
FURNACECORE
BY THE FOUNDRY

FurnaceCore User Guide

Images illustrating video interlacing courtesy of Ron Brinkmann (http://www.digitalcompositing.com).

Software engineering: Ben Kent, Andy Whitmore, Ralph McEntagart, and Robert Fanner.

Product testing: Jack Binks and Dan Allum.

Writing and layout design: Eija Narvanen using Adobe FrameMaker.

Proof reading: Eija Narvanen.

Contents

INTRODUCTION

Welcome to this User Guide for FurnaceCore on Final Cut Pro.

FurnaceCore is a rich collection of image processing tools to help compositors tackle common problems when working on films. It is the product of many years development. It has been refined by drawing on feedback from the industry in general, and also from our close working relationship with London post-production houses.

About this User Guide

This User Guide will tell you how to install, license and use FurnaceCore plug-ins. Each plug-in is described in detail in later chapters.

This guide assumes you are familiar with Final Cut Pro and the machine it is running on.

What's New?

Have a look at the new features and improvements in Appendix A.

Example Images

Example images are provided for use with most of the plug-ins. You can download these images from our web site http://www.thefoundry.co.uk and try FurnaceCore out on them. (From the FurnaceCore product page, click on the Tutorials & Example Images link.)

Notation

In this User Guide we will refer to machines running FurnaceCore and Final Cut Pro as *clients* and machines that are running the Foundry FLEXlm Tools software as *servers.*

Installing FurnaceCore

FurnaceCore is available as a download from our web site http://www.thefoundry.co.uk. The downloads are in dmg format. FurnaceCore should be installed on the *client* machines.

To install FurnaceCore on a Mac OS X machine running Final Cut Pro, follow these instructions:

1. Download the file from our web site (www.thefoundry.co.uk).
2. Double click on the downloaded dmg file.
   FurnaceCore_4.1v1_FCP-mac-universal-release-32.dmg
3. Double click on the pkg file that is created.
   FurnaceCore_4.1v1_FCP-mac-universal-release-32.pkg
4. Follow the on-screen instructions to install FurnaceCore directly into the /Library/Plug-ins/FxPlug directory.
5. Proceed to "Activating FurnaceCore" on page 7.

Uninstalling FurnaceCore

The following directories are copied during the default installation process to the following paths. Removing these directories will uninstall FurnaceCore.

/Library/Plug-ins/FxPlug/FurnaceCore_4.1_FCP
/Applications/TheFoundry/FurnaceCore_4.1_FCP

Activating FurnaceCore

Once you have installed FurnaceCore, you will need to activate the product. Successful activation gives you a license key that unlocks the software.

If you have a serial number for FurnaceCore, you can activate the plug-ins via the Internet or by telephone.

If you don't have a serial number for FurnaceCore, you can activate the plug-ins by telephone.

Activation via Internet

1. Start Final Cut Pro.
2. Apply a FurnaceCore plug-in to an image sequence.
3. In the plug-in controls, press the **Help** button and make a note of your System ID (sometimes called host ID or lmhostid). This number is unique to your machine and your license key will be locked to it.
4. Press the **Enter Serial Number** button in the bottom left corner of the dialog.
5. This launches a web browser. Check that the System ID shown on the web page is correct. If it is not, or you wish to activate for a different computer, follow the on-screen instructions.
6. Type your serial number into the box provided. Make sure you type it in exactly as shown. Click **continue**.
7. If successful, you will be taken to a page that lets you download your license file and the Foundry License Installer (FLI). Click **download license**.
8. Double-click on the downloaded file to extract the license key and the Foundry License Installer (your computer may do this automatically). The license will be in a plain text file called foundry.lic. The license will look a bit like this:

```
INCREMENT furnace_fxplug_i foundry 1.0 permanent
uncounted \
HOSTID=0022411f0759 ISSUED=7-jul-2009 SIGN="0118 0259
3106 \
```

```
D626 F32A 54BC EA70 EFC6 AC23 0575 BD01 67F6 0D9B 9176
36A7 \
128A C706 C495 C017 34B8 8125"
```

Once you have downloaded the license file, proceed to "Installing a License" on page 8.

Activation by Phone

Call our London office on 020 7434 0449 (country code 44) or phone our Los Angeles office on 310 399 4555. You will need your System ID ("System ID (lmhostid)" on page 9), an e-mail address for us to send the license key, and, if you have one, a serial number.

Installing a License

Once you have received your license file, you need to install the license. The Foundry License Installer (FLI) application helps you with this. You may have received this application in an e-mail or downloaded it from http://www.thefoundry.co.uk/licensing.

To install a license:

1. Open the directory where you have saved the license file and the Foundry License Installer.
2. Double-click on the Foundry License Installer application.
3. In the window that opens, click **Install**.

   This checks the license file and installs it into the correct directory.

You're good to go. Start Final Cut Pro and check whether your plug-ins are licensed. If they are not, check that you have a foundry.lic license file in the correct directory. See "Where Does the License File Go?" on page 9.

System ID (lmhostid)

The System ID (sometimes called host ID or lmhostid) is a unique number that identifies your computer. We use this number to generate a license key for that, and only that, computer.

The System ID is shown at the bottom of the help dialog. Apply any one of the FurnaceCore plug-ins and click on the **Help** button in its controls to display this number.

Where Does the License File Go?

The license file should be called foundry.lic and saved in the following directory as a plain text file:
/Library/Application Support/TheFoundry/FLEXlm/foundry.lic

There are also other places where you can place the license file. See "Alternative License Directories" on page 9.

Alternative License Directories

If you like, you can also put the license file in an arbitrary directory and point to it with the environment variable: FOUNDRY_LICENSE_FILE

This can be useful for large post houses that have centrally managed license servers, but will not be necessary for most customers.

See "Further Reading" on page 10.

Watermark

If you don't have a valid license key, a warning will be displayed and the finished render will have coloured dots scattered over it.

License Problems

If you can't get your licenses to work, you can download the Foundry License Diagnostics (FLD) utility from http://www.thefoundry.co.uk/licensing. Run the FLD and e-mail the resulting text file to support@thefoundry.co.uk with a clear description of the problem.

Error Log Files

If the plug-ins fail to get a license, the incident is recorded in an error log file. The time, date and nature of the problem are appended to the end of the file. The error log file can then be found in the following location: /Library/Application Support/TheFoundry/FLEXlm/log/license.log

Further Reading

System Administrators may wish to find out more about licensing FurnaceCore with FLEXlm. We recommend reading the Foundry FLEXlm Tools (FFT) User Guide available to download from our web site (http://www.thefoundry.co.uk/licensing). In addition, there are general FLEXlm licensing guides on Macrovision's web site (http://www.macrovision.com).

About FurnaceCore Plug-ins

All FurnaceCore plug-ins integrate seamlessly into Final Cut Pro. They are applied to your clips as you would any other plug-in and they can all be animated using the standard animation tools.

You can load FurnaceCore plug-ins from **Effects** > **Video Filters** > **FurnaceCore**. Their parameters appear on the **Filters** tab of the Viewer.

All FurnaceCore parameters include two important buttons:

- the **Help** button. When clicked, this displays instructions related to the plug-in.
- The user message button. By default, this is grayed out. The button only activates when FurnaceCore has a message for the user. When this happens, you should click the button to see the message.

Computational Requirements

FurnaceCore brings high-end film industry plug-ins to Final Cut Pro. Generally, they solve more complex problems, and so require significantly more processing power and memory than run-of-the-mill Final Cut Pro plug-ins. Broadly speaking, simple plug-ins often only need to take a single pixel, or perhaps a few of its neighbours into account. FurnaceCore plug-ins, on the other hand, solve problems which require information to be gathered from neighbouring pixels in the current frame, as well as pixels from neighbouring frames - many more pixels need to be looked at in order to compute a single output pixel.

FurnaceCore plug-ins were originally developed for use in high-end compositing applications, such as Nuke and Shake. These products all have a ways of caching intermediate rendering results, which gives tremendous time savings. Final Cut Pro, on the other hand, does not yet have sophisticated caching. So, whenever it computes an output frame, it does quite a bit from scratch, without exploiting previously rendered items, which, if you're not wary, can lead to very long rendering times.

This brings us to the following important rule of thumb: Avoid applying several FurnaceCore plug-ins to the same clip at once. Rather, try to apply and render them out one at a

time.

To further illustrate the problem, consider the rendering time of applying F_DeNoise, followed by F_DeFlicker, followed by F_MotionBlur. For simplicity, we'll assume Final Cut Pro does no caching at all. Also, we'll assume that the clip is 100 frames long, that each of these effects evaluates 10 input frames at a time to produce one output frame, and that in isolation, each effect requires 1 minute to render an output frame. We have two options – the first being to apply all effects at once, and the second is to render them out, and apply them one at a time.

1. Rendering all effects at once: In this instance, we have 100 motion blurs, each needing 10 previous deflicker renders, each of which needs 10 previous denoise renders. This requires 100x10x10x1 minutes, for a total of 10,000 minutes.
2. Rendering one at a time: Here, we have 100 denoise renders, followed by 100 deflickers, followed by 100 motion blurs. The total time required for the three renders is 300 minutes – this is more than 33 times faster than option 1!

Field Dominance Settings

When using the FurnaceCore plug-ins, it is important that Final Cut Pro's field dominance setting for the input clip is correct. If you inadvertently mislabel the field dominance of input clips, you may get artifacts in the output sequence. generally, these artifacts will be visible on the leading and trailing edges of moving objects.

For example, if you apply F_DeFlicker to a fielded clip – for which you have inadvertently set the **Field Dominance** to **None** in Final Cut Pro – then you are likely to see artifacts in the output footage. This is because complex plug-ins access multiple input pixels, often spread over several frames, to

compute the result for one output pixel. When you mislabel field dominance, you cause information from the separate fields to be mixed together, which ultimately results in artifacts.

In short, you should be careful about inadvertently mislabelling the field dominance of input clips. To check that you are using the correct settings, right-click on your clip and select **Item Properties** > **Format** > **Field Dominance**.

A note on shifting field order

Sometimes, you will need to alter the field dominance of a lower-first input clip for use in an upper-first output sequence, or vice versa. FurnaceCore plug-ins will not alter the field-dominance for you automatically. To alter the field dominance, you should use Final Cut Pro's built-in **Shift Fields** video filter.

If you apply a FurnaceCore plug-in to a clip with the opposite field dominance of the output sequence, the plug-in will issue a user alert to remind you that you may need to use the **Shift Fields** video filter. This warning can be viewed by clicking the user message button (see "About FurnaceCore Plug-ins" on page 10).

