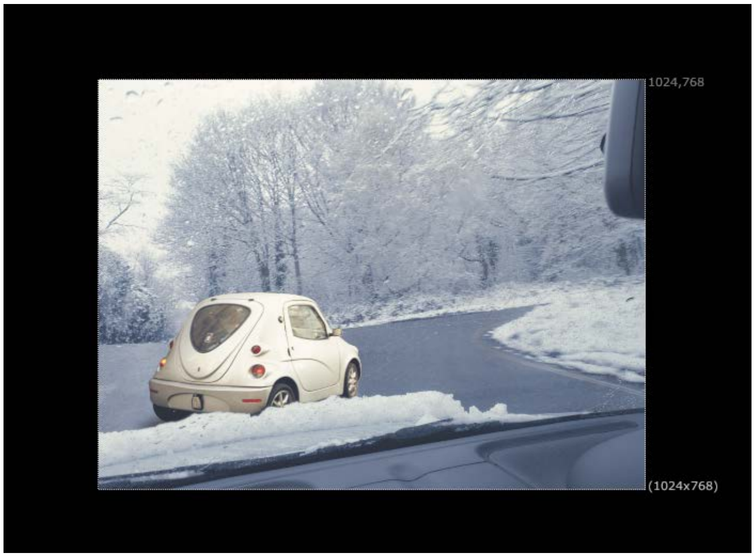


**FIGURE 4.38** Shadows and highlights now match.

As you can see from the image, the shadows and highlights are matched, but the image is far from looking matched. The midtones, in this case, make a lot of difference.

## Matching midtones by eye

You now need to match the midtones, which is a much more difficult task. You'll start by matching its luma level by eye. Because it is hard to tell what the midtones are, though, you are going to view the luminance of the image in the Viewer.

1. Hover your mouse pointer in the Viewer and press the Y key to view the luminance.

   To change the midtones now, use the Gamma property. You can see that the whitish snow on the right is a darker gray than the whitish car. Let's bring down the whitish car to that level.

2. Start dragging the Gamma slider down. I stopped at around 0.6.

   Notice that the midtones don't match well with a higher Gamma value. Now, however, the lower midtones aren't matching well. You need to use the Multiply property to produce a good match.

3. Bring the Gamma slider up to 0.85 and bring the Multiply slider down a bit to 0.9 (**FIGURE 4.39**).

4. Hover your mouse pointer in the Viewer and press the Y key to view the RGB channels (**FIGURE 4.40**).

   OK, so the midtones' brightness is better now, but you need to change the color of the car's midtones. At the moment, the car is too warm for this winter's day. Matching color is a lot more difficult because you always have three options: red, green, and blue. Matching gray is a lot easier because you need to decide only whether to brighten or darken it. However, as each color image is made out of three gray channels, you can use the individual channels to match color too. Here's how.