A note on Final Cut Pro's viewer

Final Cut Pro does not always notify plug-ins when clip settings have changed. Therefore, if you change settings, for example the field dominance, you may need to alter the of the Canvas zoom level, or toggle one of the applied effects on-and-off to see the updated result. Note that this is not a FurnaceCore bug - it affects built-in Final Cut Pro video filters as well.

Using Progressive Footage with F_Kronos and F_DeNoise

At the time of writing, Final Cut Pro 6 and 7 are unable to provide full-resolution progressive frames to plug-ins that require random access to the timeline. In FurnaceCore, there are two such plug-ins: F_Kronos and F_DeNoise. If you use progressive footage with these plug-ins, the results will usually appear slightly smoothed.

Final Cut Pro does, however, allow full-resolution access to interlaced frames. Therefore, the workaround for the problems is to make Final Cut Pro believe the footage is interlaced (fielded) rather than progressive. To label your progressive input clip as being upper fielded, right-click on the clip, and select **Item Properties** > **Format**. Then, set **Field Dominance** to **Upper (Odd)**. This ensures the effects are applied to the full-resolution progressive footage. However, because Final Cut Pro now assumes the footage to be interlaced, it automatically tries to deinterlace it. To stop this from happening, set **Deinterlacing Method** to **None** in the F_Kronos or F_DeNoise controls.

Note that using this workaround produces a very subtle change in the output F_DeNoise. When used with F_Kronos, the effect is far more noticeable.

Multiple Scenes in a Clip

FurnaceCore plug-ins will only work on clips containing frames from a single scene or shot. If your footage contains scene changes, then you need to take care to apply the FurnaceCore plug-in to the relevant time domain in the clip. In Final Cut Pro, this can be done in the following way:

1. Add the FurnaceCore effect to the clip
2. Set the region of the timeline to which the effect will be applied.
    - Go the timeline directly right of the effect name.

- Drag the left and right black bars, which indicate the active boundaries to which the effect will be applied, to the positions where they need to be (that is, so the effect is only applied to the separate scene).

3. Add the same or additional FurnaceCore effects to the clip, and repeat step 2 for the remaining shots/scenes.

Customer Support

Should questions arise that this user guide fails to address, you can contact Customer Support directly via e-mail at support@thefoundry.co.uk or via telephone to our London office on +44 (0)20 7434 0449 or to our Los Angeles office on (310) 399 4555 during office hours.

Other Foundry Products

The Foundry is a leading developer of visual effects software for film and video post production. Its products include Nuke, a high-end compositing application, and plug-ins, such as Furnace, Tinder, Tinderbox, Keylight, Ocula, and RollingShutter. The plug-ins run on a variety of compositing platforms, including After Effects, Autodesk® Media and Entertainment Systems, Avid DS, Nuke, Shake, and Final Cut Pro. For the full list of products and supported platforms, see our web site http://www.thefoundry.co.uk.

Nuke is an Academy Award® winning compositor. It has been used to create extraordinary images on scores of feature films including *The Dark Knight*, *The Golden Compass*, *Iron Man*, *Transformers*, *King Kong*, and *Pirates of the Caribbean: At World's End.*

Tinder and Tinderbox are collections of image processing

effects, including blurs, distortion effects, background generators, colour tools, wipes, matte tools, paint effects, lens flares and much more.

Keylight is an industry-proven blue/green screen keyer, giving results that look photographed, not composited. The Keylight algorithm was developed by the Computer Film Company, who were honoured with a technical achievement award for digital compositing from the Academy of Motion Picture Arts and Sciences.

Ocula is a collection of tools that solve common problems with stereoscopic imagery, improve productivity in post production, and ultimately help to deliver a more rewarding 3D-stereo viewing experience.

RollingShutter is a plug-in that tackles image-distortion problems often experienced by users of CMOS cameras. The plug-in will often vastly improve the look of distorted footage, by either minimising or eradicating image distortions. Unlike solutions tied to camera stabilisation, that stretch the image as a whole, the RollingShutter plug-in compensates for local skewing and distortion in the scene, by correcting each object individually.

For further details, visit The Foundry's web site at http://www.thefoundry.co.uk.

DEFLICKER

When working in film you sometimes have to deal with shots that have a luminance flicker.

Introduction

F_DeFlicker is used to remove flicker that is localised and dependent on the geometry of the scene, such as that caused by an unsynchronised fluorescent light in a shot. It works by calculating the gain between the current frame and each frame in a small window surrounding it. It then tries to adjust the gain so that it varies smoothly over this temporal window. This means it is better at reducing fast flicker than flicker which varies slowly over the image sequence, as the latter will already appear smooth over the window and F_DeFlicker will leave it largely untouched.

The algorithm used by F_DeFlicker can introduce blurring in areas where there is rapid motion. If this happens, using local motion estimation to align the frames before deflickering them can help. However, this process is complicated by the fact that the presence of flicker can adversely affect the results of the motion estimation. F_DeFlicker therefore adopts a two stage approach to this problem. First, the normal deflickering process is performed. The motion vectors for the sequence are calculated on the resulting deflickered frames, then applied to the original frames in order to align them. The deflicker calculation is then performed on the aligned frames to give the final result. To use this approach, turn on **Use Motion** in F_DeFlicker.

Note that F_DeFlicker can be a computationally expensive plug-in that requires input frames from outside the current

time, as such, using more than two instances of F_DeFlicker in an effect stack will dramatically increase render times. It is strongly advised therefore, that you render each instance out separately.

Quick Start

Import a sequence to deflicker and place it on the timeline. Choose **Effects** > **Video Filters** > **FurnaceCore** > **F_DeFlicker (4.1)**. Render.

Parameters

The parameters for this plug-in are described below.

Scale Down - use this to scale down the image for deflickering. This can speed up the calculation.

Clamp Flicker - use this to reduce flicker without removing it entirely; smaller values mean more will be left behind.

Use Motion - turn this on to do a second deflicker pass using motion-compensated frames. This can improve results in areas where there is fast motion, where the initial deflicker pass can introduce blurring.

Vector Detail - determines the accuracy of the motion vectors used when **Use Motion** is turned on. The maximum value of 1 will generate one vector per pixel. This will produce the most accurate vectors but will take longer to render.

Window Size - the size of the temporal window to use to remove flicker.

Example

In this example, we'll look at removing flicker from a clip using F_DeFlicker. The clip used here can be downloaded from our web site. For more information, please see "Example Images" on page 5.

Step by Step

1. Import ToDeflicker.mov into Final Cut Pro and place it on the timeline.
2. Render the sequence loaded in the previous step. Notice the flickering fluorescent light in the window, which we're going to try to remove.
3. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_DeFlicker (4.1)**.
4. Render the output.
5. Compare the original and deflickered sequences by viewing them one after the other. Notice that the flickering from the light in the window has been substantially reduced.

DENOISE

This chapter looks at removing noise or grain from an image sequence using FurnaceCore's plug-in F_DeNoise. For hints, tips, tricks, and feedback please visit http://support.thefoundry.co.uk.

Introduction

F_DeNoise is designed to remove noise or grain from a clip. Assuming there is no motion in a sequence, the best way to reduce the noise is to take an average across a number of frames (temporal averaging). The noise which is different on each frame will be reduced and the picture which is the same will be reinforced. Temporal averaging is far superior to averaging pixels from within the same frame (spatial averaging) as it doesn't soften the image. Unfortunately, if there is motion in the sequence, the averaged image will be blurred as the image appears at different locations in each frame. However, by estimating the motion in the sequence using The Foundry's advanced motion estimation technology, it is possible to compensate for any motion and so average frames temporally without introducing any significant blurring artefacts.

Quick Start

Import the footage to be noise reduced and place it on the timeline. In the Canvas, set the zoom level to 100% or higher. Choose **Effects** > **Video Filters** > **FurnaceCore** > **F_DeNoise (4.1)**. By default, the analysis method of F_DeNoise is set to **Auto-analysis**. This means F_DeNoise automatically tries to pick a suitable sample region defining

an area of noise and uses that region when calculating the noise reduction.

If you're not happy with the results of **Auto-analysis**, you can set **Analysis Method** to **Sample Region**. This way, F_DeNoise works by analysing the grain structure in the region around the on-screen sample crosshair. Click on the crosshair button under **Analysis Centre** and move the crosshair over a plain area of the image in the Canvas. To get a good result, it is important that this area is free from image detail, so no textures or edges. The output should now show the denoised frame.

If you are still not satisfied with the results, try moving the sample crosshair to a different, flat area of a frame. F_DeNoise will reanalyse the grain structure every time this crosshair is repositioned.

To remove more noise simply increase the **Tune** parameter. You can also remove different amounts of noise from the red, green and blue channels by altering the **Tune Red**, **Tune Green** and **Tune Blue** parameters.

Parameters

The parameters for this plug-in are described below.

Tune - This adjusts the overall amount of noise or grain that is removed. Increase this value to remove more noise.

The **Fine Tuning** parameters allow you to remove different amounts of noise in each of the colour channels.

- **Tune Red** - increases or decreases the amount of noise removed in the red channel.

- **Tune Green** – increases or decreases the amount of noise removed in the green channel.
- **Tune Blue** – increases or decreases the amount of noise removed in the blue channel.

Analysis Method – Choose how you want to find the flat sample area that is used to analyse the grain structure in the image.

- **Auto-analysis** – let F_DeNoise automatically pick a sample area from the image.
- **Sample Region** – manually pick a sample area from the image.

Auto-analyse button – If **Analysis Method** is set to **Auto-analysis** but you are not happy with the results of the analysis, you can change the frame to analyse on the timeline and then click this button to perform a new automatic analysis.

Analysis Centre – change the position of the centre of the analysis region. Click the button to move the crosshair into a new position on the Canvas, or enter new coordinates for the centre point in the coordinates fields.

Sample Region – shows what is included in the analysis region when you are picking the sample region manually. This area should be free of image detail.

Analysis Frame – shows the frame to analyse on. You cannot adjust this parameter.

Deinterlacing Method – this parameter is mostly needed when applying F_DeNoise to interlaced (fielded) footage in order to produce progressive footage. It allows you to

choose the algorithm that is used to generate the missing field. For more information on interlaced footage, see "Frames and Fields" on page 39.
The **None** option is also useful if you are using progressive footage with F_Denoise. For more information, see "Using Progressive Footage with F_Kronos and F_DeNoise" on page 14.

- **None** - do not deinterlace the image. This option retains both fields as they are, and places them in the progressive image.
- **Duplicate** - the missing line is simply copied from the previous line. This is very quick but of lower quality than the interpolation methods. See Figure 3.
- **Interpolate** - linear interpolation from the previous and next lines. See Figure 4.
- **Wide Interpolate** - weighted interpolation using a wider range of lines than just the previous and next. This gives very subtle differences to the **Interpolate** method. See Figure 5.
- **Slope Adaptive** - complex interpolation of the missing lines. This method gives a particularly good result on slopes. It produces the same result on vertical lines as **Interpolate**, and a slightly better result on horizontal lines. See Figure 6.

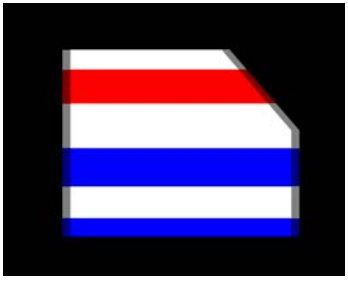

Figure 1. Source Image.

Figure 2. Close up of fields.

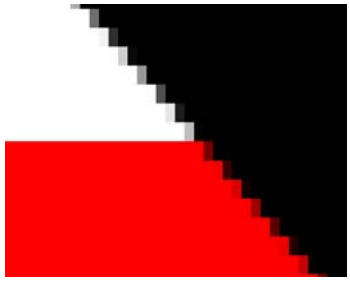

Figure 3. Duplicate.

Figure 4. Interpolate.

Figure 5. Wide Interpolate.

Figure 6. Slope Adaptive.

Example

The footage used in the following example can be downloaded from our web site. For more information, please see "Example Images" on page 5.

Mike

In this example, we'll use F_DeNoise to remove noise from a sequence.

Step by Step

1. Import MikeWire.mov and place it on the timeline.

Figure 7. Original image.

2. In the Canvas, set the Zoom pop-up menu to 100% or higher.
3. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_DeNoise (4.1)**.
4. Render out the sequence to see the results.

5. Compare the original and denoised sequences by viewing them one after the other. Notice that the noise has been reduced.

Figure 8. Zoomed original image.

Figure 9. Zoomed output image.

DIRTREMOVAL

This chapter looks at the removal of dust and dirt from images using FurnaceCore's plug-in F_DirtRemoval. For hints, tips, tricks, and feedback please visit http://support.thefoundry.co.uk.

Introduction

F_DirtRemoval will automatically detect and remove specs of dust and dirt from a frame.

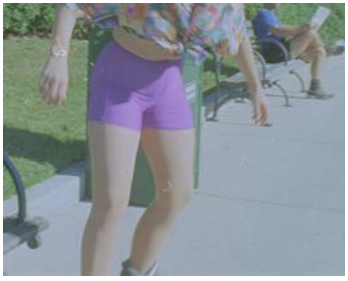

Figure 10. Before.

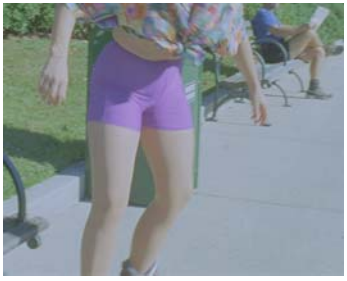

Figure 11. After.

The plug-in works by looking for objects that appear for only one frame, after taking account of the motion in the sequence. For example:

- A spec of dirt that appears for only one frame will be classified as dirt.
- A football being kicked across the image will not be classified as dirt because, after taking account of motion, it appears in each frame of the sequence.
- A vertical scratch in a sequence will not be classified as dirt as it appears in the same place in each frame.

- Dirt on the camera lens or in the telecine gate will not be classified as dirt as it appears in the same place on each frame.

Having detected the location of the dirt, the algorithm produces a seamless repair by taking motion compensated pixels from the surrounding frames and interpolating them into the dirt region. In order to see which regions have been repaired, you can select **Dirt** under **Output** and render the result, which shows the pixels detected as dirt.

The main control provided by the plug-in is the set of **Presets**. These control the trade off between falsely identifying dirt and failing to spot the dirt. Often, even if a region of image has been falsely detected as dirt, it will be repaired perfectly as the dirt will not have corrupted the motion in the region, allowing a high quality motion compensated repair.

In order to understand how to tune the parameters it is first necessary to understand a bit more about the algorithms involved. F_DirtRemoval relies heavily on motion estimation to both detect the dirt and repair the image. Where the motion is complex, e.g. multiple objects moving fast in multiple directions, we are unable to correctly calculate the motion. This means both the dirt detection and repair will fail. In order to improve results in these regions we have a complex motion detector. This detector is designed to flag regions where we are unlikely to calculate the correct motion. In these regions, we detune the motion based dirt detector and add a spatial dirt detector. Only if both detectors flag dirt do we actually believe there to be dirt.

Quick Start

Import the sequence to be cleaned and place it on the timeline. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_DirtRemoval (4.1)**. Render and view the result. To see where the dirt has been detected and removed, set **Output** to **Dirt** and render again.

If you think the results could be better, vary the preset applied as required. If regions of motion have been incorrectly classified as dirt, choose a lower preset. If dirt has been missed, try choosing a higher preset.

Parameters

The parameters for this plug-in are described below.

Presets - this is the main control for the plug-in which trades off the amount of dirt detected and repaired verses the number of false detections and incorrect repairs. For archive footage, you should typically choose **Very Aggressive**, whereas for a modern scan with isolated patches of dust you should choose **Cautious** or **Neutral**.

Output - as well as the repaired image it is possible to output diagnostic images which are useful for tuning the parameters.

- **Source** - the original frame containing dirt.
- **Dirt** - a black and white image showing where dirt has been detected.
- **Repair** - the repaired output image.
- **Repair Dirt In Alpha** - the repaired output image with the final dirt shown in the alpha channel. Any alpha in the source image will be removed.

Examples

The images for the following example can be downloaded from our web site.

Roller blades

This clip of a roller blader suffers from a lot of dust.

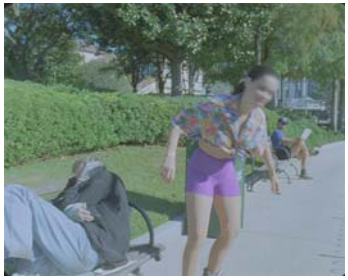

Figure 12. Roller blader.

Step-by-Step

1. Import Rollerblade.mov and place it on the timeline. Play it and look at the dirt.
2. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_DirtRemoval (4.1)**.

3. Make sure **Output** is set to **Repair** and render. Notice how the dust has been reduced in the result.

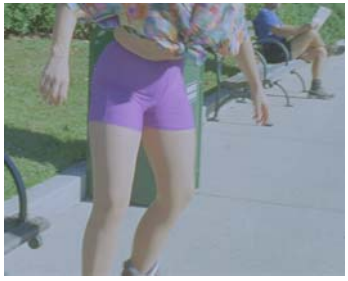

Figure 13. Repaired image.

4. To see where dirt was detected and removed, set **Output** to **Dirt** and render again.

Figure 14. The detected dirt.

KRONOS

Introduction

Kronos is FurnaceCore's re-timer and is designed to slow down or speed up footage. It works by calculating the motion in the sequence in order to generate motion vectors. These motion vectors describe how each pixel moves from frame to frame. With accurate motion vectors it is possible to generate an output image at any point in time throughout the sequence by interpolating along the direction of the motion.

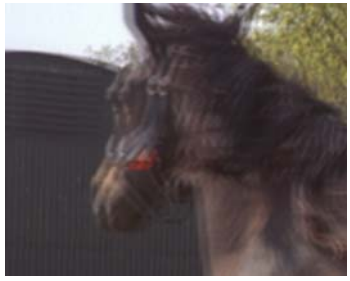

Figure 15. Simple mix of two frames to achieve an inbetween frame.

Figure 16. Kronos vector interpolation of the same two frames.

Kronos contains a number of controls to allow you to trade off render time verses accuracy of vectors. The time controls can be used to generate arbitrary shaped speed curves.

Quick Start

Import the sequence you wish to retime and place it on the timeline. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_Kronos (4.1)**. By default, the speed control will be set to perform a half speed slow down. Note that the process of

retiming inherently results in a slight softening in the retimed frames. If you were to combine retimed and original frames in a video sequence, a slight pulsing effect between the soft retimed and sharp original frames would be evident. To avoid this, Kronos never uses original frames in its output (unless you have set the Method to Frame - see the description on page 42). So, when doing a half-speed slowdown for a frame at position 1, Kronos does not simply create an additional retimed frame at position 1+0.5. Rather, creates two new frames - one at position 1-0.25 and one at position 1+0.25, and discards the original frame at position 1.

A Note on Keyframes

To fully explore the rich retiming and motion effects Kronos provides, it is useful to know something about keyframes and the animation of parameters in Final Cut Pro. In particular, the parameters for the two Timing methods - Speed and Frame - are excellent candidates for animation. To set a keyframe, click on the Ins/Del Keyframe button in the Nav column. Thereafter, you can set the parameter to a desired value for the keyframe. Specifying more keyframes will allow greater flexibility in animating the parameter over time.

Time Curves

Kronos provides two ways of specifying the retiming. The first is to set the speed of the output clip directly, and the second is to specify the time mapping between input frames and output frames. Note that both of these methods support animation, so that you can specify the desired speed/time mapping at particular points in the timeline.

By default, the **Timing** method is set to **Speed**. This allows

you to speed up or slow down an entire clip by a constant factor, or, by animating the parameter, to specify the speed-up/slow-down factor of the output clip for specified keyframes.

Setting the **Timing** method to **Source Frame** allows you specify the time mapping between the input and output clips. To do this, make sure **Frame** is keyframed. Then, select an output frame from the Timeline and set **Frame** to the input frame you want to appear at that output position. Repeat this for at least one more output position to get a linear time curve. For example, if we wish to do a 4 times slow down, move to frame 1 and set **Frame** to 1, then move to frame 19 and set **Frame** to 5. You can use all the normal time curve tools provided by Final Cut Pro to create any time curve you might need. If the motion is speeded up, motion blur will be seen.

Tuning Parameters

At this point, you can render a re-timed sequence using the default parameter settings. Better results may be achieved by tuning Kronos using the following parameters.

Kronos uses motion estimation to retime footage. These motion estimates are stored in a vector field. You can adjust the **Vector Field** parameter to get a higher resolution vector field. The larger **Vector Detail** is, the greater the processing time, but the more accurate the vectors should be. A value of 1 will generate a vector at each pixel. A value of 0.5 will generate a vector at every other pixel.

Note *In some circumstances, you may find that a lower Vector Detail may give smoother, more natural looking results, even though they are less accurate.*

Use **Smoothness** to adjust the smoothness of the vectors generated. A high smoothness will miss lots of local detail, but is less likely to provide you with the odd spurious vector. A low smoothness will concentrate on detail matching, even if the resulting field is jagged.