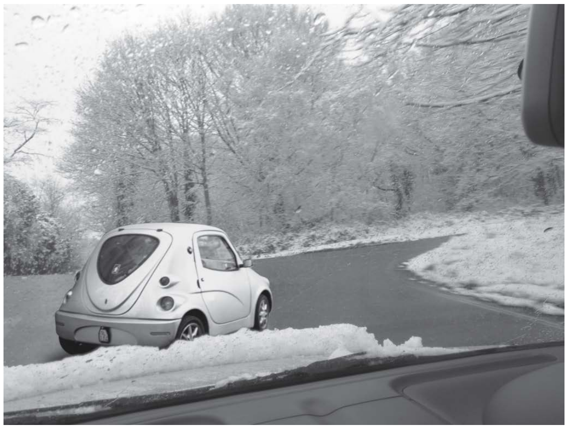


**FIGURE 4.39** The midtones match better now.

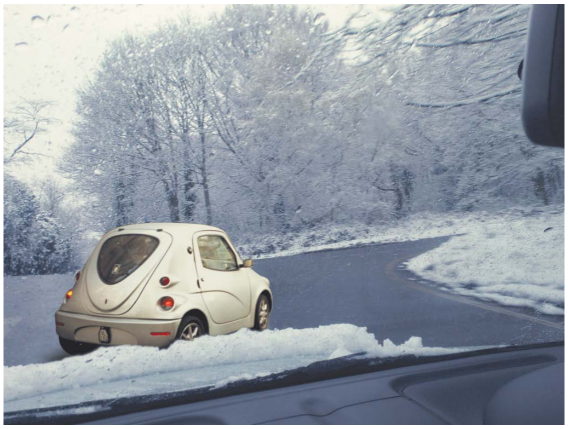


**FIGURE 4.40** You still have work to do on the color of the midtones.

5. Hover your mouse pointer in the Viewer and press the R key to view the red channel (**FIGURE 4.41**).

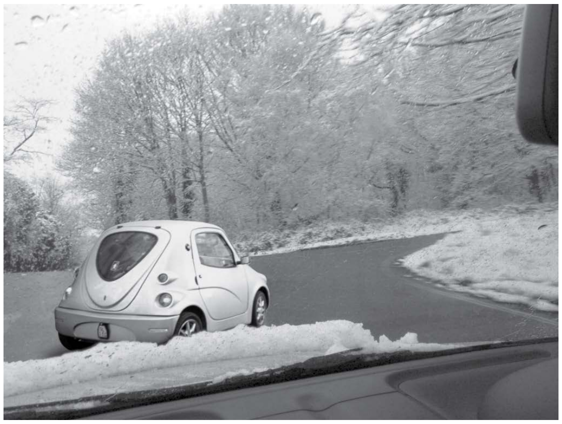


**FIGURE 4.41** Viewing the red channel

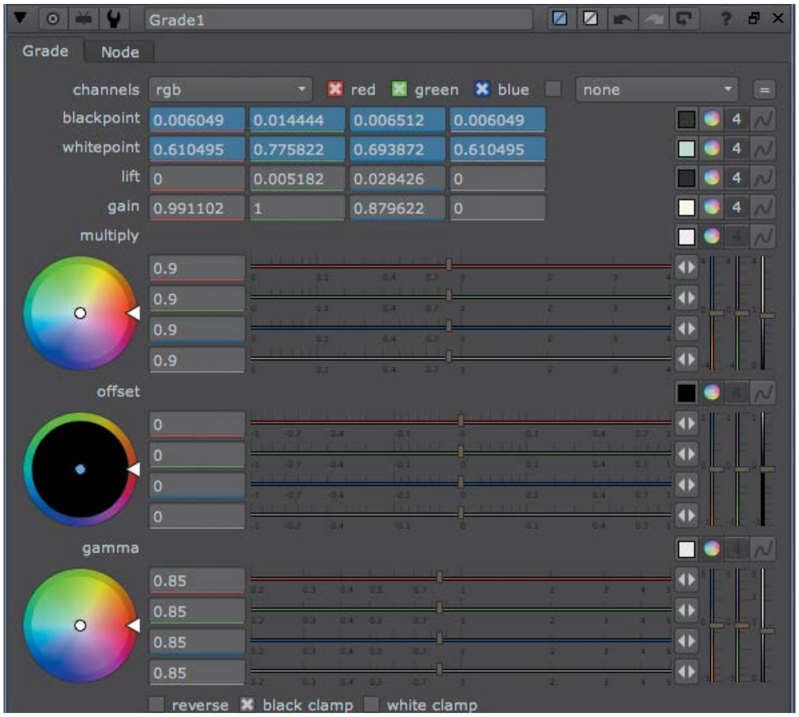
Now you are looking only at levels of gray. If you change the red sliders, you will get a better color match while still looking only at gray.

6. Display the In-panel Color Picker for the Gamma property by clicking the Color Picker button.

   You also want to change the Multiply and Offset values to achieve a perfect result. This is because, even though you matched the black point and white point, the distance of the car from the camera means the black point will be higher and the white point lower. At the end of the day, it will look right only when it does match, math aside.

   Let's display those extra color wheels.

7. Click the Color Picker button for the Multiply and Offset properties. Your screen should look like **FIGURE 4.42**.



Grade1

Grade | Node

| | | | | | |
|---|---|---|---|---|---|
| channels | rgb | ✖ red | ✖ green | ✖ blue | none |
| blackpoint | 0.006049 | 0.014444 | 0.006512 | 0.006049 | 4 |
| whitepoint | 0.610495 | 0.775822 | 0.693872 | 0.610495 | 4 |
| lift | 0 | 0.005182 | 0.028426 | 0 | 4 |
| gain | 0.991102 | 1 | 0.879622 | 0 | 4 |
| multiply | 0.9 | 0.9 | 0.9 | 0.9 | |
| offset | 0 | 0 | 0 | 0 | |
| gamma | 0.85 | 0.85 | 0.85 | 0.85 | |

reverse ✖ black clamp white clamp

**FIGURE 4.42** Opening three color wheels to easily control three properties

8. Since you are looking at the red channel in the Viewer, change the red sliders for Gamma, Multiply, and Offset until you are happy with the result; little changes go a long way. I left mine at Gamma: 0.8, Multiply: 0.82, and Offset: 0.02.