Using Placeholder Frames in Slow Downs

When slowing down footage, you may notice that you cannot access the entire clip after applying F_Kronos to it. For example, if your clip is 50 frames long and you slow it down to half a speed, 25 frames of the original clip are left outside the accessible frame range. To be able to access the entire original clip, you can make the clip longer by adding black placeholder frames to it and embedding them and the original clip in a sequence before using F_Kronos. Do the following:

1. In the Browser, go to the **Effects** tab.
2. Double-click on **Video Generators**.
3. Select **Slug** and drag it to the timeline either before or after your footage, depending on where you need the black placeholder frames.
4. On the timeline, resize the slug as necessary by dragging its edges to a new position. The slug and the clip together should cover the frame range you want to use for the retimed sequence.
5. Ensure that the concatenated clip and slug are flush with the left-hand side of the timeline, so that there is no gap between the start of the sequence and the first clip or slug frame.
6. In the Browser, go to the project tab.
7. Right-click on an empty spot and select **New Sequence**.
8. Right-click on the newly created sequence and select **Open Timeline**. A new tab appears on the timeline.

9. Drag the earlier sequence that contains the original footage and slug on the new timeline tab.
10. Right-click on this sequence on the timeline and select **Open in Viewer**.
11. Apply F_Kronos.

    Note that F_Kronos will be applied to the entire temporary sequence created in step 7. If you see unexpected blank frames in the rendered sequence, you may have omitted step 5.

Motion Blur without Retiming

You can add motion blur without retiming. To do this, set the **Speed** to 1. Then set **Shutter Time** (the output shutter time) to be a value greater than 1 (Figure 20 on page 45). Increase **Shutter Samples** until you don't see multiple images (say 10). This will add motion blur along the direction of motion without retiming (Figure 23 on page 45). Alternatively, you can use the simpler F_MotionBlur plug-in, described on page 57.

Frames and Fields

For various historical reasons, video frames are usually divided into two *fields*. Each field contains only half of the image information, drawn as horizontal scan lines. The first field of a frame contains every other scan line. These scan lines are displayed first from top to bottom. Immediately after, the remaining scan lines are displayed, similarly from top to bottom. They make up the second field of the frame. When both fields are displayed in rapid sequence, they appear to produce a normal, complete frame. This technique of dividing frames into fields and then displaying them

immediately after each other is called *interlacing*. It is illustrated in Figure 17.

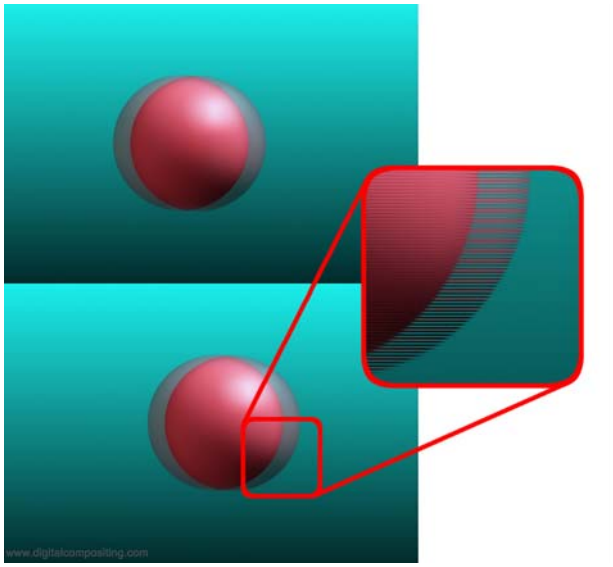

Figure 17. Video frames consist of two interlaced fields.

When a moving object is captured using an interlaced video format, the object appears in a slightly different place in each field of the same frame. This is because the second field is recorded after the first field. For example, let's imagine a clip that shows a ball moving from left to right across the screen. Figure 18 shows what two frames from this clip might look like and how these frames could be represented as two fields. As you can see, the ball moves slightly to the right between fields, and the fields appear squashed, because they only contain half of the image information. You should also notice that because the fields are recorded and displayed immediately after each other and two fields make up one frame, the ball does not move quite

as far between two fields as it does between two frames.

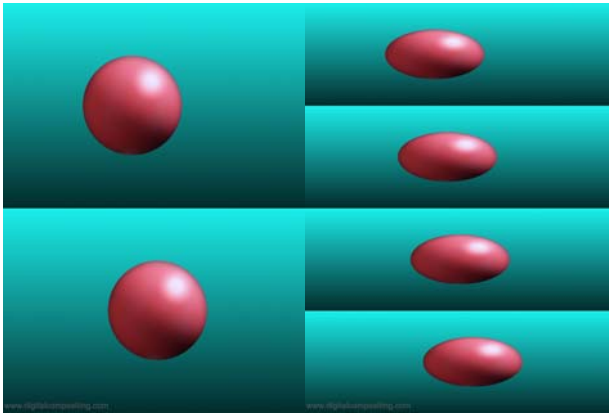

Figure 18. Two frames (on the left) divided into fields (on the right).

Kronos is able to retime between frames and between fields. This is illustrated in Figure 19.

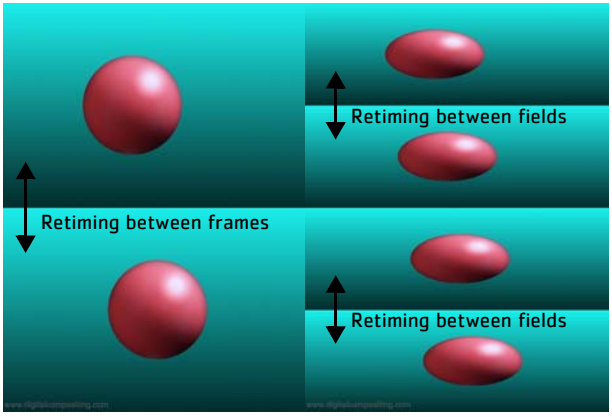

Figure 19. Retiming between frames and between fields.

By default, the retime operation is applied to frames. Because any moving objects will be further apart between frames than between fields, this can produce a slightly riskier retime. To get better results for the retime, you can perform the retime on fields instead of frames. This way, the moving objects will be closer to each other, which improves the results of the motion estimation. However, because the retime is done between fields, and the scan lines that produce the image are slightly apart vertically, the results are vertically shifted by one pixel. This can introduce vertical jitter in the retimed clip.

You can choose whether to retime between frames or fields using the Retime Fields parameter.

Parameters

The parameters for this plug-in are described below.

Method - sets the interpolation algorithm.

- **Frame** - the nearest original frame is displayed.
- **Blend** - a mix between two frames is used for the in-between frame. This is quick to render and useful when tweaking the timing on the curve before setting the method to motion.
- **Motion** - vector interpolation is used to calculate the in-between frame.

Timing - sets how to control the new timing of the clip.

- **Speed** - select this if you wish to increase or decrease the clip duration by a constant factor, that is, double speed will halve the duration of the clip or half speed will double the duration of the clip.

- **Source Frame** – select this if you wish to retime your footage by specifying a time mapping between input and output frames. This method is also useful if you wish to describe the retiming in terms of "at frame 100 in the output clip I want to see frame 50 of the source clip". You'll need to set at least 2 keyframes for this to retime the clip. Note that by specifying time mappings, you can repeat, or even reverse your footage. For example, to reverse footage, you map the final frame of the input clip to the first frame of the output clip, and then map the first frame of the input clip to the final frame of the output clip.

Speed – this parameter is only active if **Timing** is set to **Speed**. Values below 1 slow down the clip. Values above 1 speed up movement. For example, to slow down the clip by a factor of two (half speed) set this value to 0.5. Quarter speed would be 0.25.

Frame – this parameter is active only if **Timing** is set to **Source Frame**. Use this to specify the source frame at the current frame in the timeline. For example, to slow down a 25 frame clip by half, first pad it by appending a 25 frame slug (see "Using Placeholder Frames in Slow Downs" on page 38). Then, set frame 1 as a keyframe (by clicking on the Ins/Del Keyframe button in the Nav column), and set the Frame parameter to 1. Thereafter, set frame 50 (the last frame in the slug) as the second keyframe, and set the Frame parameter to 25. Thus, frame 1 will stay put, and frame 25 of the input clip will be mapped to frame 50 of the output clip to create a half-speed retime. Arbitrarily complex retimings can be made by further animation of the Frame parameter.

Vector Detail – Adjust this to vary the resolution of the

vector field. The larger vector detail is, the greater the processing time, but the more detailed the vectors should be. A value of 1.0 will generate a vector at each pixel. A value of 0.5 will generate a vector at every other pixel. For some sequences, a high vector detail near 1.0 generates too much unwanted local motion detail; a low value is often more appropriate.

Smoothness - Vector fields usually have two important qualities: they should accurately match similar pixels in one image to another and they should be smooth rather than noisy. Often, it is necessary to trade one of these qualities off against the other. A high smoothness will miss lots of local detail, but is less likely to provide you with the odd spurious vector. A low smoothness will concentrate on detail matching, even if the resulting field is jagged. The default value of 0.5 should work well for most sequences.

Warp Mode - sets how to control the new timing of the clip.

- **Simple** - this is the quickest option, but may produce less than optimal results around moving objects and image edges.
- **Normal** - this is the standard option, with more optimal treatment of moving objects and image edges.
- **Occlusions** - this is an advanced option that attempts to reduce the level of background dragging between foreground and background objects.

Shutter Time – sets the equivalent shutter time of the re-timed sequence. A shutter time of 1 is equivalent to averaging over plus and minus half an input frame which is equivalent to a shutter angle of 360 degrees. A shutter time of 0.5 is equivalent to a shutter angle of 180 degrees. Imagine a grey rectangle moving left to right horizontally

across the screen. Figure 20 and Figure 21 show how **Shutter Time** affects the retimed rectangle.

Figure 20. Shutter Time 1

Figure 21. Shutter Time 0.5

Decreasing Shutter Samples to any value less than 2 will effectively disable motion blur. The default value is 1, so that motion blur is off by default.

Shutter Samples – sets the number of in-between images used to create an output image during the shutter time. Increase this value for smoother motion blur.

Figure 22. Shutter Samples 2

Figure 23. Shutter Samples 20

Auto ShutterTime - works in conjunction with the Timing parameters (Speed or Frame). If you've animated the Speed or Frame parameters so that your footage is sped up by

varying amounts at different times, then Auto ShutterTime will attempt to apply a suitable amount of motion blur to the retimed footage. Essentially, it animates the Shutter Time parameter automatically. As always, Shutter Samples specifies the smoothness of the motion blur. Note that the effect of Auto ShutterTime will only be noticeable if (a) you have sped up your footage, and (b) you have set Shutter Samples to a value greater than or equal to 2.

Retime Fields - sets whether the retime operation is performed between frames or fields.

- **From Frame to Frame** - retimes between complete frames. This can produce a slightly riskier retime, but the results will be on the correct scan line.
- **From Field to Field** - retimes between the fields that are used to make up frames. This produces a better retime, but the results are shifted vertically by one pixel, which may introduce vertical jitter in the retimed clip.

Deinterlacing Method - this parameter is mostly needed when applying F_Kronos to interlaced (fielded) footage in order to produce progressive footage. It allows you to choose the algorithm that is used to generate the missing field. For more information on interlaced footage, see "Frames and Fields" on page 39.
The **None** option is also useful if you are using progressive input footage with F_Kronos. For more information, see "Using Progressive Footage with F_Kronos and F_DeNoise" on page 14.

- **None** - do not deinterlace the image. This option retains both fields as they are, and places them in the progressive image.

- **Duplicate** - the missing line is simply copied from the previous line. This is very quick but of lower quality than the interpolation methods. See Figure 26.
- **Interpolate** - linear interpolation from the previous and next lines. See Figure 27.
- **Wide Interpolate** - weighted interpolation using a wider range of lines than just the previous and next. This gives very subtle differences to the **Interpolate** method. See Figure 28.
- **Slope Adaptive** - complex interpolation of the missing lines. This method gives a particularly good result on slopes. It produces the same result on vertical lines as **Interpolate**, and a slightly better result on horizontal lines. See Figure 29.

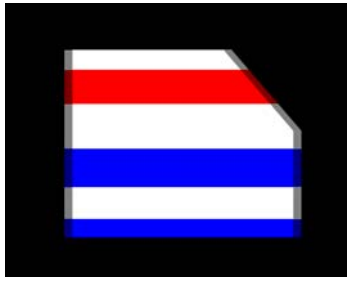

Figure 24. Source Image.

Figure 25. Close up of fields.

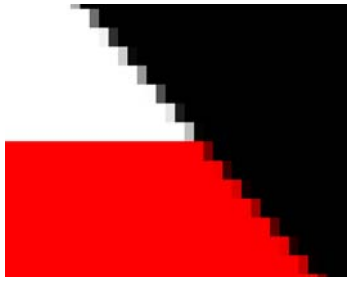

Figure 26. Duplicate.

Figure 27. Interpolate.

Figure 28. Wide Interpolate.

Figure 29. Slope Adaptive.

Examples

All the images for the following example can be downloaded from our web site.

Taxi

This example, Figure 30 on page 49, shows a taxi driving past our offices in Soho. We'll retime this sequence to speed up the taxi at the start and slow down at the end.

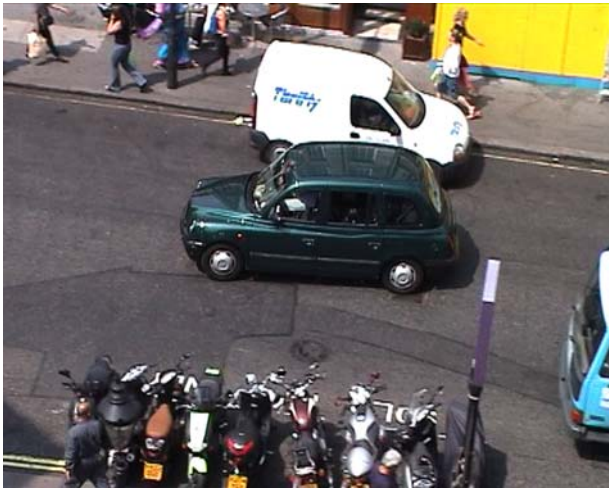

Figure 30. London taxi.

Step by Step

1. Import Taxi.mov and place it on the timeline.
2. Render the taxi clip to get a sense of the motion.
3. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_Kronos (4.1)**.
4. By default, the clip will be slowed down to half speed. However, we wish to have a fast start and slow end. So, set **Timing** to **Source Frame**. At frame 1, set a **Frame** key to be 1, and at frame 50 set a **Frame** key to 25. Now to get that fast start, try setting the **Frame** key to be 12 at frame 15. Use the keyframe graph area to modify the animation graph.
5. To get a sense of the motion you can set **Method** to **Blend**, render and adjust your timing curve from there.

6. When you're happy, set **Method** to **Motion**. With **Vector Detail** set to 0.2, you may see part of the road under the taxi being dragged by the taxi. To improve this, increase this value to 1.

MATCHGRADE

This chapter looks at automatic colour matching using FurnaceCore's plug-in F_MatchGrade. For hints, tips, tricks, and feedback please visit http://support.thefoundry.co.uk.

Introduction

It is often necessary to match the colours of one clip with those of another. When filming outside at different times of the day you will inevitably get colour and luminance differences that will have to be corrected if the sequences are to be composited or edited together.

You can, of course, use colour correction tools and trial and error to try and match the clips. But this tends to be time consuming and requires some considerable skill. F_MatchGrade does it all for you by automatically modifying the colour histogram of an image to match a target image.

The target image contains the target colours. Essentially, you're going to change the source image so that it looks more like the target image.

Figure 31. Source image

Figure 32. Target image

Figure 33. Output image

Quick Start

Import the clips you want to use as the source and target clips and place them on the timeline. The source clip is the sequence to which the colour transform will be applied. The target clip is the sequence you want the source clip to match. Select the source clip and choose **Effects** > **Video Filters** > **FurnaceCore** > **F_MatchGrade (4.1)**. Drag the target clip to the clip well of the **Target** parameter. View the output which should now match the look of the target clip. Try increasing **Iterations** if the match isn't close enough.

By default, F_MatchGrade calculates the transform needed to match the source image frame to the target image frame, and applies this transform to every frame of the source sequence. This way, the transformation is temporally uniform. However, if you want the colour transfer to update according to the current frame of the source and target clips, you can set the **Source Colour From** and **Target Colour From** parameters to **Current Frame**.

Parameters

The parameters for this plug-in are described below.

Target – drag the target clip (the clip whose colours you want the source clip to match) here.

Iterations – the number of refinement passes. More iterations should produce a better match but will take longer. This is an integer parameter, so animating it will not produce a smooth grade change but one with obvious steps. To achieve a smooth grade change, mix the output from F_MatchGrade with the original input sequence and animate the mix amount.

Source Colour From – sets how the source clip is used to calculate the colour transformation.

- **Specified Frame** – the colour transformation is calculated using a single frame from the source clip. You can specify this frame using the **Source Frame** parameter below.
- **Current Frame** – the colour transformation is calculated so that it updates according to the frames in the source clip.

Source Frame – sets the source clip frame that is used to calculate the colour transformation.

Target Colour From – sets how the target clip is used to calculate the colour transformation.

- **Specified Frame** – the colour transformation is calculated using a single frame from the target clip. You can specify this frame using the **Target Frame** parameter below.
- **Current Frame**– the colour transformation is calculated so that it updates according to the frames in the target clip.

Target Frame – sets the target clip frame that is used to calculate the colour transformation.

Example

Mike

In this example, we will use F_MatchGrade to match the look of two images, MikeWalking and MikeLightWand. The images used here can be downloaded from our website. For more information, please see "Example Images" on page 5.

Step by Step

1. Import MikeWalking.tif and MikeLightWand.tif and place them on the timeline. Our goal is to make the image with the light stick lighter and more like the other. They are shown in Figure 34 and Figure 35.

Figure 34. Source image

Figure 35. Target image

2. Select the light stick image and choose **Effects** > **Video Filters** > **FurnaceCore** > **F_MatchGrade (4.1)**.

3. Drag the walking image into the clip well of the **Target** parameter. You should get the result in Figure 36.

Figure 36. Output image

4. The result is slightly garish, so decrease the **Iterations** parameter to 3, the result of which is shown in Figure 37.

Figure 37. Output image

MOTIONBLUR

This chapter looks at adding motion blur using FurnaceCore's plug-in F_MotionBlur. For hints, tips, tricks, and feedback please visit http://support.thefoundry.co.uk.

Introduction

F_MotionBlur uses the Foundry's advanced motion estimation technology to add realistic motion blur to a sequence. F_MotionBlur uses the same techniques and technology as the motion blur found in F_Kronos, but presents the controls in a less complex, more user friendly way. However, if you need precise control over the motion vectors used for adding blur, or a large temporal range (i.e. a very high shutter time), you should use F_Kronos.

Quick Start

Import a clip and place it on the timeline. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_MotionBlur (4.1)**. Select a suitable **Shutter Time**, depending on the amount of blur you wish to add. Process the sequence to see the motion blurred result. Increasing **Shutter Samples** will result in more in-between images being used to generate the motion blur and a smoother blur. If you can see that the motion blur has been created from a few discreet images, try increasing **Shutter Samples**.

If your sequence is composed of a foreground object moving over a background, the motion estimation is likely to get confused at the edge between the two.

Frames and Fields

For various historical reasons, video frames are usually divided into two *fields*. Each field contains only half of the image information, drawn as horizontal scan lines. The first field of a frame contains every other scan line. These scan lines are displayed first from top to bottom. Immediately after, the remaining scan lines are displayed, similarly from top to bottom. They make up the second field of the frame. When both fields are displayed in rapid sequence, they appear to produce a normal, complete frame. This technique of dividing frames into fields and then displaying them immediately after each other is called *interlacing*. It is illustrated in Figure 38.

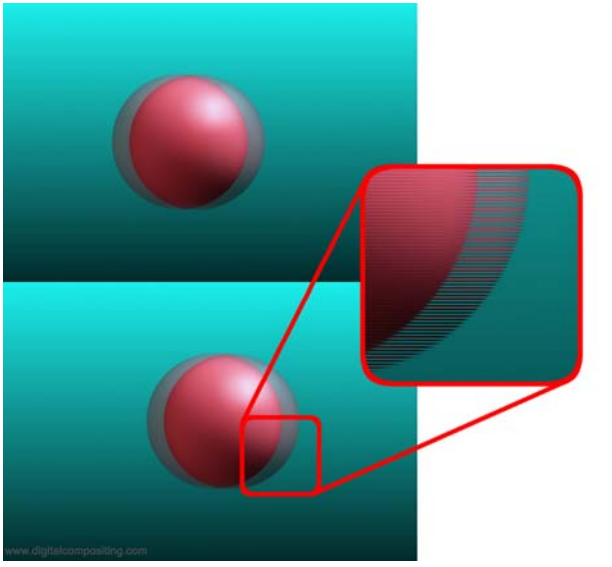

Figure 38. Video frames consist of two interlaced fields.