9. Display the green channel in the Viewer, and then move the green sliders to change the level of green in your image. My settings are Gamma: 0.85, Multiply: 0.89, and Offset: 0.025.

10. Do the same for the blue channel. My settings are Gamma: 1.05, Multiply: 1, and Offset: 0.065.

11. Switch back to viewing the RGB channels (**FIGURE 4.43**).

This is as far as I will take this comp. Of course, you can use your already somewhat-developed skills to make this a better comp, but I'll leave that to you.

Save your script if you wish, and we will move on.

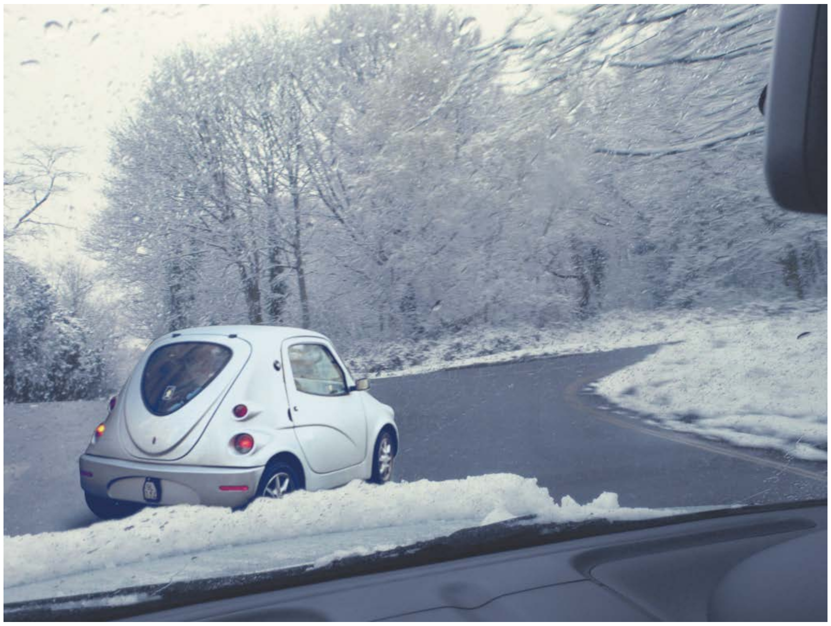


**FIGURE 4.43** Not a bad result at the end of it all.

## ACHIEVING A "LOOK" WITH THE COLORCORRECT NODE

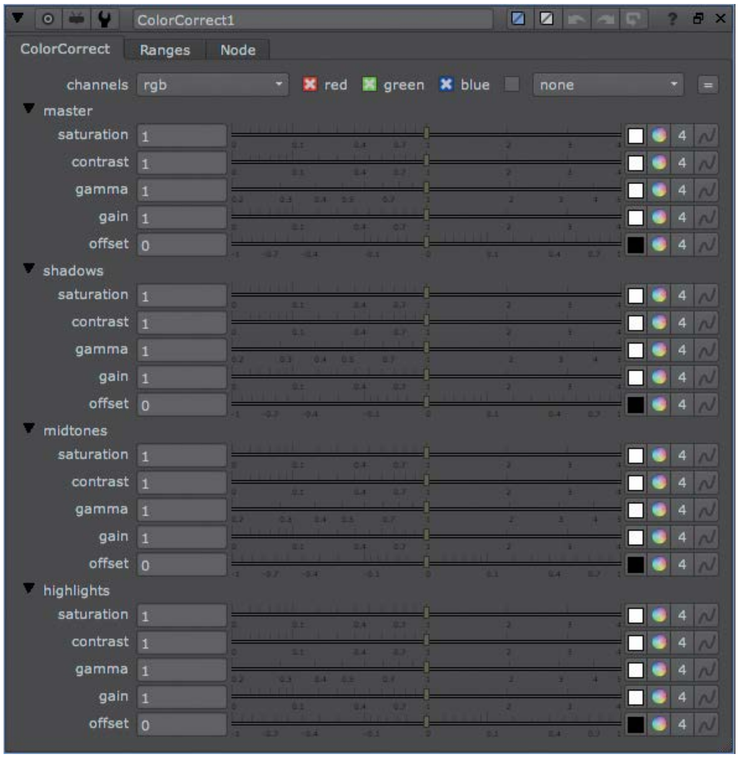
Giving an image a "look" is a very different practice than matching color. While matching color has a very specific purpose and methodology, giving an image a look refers to an artistic practice that gives an image a different feel to how it was shot. For example, you might want it to look brighter, warmer, or colder, depending on the feeling you want to create.

### Using the ColorCorrect node

The ColorCorrect node is a very good tool to use for this, as it has a lot of control over the different parts of the image—even more control than the Grade node. But as with everything else, it is still made out of the basic mathematical building blocks covered in the beginning of this chapter.

Let’s bring in an image and give it a look.

1. Press Ctrl/Cmd-W to close the color matching script and start a new one.
2. Press the R key and bring in, from the chapter04 folder, the car.png image again.
3. While the newly imported Read1 node is selected, press the C key to create a ColorCorrect node. You can also find the ColorCorrect node in the Color toolbox.
4. View ColorCorrect1 in the Viewer (**FIGURE 4.44**).



channels rgb ✓ red ✓ green ✓ blue none

master
saturation 1
contrast 1
gamma 1
gain 1
offset 0

shadows
saturation 1
contrast 1
gamma 1
gain 1
offset 0

midtones
saturation 1
contrast 1
gamma 1
gain 1
offset 0

highlights
saturation 1
contrast 1
gamma 1
gain 1
offset 0

**FIGURE 4.44** The ColorCorrect node’s Properties panel

As you can see in ColorCorrect1’s Properties panel, the ColorCorrect node includes controls for Saturation, Contrast, Gamma, Gain (Multiply), and Offset (Add). These properties are available over either the whole dynamic range—called Master—or parts of the dynamic range called shadows, midtones, and highlights. Having individual control over the different areas of the dynamic range makes creating a look somewhat easier.

This idea of midtones, highlights, and shadows changes from image to image. An image of a dark room will have no whites, but in that darkness, you can still pick out the brighter areas that are the image’s highlights, the slightly lighter blacks that are the midtones, and the darkest colors that are the shadows. You can also define these in the ColorCorrect node’s Ranges tab.

5. Click the Ranges tab in ColorCorrect1's Properties panel.

   In this tab (it's similar to ColorLookup, isn't it?) you have three graphs, all selected. One represents the shadows, another the midtones, and a third the highlights (**FIGURE 4.45**).