When a moving object is captured using an interlaced video format, the object appears in a slightly different place in each field of the same frame. This is because the second

field is recorded after the first field. For example, let's imagine a clip that shows a ball moving from left to right across the screen. Figure 39 shows what two frames from this clip might look like and how these frames could be represented as two fields. As you can see, the ball moves slightly to the right between fields, and the fields appear squashed, because they only contain half of the image information. You should also notice that because the fields are recorded and displayed immediately after each other and two fields make up one frame, the ball does not move as far between two fields as it does between two frames.

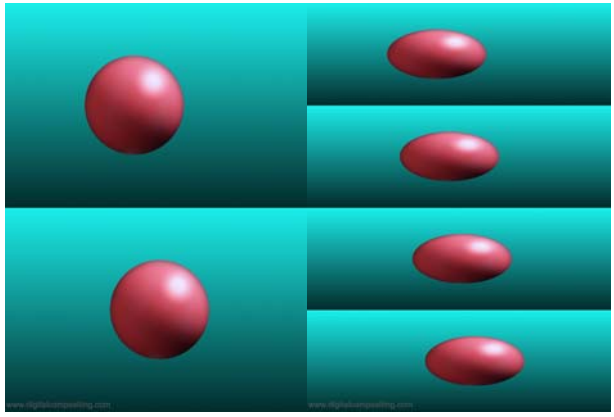

Figure 39. Two frames (on the left) divided into fields (on the right).

MotionBlur is able to retime between frames and between

fields. This is illustrated in Figure 40.

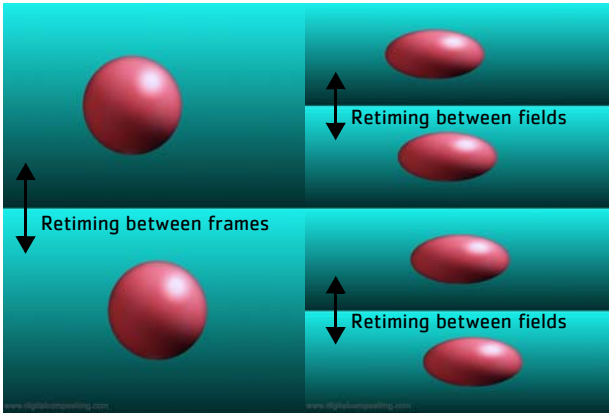

Figure 40. Retiming between frames and between fields.

By default, the retime operation is applied to frames. Because any moving objects will be further apart between frames than between fields, this can produce a slightly riskier retime. To get better results for the retime, you can perform the retime on fields instead of frames. This way, the moving objects will be closer to each other, which improves the results of the motion estimation. However, because the retime is done between fields, and the scan lines that produce the image are slightly apart vertically, the results are vertically shifted by one pixel. This can introduce vertical jitter in the retimed clip.

You can choose whether to retime between frames or fields using the Retime Fields parameter.

Parameters

The parameters for this plug-in are described below.

Shutter Time - sets the equivalent shutter time of the retimed sequence. A shutter time of 1 is equivalent to averaging over plus and minus half an input frame which is equivalent to a shutter angle of 360 degrees. A shutter time of 0.5 is equivalent to a shutter angle of 180 degrees. Imagine a grey rectangle moving left to right horizontally across the screen. Figure 41 and Figure 42 show how Shutter Time affects the retimed rectangle.

Figure 41. shutterTime 1

Figure 42. shutterTime 0.5

Note that the effect of Shutter Time will only be noticeable if you've enabled motion blur by setting Shutter Samples to a value greater than or equal to 2.

Shutter Samples - sets the number of in-between images used to create an output image during the shutter time. Increase this value for smoother motion blur.

Decreasing Shutter Samples to any value less than 2 will effectively disable motion blur. The default value is 1, so

that motion blur is off by default.

Figure 43. shutterSamples 2

Figure 44. shutterSamples 20

Vector Detail – the amount of detail used for the motion estimation. The maximum value of 1 will produce the most accurate motion vectors, but will take longer to render.

Retime Fields – sets whether the retime operation is performed between frames or fields.

- **From Frame to Frame** – retimes between complete frames. This can produce a slightly riskier retime, but the results will be on the correct scan line.
- **From Field to Field** – retimes between the fields that are used to make up frames. This produces a better retime, but the results are shifted vertically by one pixel, which may introduce vertical jitter in the retimed clip.

Example

All the images for the following example can be downloaded from our web site. For more information, please see "Example Images" on page 5.

BelleWalking

In this example, we'll use F_MotionBlur to add motion blur to the sequence.

Step by Step

1. Import the BelleWalking.mov clip and place it on the timeline.
2. Render the clip to get a sense of the motion. Go to frame 16 on the time line (this merely gives a better example than the initial frames).
3. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_MotionBlur (4.1)**.
4. Set **Shutter Time** to 10, and **Shutter Samples** to 10. You should get an image as in Figure 45.

Figure 45. Output image

5. We can see the individual samples still, for example at the back of the dress and the right-hand foot. Increase **Shutter Samples**

to 20 to sample more frames, and you should get a result as in Figure 46.

Figure 46. Output image

ReGrain

Introduction

FurnaceCore's ReGrain plug-in is used to add grain to a sequence. It has been designed to sample grain from one image and then to generate unlimited amounts of this grain with exactly the same statistics as the original. This new grain can then be applied to another image.

Figure 47. Kodak 320.

Figure 48. ReGrain.

Figure 47 shows an enlarged and exaggerated sample of grain from Kodak 320 film stock. FurnaceCore's ReGrain was used to sample the original Kodak 320 stock and synthesize a plate of grain. The result is shown in Figure 48. Note that the grain characteristics closely match the original.

Similarly, Figure 49 on page 66 is a sample from Kodak 500 film stock and Figure 50 on page 66 shows this replicated

using FurnaceCore's ReGrain.

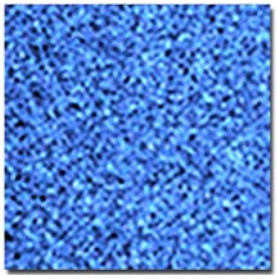

Figure 49. Kodak 500.

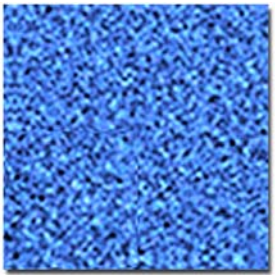

Figure 50. ReGrain.

Quick Start

Import the clip to which you want to add grain and place it on the timeline. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_ReGrain (4.1)**. There are a variety of presampled grain types to choose from. Try the different grain types using the **Stock Type** pulldown menu.

If you don't want to use one of the presampled grain types, you can also sample grain from a clip. To do so, set **Stock Type** to **Sample Grain**. Import a clip you want to sample and drag it into the clip well of the **Grain** parameter. Check **View Grain Clip** to display this clip in the Canvas. Click on the button under **Analysis Centre** and move the selection area over a plain area of the image in the Canvas. Your selection is important to get right. You should avoid any image detail or even a plain area that has luminance variations underneath the grain. The better this initial selection the better the result will be. See Figure 51. If you can't find a decent sample area on the frame, then try other frames from

the same film stock.

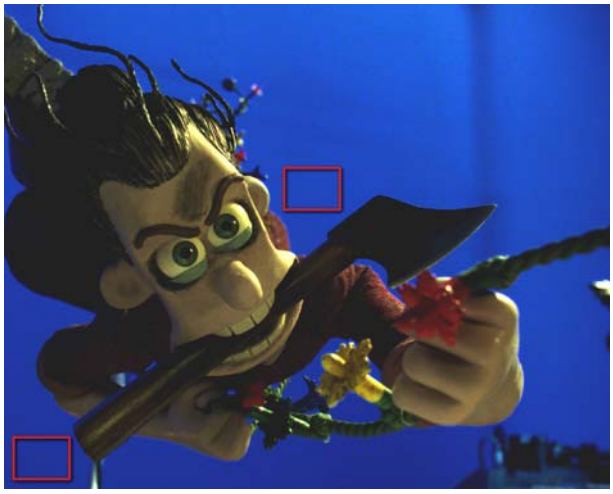

Figure 51. This shows two possible selection regions that contain no edge detail and little luminance variation.

Note *The red selection box on the Canvas is not an overlay but actual pixels on the image displayed. If you render the grain clip, the selection box will be included in the output.*

Once you have positioned the sample area, view the output of the F_ReGrain node to judge the results. The output will now contain the source image with grain from the sample image applied. Both the size and the luminance of the new grain can be manually tweaked using **Grain Scale** and **Grain Gain** respectively.

The grain is sampled on a single frame which you set using the timeline. Although the grain is sampled on only one frame, the algorithmically created grain will change from frame to frame but mirror the characteristics of the sample

grain.

There is a minimum size of the sample clip below which the statistical analysis of the grain will be unreliable. The minimum size is about 128x128.

Note *ReGrain results will differ significantly if you change the render resolution (otherwise known as the proxy resolution) setting of your sequence from lower (25%) to higher (100%), because the render resolution affects the grain computation. Therefore, you should only use ReGrain for full resolution renders. To change the render resolution of your sequence, right click on the sequence, click* ***Settings****, navigate to the* ***Render Control*** *tab, and set the resolution to 100%. Note that a user warning will be set in the user message button if you use ReGrain at less than full proxy resolution.*

Grain Stock

To add grain from a standard film stock, select from the **Stock Type** list. 2K, 4K, aperture corrected and non aperture corrected stocks are included. Individual colour channels can be selected and adjusted using the **Gain** and **Scale** parameters.

Response

In its default setting, F_ReGrain adds the same amount of grain over the whole image. However, the amount of grain on an image is normally a function of luminance. You can adjust how the amount of grain added varies with luminance. The **Low Gain, Mid Gain** and **High Gain** parameters allow you to adjust the amount of grain added to the lowlights, midtones and highlights of the image.

Checking the Result

To test that the new grain is the same as the old grain you

can select **Show - Grain Plate**.

Figure 52. Good selection area...

Figure 53. ...producing a good test plate of grain, free of artifacts.

This generates a sheet of grain with the same luminance level as the mean of the sample region. The sample region with the original grain is also displayed. It should be impossible to differentiate between the two regions. Figure 52 shows a good selection area giving a good test plate of grain in Figure 53. Figure 54 shows a poor selection area since it contains image detail. Figure 55 shows the resulting test plate, which clearly highlights the problem.