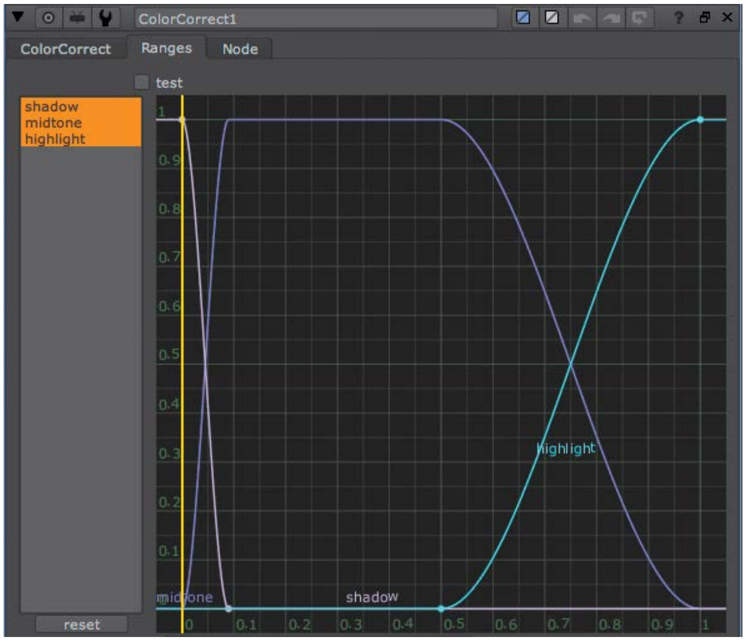


**FIGURE 4.45** ColorCorrect's Ranges is a lookup curve that defines the brightness ranges.

6. Click the Test check box at the top of the graph (**FIGURE 4.46**).

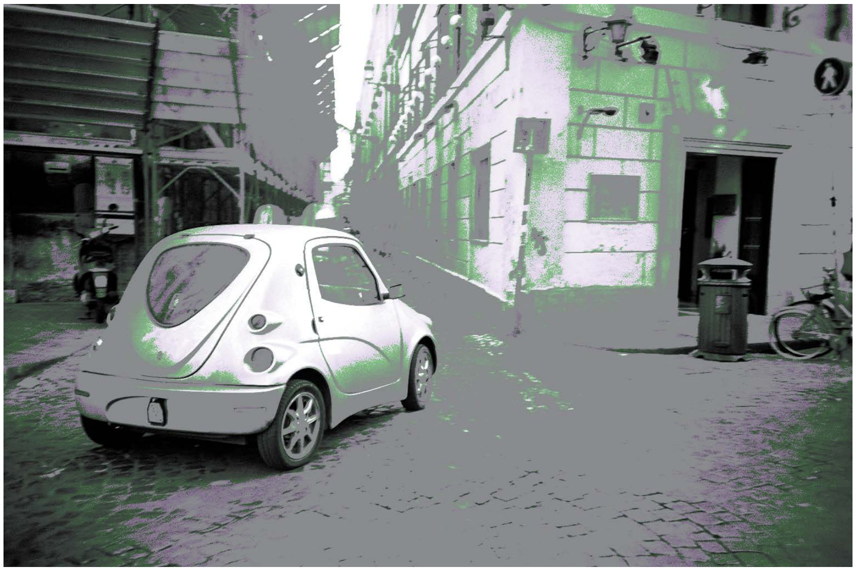


**FIGURE 4.46** The test shows the parts of the dynamic range in the Viewer.

This shows a representation in the Viewer of what parts of the image are shadow, midtone, and highlight. Highlights are represented by white, midtones as gray, and shadows as black. Green and magenta represent areas that are a mix of two ranges.

7. Click the Test button at the top of the graph again to turn it off.

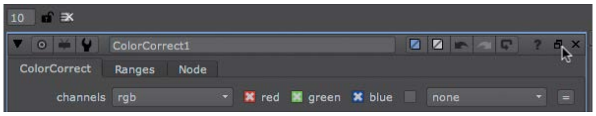
   The ranges are fine for this image, so we won't change anything and will continue working.

8. Switch back to the ColorCorrect tab.

   You will now give this image a dreamy, car commercial look—all soft pseudo blues and bright highlights. If you don't define the look you are after in the beginning, you can lose yourself very quickly.

   Before changing the color of this image, I'll show you my preferred interface setup for color correcting.

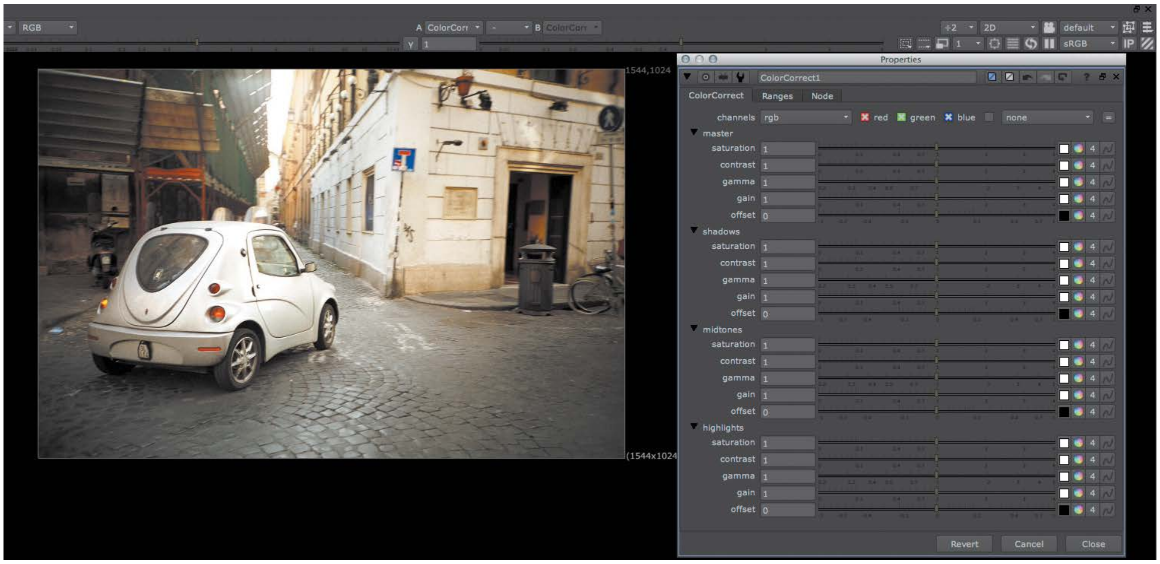
9. In ColorCorrect1's Property panel, click the Float Controls button. This will float the Properties panel instead of docking it in the Properties Bin (**FIGURE 4.47**).



channels rgb ✖ red ✖ green ✖ blue none

**FIGURE 4.47 Click the Float Controls button to float the Properties panel.**

10. Hover your mouse pointer in the Viewer and press the spacebar to maximize the Viewer to the size of the whole interface (**FIGURE 4.48**).



channels rgb ✖ red ✖ green ✖ blue none

master: saturation 1, contrast 1, gamma 1, gain 1, offset 0
shadows: saturation 1, contrast 1, gamma 1, gain 1, offset 0
midtones: saturation 1, contrast 1, gamma 1, gain 1, offset 0
highlights: saturation 1, contrast 1, gamma 1, gain 1, offset 0

**FIGURE 4.48 This is a good way to set the interface for color correction.**

Since the Properties panel is floating, it is still there. This way, you can look at the image at its maximum size without wasting space on things like the DAG, yet you are still able to manipulate the ColorCorrect node.

What I am aiming for is something like that in **FIGURE 4.49**. You can try to reach this look yourself, or you can follow my steps.

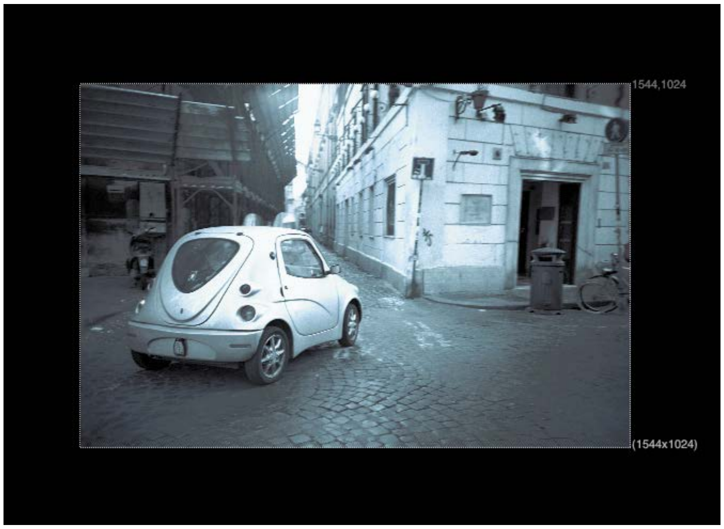


**FIGURE 4.49** This is the image look I am referring to.

11. Let's start by desaturating the whole image a little, so in the Master set of properties, set the Saturation property to 0.5.

    Now for the shadows. I would like to color the shadows a little bluer than normal.

    Remember, in addition to the In-panel Color Picker, you can also use the Floating Color Picker. To use this, Ctrl/Cmd-click the the Color Picker button. The benefit of using the Floating Color Picker is that all sliders also have Input fields, so you can type things up numerically.

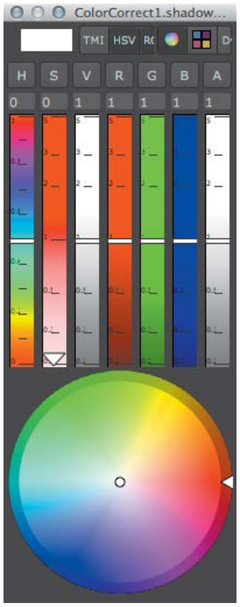
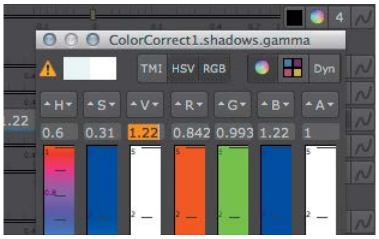


**FIGURE 4.50** The Floating Color Picker

12. Ctrl/Cmd-click the Color Picker button for shadows.gamma (**FIGURE 4.50**).
13. From the Hue slider, choose a blue hue. I selected 0.6. Now change the Saturation for the shadows.gamma color. I set it to 0.31. Finally, adjust the brightness, or Value slider in the Floating Color Picker. I have it at 1.22 (**FIGURE 4.51**).



| H | S | V | R | G | B | A |
|---|---|---|---|---|---|---|
| 0.6 | 0.31 | 1.22 | 0.842 | 0.993 | 1.22 | 1 |

**FIGURE 4.51** Setting the shadow's Gamma properties using the Floating Color Picker

This results in RGB values of 0.8418, 0.993, and 1.22, respectively. It gives the image a nice-looking blue shadow tint. Notice that there are actually no hue and saturation sliders in the real Properties. The hue and saturation sliders in the Floating Color Picker are there only so it will be easier to set the RGB sliders.