Figure 54. Bad selection area...

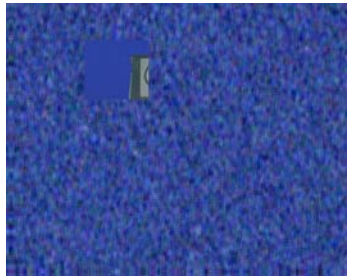

Figure 55. ...producing a poor result.

Parameters

The parameters for this plug-in are described below.

Grain - drag a clip you want to sample grain from into the clip well here. For this to work, **Stock Type** needs to be set to **Sample Grain**.

Stock Type - selects whether the grain is sampled from the grain image (**Sample Grain**) or from a set of standard stocks. 2K, 4K, aperture corrected and non aperture corrected stocks are supplied.

- **Sample Grain** - samples and reconstructs the grain characteristics from the **Grain** image.
- **<Stock><Exposure><Size>** - grain characteristics sampled from a supplied film stock. Common Fuji and Kodak stocks are supplied. The exposure can be under, over or, if left blank, non aperture corrected. The size is either 2K or 4K pixels. For example, FUJIF500 2K refers to the grain characteristics sampled from a 2K plate of Fuji Film 500 film stock non aperture corrected.

Show - sets whether to render the result or a test image.

- **Result** - shows the image with the grain applied.
- **Grain Plate** - shows a test image with the grain applied. This test image is composed from a section of the input image surrounded by a uniform solid colour sampled from the image with the grain applied (Figure 55). If the inner area is indistinguishable from the outer area, then you have a good grain sample (Figure 53).

Grain Scale - adjusts the size of the grain granules.

Grain Gain - adjusts the brightness of the grain.

Red Gain - sets the brightness of the grain in the red

channel.

Green Gain - sets the brightness of the grain in the green channel.

Blue Gain - sets the brightness of the grain in the blue channel.

Red Scale - adjusts the size of the grain granules in the red channel.

Green Scale - adjusts the size of the grain granules in the green channel.

Blue Scale - adjusts the size of the grain granules in the blue channel.

Low Gain - adjusts the gain of the grain in the lowlights.

Mid Gain - adjusts the gain of the grain in the midtones.

High Gain - adjusts the gain of the grain in the highlights.

View Grain Clip - check this to view the clip in the image well of the **Grain** parameter.

Analysis Center - change the position of the centre of the sample region. Click the button to move the selection area into a new position on the Canvas, or enter new coordinates for the centre point in the coordinates fields.

Sample Size - change the size of the sample area.

Analysis Frame - the frame to sample the grain from.

Example

The image used in the following example can be downloaded from our website. For more information, please see "Example Images" on page 5.

Rachael

In this example, we will degrain the image using F_DeNoise and then, using F_ReGrain, sample the grain from the original image and reapply it to the degrained image. Switching between the resulting outputs of F_DeNoise and F_ReGrain will show different grain but with the same characteristics.

1. Import Rachael.tif and place it on the timeline.
2. Select **Effects** > **Video Filters** > **FurnaceCore** > **F_DeNoise (4.1)**.
3. Set **Analysis Method** to **Sample Region** and position the sample region over a plain area.
4. Now load F_ReGrain using the processed output of F_DeNoise as the clip to add grain to and the original unprocessed image as the **Grain** sample clip. Make sure you have set **Stock Type** to **Sample Grain**.
5. Compare the original to the regrained output of F_ReGrain. Note that although the actual grain differs on both images, the grain characteristics are the same.

APPENDIX A

Release Notes

This section describes the requirements, new features, improvements, fixed bugs, and known bugs and workarounds for each release of FurnaceCore.

FurnaceCore 4.1v1

Release Date

October 2009

Requirements

1. Final Cut Pro 6.0.4 or above on Mac OS X.
2. Foundry FLEXlm Tools (FFT 5.0v1 or later) for floating licenses.

New Features

- The plug-ins' help dialogs now have an **Enter Serial Number** button at the bottom of the dialog. This button launches a web browser to get a license key for the plug-ins. For more information, see "Activating FurnaceCore" on page 7.
- F_DeNoise and F_Kronos now have a **Deinterlacing Method** parameter at the end of their controls. This parameter is useful when applying the plug-ins to fielded footage in order to produce progressive footage. It is also needed when using progressive footage with these two plug-ins. For more information, see F_DeNoise "Parameters" on page 22, F_Kronos "Parameters" on page 42, and "Using Progressive Footage with F_Kronos and F_DeNoise" on page 14.

Improvements

- Furnace has been rebranded to FurnaceCore.
- FurnaceCore will now render a watermark of dots rather than a black screen with a red cross should no valid license be found.
- The plug-ins now appear in the **Effects** > **Video Filters** > **FurnaceCore** menu with their version number after their name. For example, **F_Kronos** now appears as **F_Kronos (4.1)**. This allows you to have different versions of the plug-ins available on the same machine.

Bug Fixes

- Fixed issues relating to C++ object instances used as Objective C members. The construction and destruction of the C++ objects was not properly done (or at least undefined). This could potentially have led to memory and resource leaks.
- BUG ID 7250 - Some FurnaceCore plug-ins would introduce softening of the image, and possibly also a downward shift. These issues have been addressed.
- BUG ID 7541 - Inconsistent results in F_Kronos. This bug caused minor differences in the output of F_Kronos on each render. The results should now be consistent.
- BUG ID 8149 - Furnace would crash with High Precision YUV renders. This has been fixed.
- BUG ID 8151 - F_DeFlicker had motion estimation artifacts on the final frame. This has been fixed.
- BUG ID 8208 - F_DeNoise would show blue artifacts on 10-bit YUV High precision renders. This has been fixed.
- BUG ID 8506 - F_DeNoise and F_Kronos would cause Final cut Pro to hang and eventually crash when scrubbing rapidly on the timeline. This has been fixed.

- BUG ID 8597 - The sampled grain result in F_ReGrain was different in the final render. This has been fixed.
- BUG ID 8782 - ReGrain was missing its Show (Result/ Grain Plate) option. This has been fixed.
- BUG ID 8957 - FCP scaling resulted in incorrect grain sample scale in F_ReGrain, which would adversely affect results. This has been fixed.
- BUG ID 9236 - F_DeFlicker would produce blank output images if the input images were too small. This could occur, for instance, if the **render quality** setting of the output seqence was low, and the image in the viewing canvas was zoomed out. This has been resolved by not further downsizing the images DeFlicker uses internally. In extreme cases where the input images are still too small (smaller than 64x64 pixels), a user message is set.
- BUG ID 9291 - A double memory deallocation in F_ReGrain could cause the plug-in to crash. This has been fixed.

Known Bugs and Workarounds

- A note on the viewing Canvas in Final Cut Pro:

  In Final Cut Pro, there are two options for viewing the results of an applied effect:

  1) You can either scrub along the timeline directly, in which case you need to wait a bit before processed frame results are shown, or

  2) you can first render out the clip or sequence (by using **Cmd**+**R**, or by selecting one of the **Render** options in the **Sequence** menu), and then you can scrub and see results immediately.

  It is often the case - particularly when zooming in on frames - that the viewing Canvas will display the result of on-line rendering (a) differently from the result of off-line

rendering (b). In particular, footage rendered out (b) is often displayed so that the frames appear to have been slightly softened or deinterlaced. This is merely a Final Cut Pro display issue - the footage exported to file should be fine.

- Advanced effects may require access to multiple different frames on the timeline. There is a known issue in Final Cut Pro, whereby it will intermittently issue such effects with invalid frames.

  In these rare instances where Final Cut Pro has malfunctioned, FurnaceCore will display an error message. These error messages differ (depending on the way in which the frame is invalid), but will typically state that a program error related to an incorrect frame fields or frame sizes has occurred. When this occurs, you can generally safely ignore and close the error messages, and continue with your work.

  You will only encounter this when scrubbing along unrendered footage, and changing parameters rapidly. These errors are temporary, and do not apply when the clip been rendered out (by using **Cmd**+**R**, or by selecting one of the **Render** options in the **Sequence** menu), and also do not apply when exporting or saving your final footage.

  Apple is aware of this bug in Final Cut Pro, and will be investigating a solution to the problem in due course.
- BUG ID 8670 - When using plug-ins that have a clip well in their controls (for example, F_ReGrain), you may get poor results if you drop a new clip in without clearing the old one. This is because the plug-in may fail to re-analyse the clip. As a workaround, you should always clear the old clip first before applying a new clip. To do so, right-click on the clip well and select **Clear**.

- BUG ID 8688 - When rendering fielded footage, F_DeNoise and F_DirtRemoval plug-ins fail to render both fields for the final frame. One field is therefore transparent. Until this is remedied, it is recommended that you remove the last frame from the processed clip or sequence. To clearly see whether a field is transparent in the final frame, the Canvas viewer's zoom level should be set to 100%. At this zoom level, the Canvas viewer's background colour (white, black, or checkered) will show through the unrendered lines.
- BUG ID 8778 - The grain sample box for F_ReGrain can be offset from the user-clicked position, depending on the dimensions of the grain clip. The current work-arounds are to either
    - click further away from/nearer to the centre of the image, until the grain sample box is where you need it to be, or to
    - enter the offset for the grain sample box manually by typing it in.
- BUG ID 8780 - ReGrain does not deinterlace the noise sample before use.
- BUG ID 8958 - Integer sliders, such as F_DeFlicker's Window Size, behave as though they are floating point sliders. This does not affect the result of the effect. For example, the effect of changing an integer slider from 1 to 1.1 will still apply the effect with a parameter value of 1. Unfortunately, changing the slider to 1.1 will result in the effect to be reprocessed - thereby wasting time - even though there was no real change.

Furnace 4.0v2

This is a maintenance release of Furnace for Final Cut Pro.

Release Date

November 2008

Requirements

1. Final Cut Pro 6.0.4 or above on Mac OS X.
2. Foundry FLEXlm Tools (FFT 4.0v1 or later) for floating licenses.

New Features

There are no new features in this release.

Improvements

There are no improvements in this release.

Bug Fixes

Fixed instability in plug-ins caused by OS incompatibility with FLEXlm 10.8 licensing module. Upgraded FLEXlm to 10.8.6 for improved Leopard compatibility.

Known Bugs and Workarounds

There are no known bugs.

Furnace 4.0v1

This is the first release of Furnace for Final Cut Pro.

Release Date

September 2008

Requirements

1. Final Cut Pro 6.0 on Mac OS X.
2. Foundry FLEXlm Tools (FFT 4.0v1 or later) for floating licenses.

New Features

There are seven plug-ins in this release.