14. Close this Floating Color Picker.

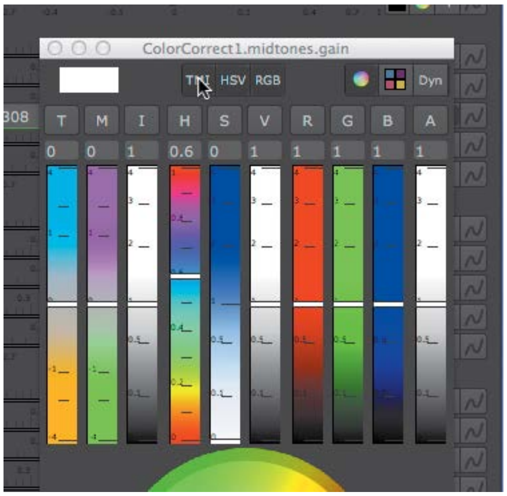
15. You have a lot more work in the midtones. First, set the Saturation to 0 so that the midtones are tinted black and white.

16. To create a flatter palette to work on, set the Contrast for midtones at 0.9.

17. To darken the midtones, set the Gamma to 0.69.

18. Use the Gain property to tint the midtones by Ctrl/Cmd-clicking the Color Picker button for Midtones/Gain.

19. In the Floating Color Picker that opens, click the TMI button at the top to enable the TMI sliders (**FIGURE 4.52**).



| T | M | I | H | S | V | R | G | B | A |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0.6 | 0 | 1 | 1 | 1 | 1 | 1 |

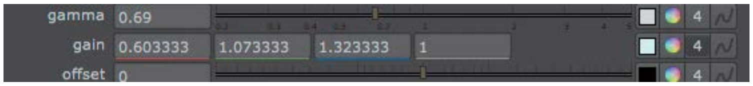
**FIGURE 4.52** Turning on the TMI sliders

If you need to make the Floating Color Picker bigger, drag the bottom-right corner of the panel.

20. Now, for a cooler looking shot, drag the T (temperature) slider up toward the blues. I stopped at 0.72.

21. To correct the hue of the blue, use the M (magenta) slider to make this blue either have more magenta or more green. I went toward the green and left it at –0.11.

22. Close the Floating Color Picker (**FIGURE 4.53**).



| | | | | |
|---|---|---|---|---|
| gamma | 0.69 | | | |
| gain | 0.603333 | 1.073333 | 1.323333 | 1 |
| offset | 0 | | | |

**FIGURE 4.53** The values are always in RGB.

As always, only the RGB values affect the image. You just used TMI sliders to set the RGB values in an easier way.

23. You now increase the highlights a little, so let's start by setting the Contrast to 1.5.

24. To color correct the highlights, first click the 4 icon to enable the individual Gain input fields.

25. Click in the right side of Gain's first input field (for the red channel) and use the up and down arrow keys on your keyboard to change the red value. I left it on 0.75 (**FIGURE 4.54**).

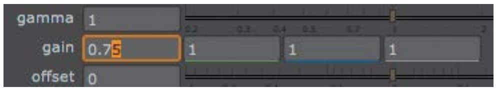


**FIGURE 4.54** **The arrow keys make it is easy to nudge values in input fields.**

26. Leave the next field (green) where it is, but use the arrow keys in the blue field to increase blue. Because I want everything to be a little bluer, I left mine at 1.5.

    The first stage of the color correction is finished. Let's bring back the rest of the interface.

27. Close the ColorCorrect1 Property panel (**FIGURE 4.55**).

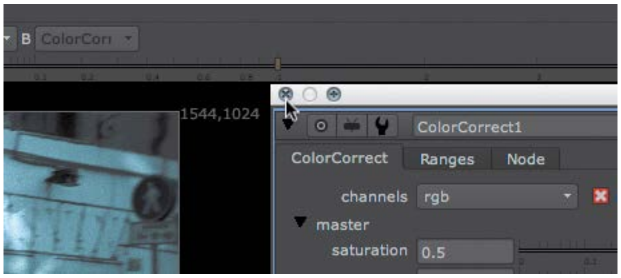


**FIGURE 4.55** **This is how to close a floating Properties panel.**

28. Press the spacebar to bring back all your panes.

## Using the mask input to color correct a portion of the image

Let's say that a commercial's director asks for the wheels to pop out of the image and have high contrast. To do this secondary color correction, first you need to define an area to apply the color correction to, then you need to use another color node and use this area in its mask input.