Improvements

This section will describe improvements to existing features in later versions.

Bug Fixes

This section will describe fixed bugs in later versions.

Known Bugs and Workarounds

There are no known bugs.

APPENDIX B

End User License Agreement

IMPORTANT: BY INSTALLING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT DO NOT INSTALL, COPY OR USE THE SOFTWARE.

This END USER SOFTWARE LICENSE AGREEMENT (this "Agreement") is made by and between The Foundry Visionmongers Ltd., a company registered in England and Wales, ("The Foundry"), and you, as either an individual or a single entity ("Licensee").

In consideration of the mutual covenants contained herein and for other good and valuable consideration (the receipt and sufficiency of which is acknowledged by each party hereto) the parties agree as follows:

Section 1. Grant of License.

Subject to the limitations of Section 2, The Foundry hereby grants to Licensee a limited, non-transferable and non-exclusive license to install and use a machine readable, object code version of this software program (the "Software") and the accompanying user guide and other documentation (collectively, the "Documentation") solely for Licensee's own internal business purposes (collectively, the "License"); provided, however, Licensee's right to install and use the Software and the Documentation is limited to those rights expressly set out in this Agreement.

Section 2. Restrictions on Use.

Licensee is authorized to use the Software in machine readable, object code form only, and Licensee shall not: (a) assign, sublicense, transfer, pledge, lease, rent, share or export the Software, the Documentation or Licensee's rights hereunder; (b) alter or circumvent the copy protection mechanisms in the Software or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software; (c) modify, adapt, translate or create derivative works based on the Software or Documentation; (d) use, or allow the use of, the Software or Documentation on any project other than a project produced by Licensee (an "Authorized Project"); (e) allow or permit anyone (other than Licensee and Licensee's authorized employees to the extent they are working on an Authorized Project) to use or have access to the Software or Documentation; (f) copy or install the Software or Documentation other than as expressly provided for herein; or (g) take any action, or fail to take action, that could adversely affect the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of The Foundry or any third party with intellectual property rights in the Software (each, a "Third Party Licensor"). Furthermore, for purposes of this Section 2, the term "Software" shall include any derivatives of the Software.

Licensee shall install and use only a single copy of the Software on one computer, unless the Software is installed in a "floating license" environment, in which case Licensee may install the Software on more than one computer; provided, however, Licensee shall not at any one time use more copies of the Software than the total number of valid Software licenses purchased by Licensee.

Furthermore, the Software can be licensed on an "interactive" or "non-interactive" basis. Licensee shall be authorized to use a non-interactive version of the Software for rendering purposes only (i.e., on a CPU, without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Licensee shall be authorized to use an interactive version of the Software for both interactive and non-interactive rendering purposes, if available.

Finally, if the Software is an "Educational Version," Licensee may use it only for the purpose of training and instruction, and for no other purpose. Educational Versions of the Software

may not be used for commercial, professional or for-profit purposes.

Section 3. Back-Up Copy.

Notwithstanding Section 2, Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location owned or leased by Licensee in order to provide a back-up in the event of destruction by fire, flood, acts of war, acts of nature, vandalism or other incident. In no event may Licensee use the back-up copy of the Software or Documentation to circumvent the usage or other limitations set forth in this Agreement.

Section 4. Ownership.

Licensee acknowledges that the Software and Documentation and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry and the Third Party Licensors. Licensee shall not remove, or allow the removal of, any copyright or other proprietary rights notice included in and on the Software or Documentation or take any other action that could adversely affect the property rights of The Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this Agreement, Licensee shall reproduce in and on all such copies any copyright and/or other proprietary rights notices provided in and on the materials supplied by The Foundry hereunder. Nothing in this Agreement shall be deemed to give Licensee any rights in the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of The Foundry or any Third Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of The Foundry or any Third Party Licensor in Licensee's promotion or publicity without The Foundry's express written approval.

Section 5. License Fee.

Licensee understands that the benefits granted to Licensee hereunder are contingent upon Licensee's payment in full of the license fee payable in connection herewith (the "License Fee").

Section 6. Taxes and Duties.

Licensee agrees to pay, and indemnify The Foundry from claims for, any local, state or national tax (exclusive of taxes based on net income), duty, tariff or other impost related to or arising from the transaction contemplated by this Agreement.

Section 7. Limited Warranty.

The Foundry warrants that, for a period of ninety (90) days after delivery of the Software: (a) the machine readable electronic files constituting the Software and Documentation shall be free from errors that may arise from the electronic file transfer from The Foundry and/or its authorized reseller to Licensee; and (b) to the best of The Foundry's knowledge, Licensee's use of the Software in accordance with the Documentation will not, in and of itself, infringe any third party's copyright, patent or other intellectual property rights. Except as warranted, the Software and Documentation is being provided "as is." THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND The Foundry DISCLAIMS ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDLESS OF WHETHER The Foundry KNOWS OR HAS REASON TO KNOW OF LICENSEE'S PARTICULAR NEEDS. The Foundry does not warrant that the Software or Documentation will meet Licensee's requirements or that Licensee's use of the Software will be uninterrupted or error free. No employee or agent of The Foundry is authorized to modify this limited warranty, nor to make additional warranties. No action for any breach of the above limited warranty may be commenced more than one (1) year after Licensee's initial receipt of the Software. To the extent any implied warranties may not be disclaimed under

applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO NINETY (90) DAYS AFTER DELIVERY OF THE SOFTWARE TO LICENSEE.

Section 8. Limited Remedy.

The exclusive remedy available to the Licensee in the event of a breach of the foregoing limited warranty, TO THE EXCLUSION OF ALL OTHER REMEDIES, is for Licensee to destroy all copies of the Software, send The Foundry a written certification of such destruction and, upon The Foundry's receipt of such certification, The Foundry will make a replacement copy of the Software available to Licensee.

Section 9. Indemnification.

Licensee agrees to indemnify, hold harmless and defend The Foundry and The Foundry's affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives (collectively, the "Released Parties") from all claims, defense costs (including, but not limited to, attorneys' fees), judgments, settlements and other expenses arising from or connected with the operation of Licensee's business or Licensee's possession or use of the Software or Documentation.

Section 10. Limited Liability.

In no event shall the Released Parties' cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands or actions arising out of or relating to this Agreement (or the Software or Documentation contemplated herein) exceed the License Fee paid to The Foundry or its authorized reseller for use of the Software. Furthermore, IN NO EVENT SHALL THE RELEASED PARTIES BE LIABLE TO LICENSEE UNDER ANY THEORY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS OR LOSS OF PROFITS) OR THE COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES,

REGARDLESS OF WHETHER THE RELEASED PARTIES KNOW OR HAVE REASON TO KNOW OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE. No action arising out of or related to this Agreement, regardless of form, may be brought by Licensee more than one (1) year after Licensee's initial receipt of the Software; provided, however, to the extent such one (1) year limit may not be valid under applicable law, then such period shall limited to the shortest period allowed by law.

Section 11. Term; Termination.

This Agreement is effective upon Licensee's acceptance of the terms hereof (by clicking on the "Accept" button) and Licensee's payment of the License Fee, and the Agreement will remain in effect until termination. If Licensee breaches this Agreement, The Foundry may terminate the License granted hereunder by notice to Licensee. In the event the License is terminated, Licensee will either return to The Foundry all copies of the Software and Documentation in Licensee's possession or, if The Foundry directs in writing, destroy all such copies. In the later case, if requested by The Foundry, Licensee shall provide The Foundry with a certificate signed by an officer of Licensee confirming that the foregoing destruction has been completed.

Section 12. Confidentiality.

Licensee agrees that the Software and Documentation are proprietary and confidential information of The Foundry and that all such information and any communications relating thereto (collectively, "Confidential Information") are confidential and a fundamental and important trade secret of The Foundry. Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" such Confidential Information, and shall advise any recipients of Confidential Information that it is to be used only as authorized in this Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information available to any other of Licensee's employees or to any third parties without the express written consent of The Foundry. Licensee agrees to segregate, to the extent it can be reasonably

done, the Confidential Information from the confidential information and materials of others in order to prevent commingling. Licensee shall take reasonable security measures, which such measures shall be at least as great as the measures Licensee uses to keep Licensee's own confidential information secure (but in any case using no less than a reasonable degree of care), to hold the Software, Documentation and any other Confidential Information in strict confidence and safe custody. The Foundry may request, in which case Licensee agrees to comply with, certain reasonable security measures as part of the use of the Software and Documentation. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that The Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or equitable relief as may be deemed proper by a court of competent jurisdiction.

Section 13. Inspection.

Licensee shall advise The Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit The Foundry or its authorized agents to inspect all such locations during normal business hours and on reasonable advance notice.

Section 14. Nonsolicitation.

Licensee agrees not to solicit for employment or retention, and not to employ or retain, any of The Foundry's current or future employees who were or are involved in the development and/or creation of the Software.

Section 15. U.S. Government License Rights.

The Software, Documentation and/or data delivered hereunder are subject to the terms of this Agreement and in no event shall the U.S. Government acquire greater than

RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication or disclosure by the U.S. Government is subject to the applicable restrictions of: (i) FAR §52.227-14 ALTS I, II and III (June 1987); (ii) FAR §52.227-19 (June 1987); (iii) FAR §12.211 and 12.212; and/or (iv) DFARS §227.7202-1(a) and DFARS §227.7202-3.

The Software is the subject of the following notices:

* Copyright (c) 2009 The Foundry Visionmongers, Ltd.. All Rights Reserved.

* Unpublished-rights reserved under the Copyright Laws of the United Kingdom.

Section 16. Survival.

Sections 2, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17 and 18 shall survive any termination or expiration of this Agreement.

Section 17. Import/Export Controls.

To the extent that any Software made available hereunder is subject to restrictions upon export and/or reexport from the United States, Licensee agrees to comply with, and not act or fail to act in any way that would violate, the applicable international, national, state, regional and local laws and regulations, including, without limitation, the United States Foreign Corrupt Practices Act, the Export Administration Act and the Export Administration Regulations, as amended or otherwise modified from time to time, and neither The Foundry nor Licensee shall be required under this Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations.

SECTION 18. MISCELLANEOUS.

This Agreement is the exclusive agreement between the parties concerning the subject

matter hereof and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject. This Agreement may be modified only by a written instrument signed by both parties. If any action is brought by either party to this Agreement against the other party regarding the subject matter hereof, the prevailing party shall be entitled to recover, in addition to any other relief granted, reasonable attorneys' fees and expenses of litigation. Should any term of this Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms of this Agreement. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches. This Agreement shall be governed by, and construed in accordance with English Law.

INDEX

A-Z