You haven't learned to create complex mattes yet, but in this case, you really need only two radial mattes. You can create those easily using the Radial node in the Draw toolbox.

First, to brighten up the wheels, use the Grade node.

1. Select ColorCorrect1 and insert a Grade node after it.

   If you use the Grade node as it is, the whole image gets brighter. You'll need to use Grade1's mask input to define the area in which to work.

2. With nothing selected, create a Radial node from the Draw toolbox (**FIGURE 4.56**).

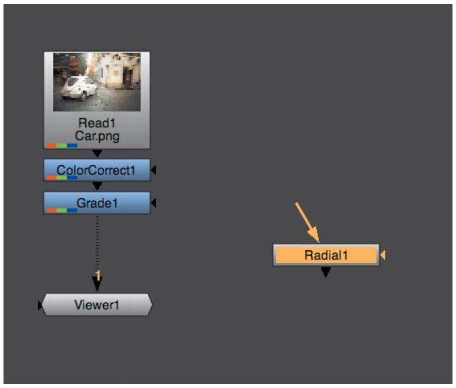


**FIGURE 4.56** Creating an unattached Radial node

3. View Radial1.

   It creates a radial, see? I told you. By moving the edges of the radial box, you can change its shape and location.

4. View Grade1.

5. Drag Radial1's edges until it encompasses the back wheel (**FIGURE 4.57**).

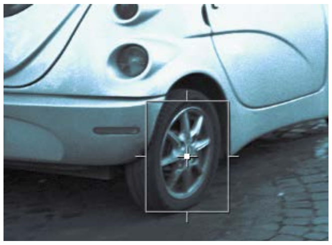


**FIGURE 4.57** Radial1 encompasses the back wheel.

   You'll need another Radial node to define the second wheel. (You can add as many Radial nodes as you need. Everything in Nuke is a node, remember?)

6. With Radial1 selected, insert another Radial node after it.

7. Adjust Radial2 to engulf the front wheel (**FIGURE 4.58**).

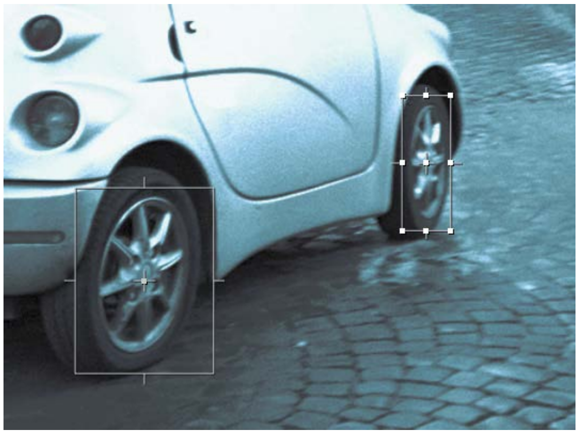


**FIGURE 4.58** Using Radial nodes to create masks for color correction

8. To make use of the radials, take the mask input for Grade1 and attach it to the output of Radial2, as in **FIGURE 4.59**.

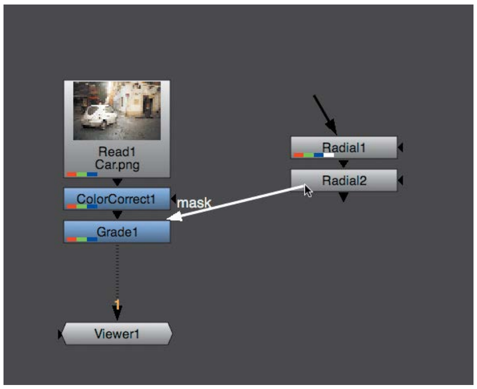


**FIGURE 4.59** Attaching the mask input to the mask image

This means whatever you now do in Grade1 affects only where the radial's branch is white.

9. Increase the whites by bringing the Whitepoint property for Grade1 down to around 0.51.

10. Some of the deep blacks have become a little too gray, so decrease the Blackpoint property a bit. I left mine at 0.022.

At this point, the grading is finished. Mask inputs can be very important in color correction because a lot of times you want to color correct only an area of the image. But remember not to confuse mask inputs with mattes or alpha channels. The use of the mask input is solely to limit an effect—not to composite one image over another or to copy an alpha channel across.

# INDEX