

PFClean 2.1 Tutorial

Cut Tool



Introduction	3
Download Footage	3
Setting up footage	3
Loading a clip into the cut tool	3
Edit point detection	4
Reviewing detected edit points	5
Correcting detected edit point	6
Committing	6
Automatic edit point detection criteria	8

Introduction

The purpose of the cut tool is to split a large clip up into a number of smaller sub-clips such that each sub-clip is a continuous take, i.e. contains no edits, fades, transitions etc... For example, if a whole reel of film has been scanned and imported into PFClean as a single clip; then the cut tool must be used to split this up into a number of smaller sub-clips, one sub-clip per edit as a minimum.

Cutting up a clip into separate continuous takes serves 2 main purposes:

1. it makes the clip more manageable, especially in the above example of a whole reel of film being imported as a single clip!
2. certain clean up operations, in particular the optical flow used for dirt/dust/scratch detection, require there to be no edits in the clip being processed. (Optical flow compares the differences between adjacent frames and if there is an edit in the clip being processed, such that adjacent frames don't relate to the same scene, then the calculations will produce invalid results).

Additionally the cut tool can be used to further sub-divide an already continuous take into a number of smaller sections if this is desirable to the user, for example, to make a long continuous take more manageable, or if the different sections require different cleanup operations on them.

Cutting up a clip into sub-clips is one of the first actions you should take after importing a clip into the system, and it certainly must be done before you start any cleanup work.

Download Footage

This tutorial requires you to download and uncompress the following footage to an easily accessible storage location: http://www.thepixelfarm.co.uk/Footage/New_Filmlight_Footage.tar.gz

Setting up footage

Load the "New_Filmlight_Footage" clip by dragging the folder into the media bins from the file browser.

Loading a clip into the cut tool

Switch into the cut tool using the "Cut" button on the lower-left side of the interface. Drag and drop a clip from the Media Bins into the cut tool.

Loading the tutorial clip into the cut tool should result in this:



Note that to load a clip into the cut tool it must not be in use elsewhere in the system, e.g. active in the edit tool, otherwise the drag and drop action will show a "no-entry" icon and fail.

It is also possible to load a clip into the cut tool by selecting it in the timeline and pressing the "Cut" button below the timeline.

Edit point detection

Scrubbing through the tutorial clip it should be obvious that the clip contains 2 distinct edit points, at frames 31 and 61. These 2 edits lead to 3 separate continuous takes:

1. Frames 01-30, walking down corridor
2. Frames 31-60, looking out of window
3. Frames 61-90, framed by "Obsession!"

Its easy enough to identify these by hand in this small tutorial clip, but this soon becomes impractical for larger clips.

The cut tool provides both automatic and manual means of identifying edit points. The manual method is described later. To automatically identify the edit points simply hit the "Detect" button. After approximately 10 seconds of analysis this should be complete.

Reviewing detected edit points

The screenshot below shows the results of doing an automatic edit point detection on the tutorial clip:



The blue markers in the scrub bar at frames 31 and 61 show the edit points that have been identified.

The "Cuts" list details the sub-clips that will be created if the current set of edit points is committed to; specifically it lists the in and out frames, the "score" of these (described below), and the sub-clip length and name.

Initially the sub-clips names are an arbitrary default value, click on the name field of an already selected item in the list to edit the name and enter something more meaningful. In future screenshots in this tutorial the sub-clips have been named "corridor", "window" and "obsession".

The "score" fields detail the confidence with which the cut tool believes that an edit point exists at that frame - the higher the score, the greater the confidence. Use the "Threshold" slider to change the threshold above which the cut tool considers the score to be an edit point - decreasing the threshold slider helps to pick up more subtle edit points but risks accidentally mis-classifying a frame as an edit point. The "Min length" slider sets a minimum frame length for the sub-clips. Note that changes to these slider are automatically reflected in the cuts list - no re-analysis of the clip is necessary.

Selecting any sub-clip in the cuts list jumps to the in frame of that sub-clip in the player windows. Similarly, the relevant sub-clip is always highlighted in the cuts list as the clip is played back and scrubbing through.

To the right of the cuts list is the "Preview" panel which shows a number of frames for the sub-clip selected in the cuts list. The preview panel can be used to determine if the the edit points have been correctly located.

By default the preview panel shows the in and out frames of the sub-clip selected in the cuts list. Optionally it can also shows the frame before the current sub-clip's in point and the frame after its out point. Enable this option by ticking the un-checked item in the preview panel so that 4 frames are visible, and ensure that the middle sub-clip is active in the cuts list:



In the above screenshot the "window in" and "window out" icons show the first and last frame of the "window" sub-clip. These pair of frames need to show a similar images otherwise an edit point in the middle of the sub-clip has been overlooked.

The "Before window in" icon shows the frame immediately prior to "window in". These pair of frames need to show different images otherwise the in point has been set too late. Similarly, "After window out" shows the frame immediately subsequent to "window out". These pair of frames need to show different images otherwise the out point has been set to early.

So, in the screenshot above you can say with confidence that the 2 cuts that form the in and the out points of the "window" sub-clip are correct. The sub-clip starts with the window scene (the previous frame having been the corridor) and ends with the window scene (the next frame being "obsession!").

Correcting detected edit point

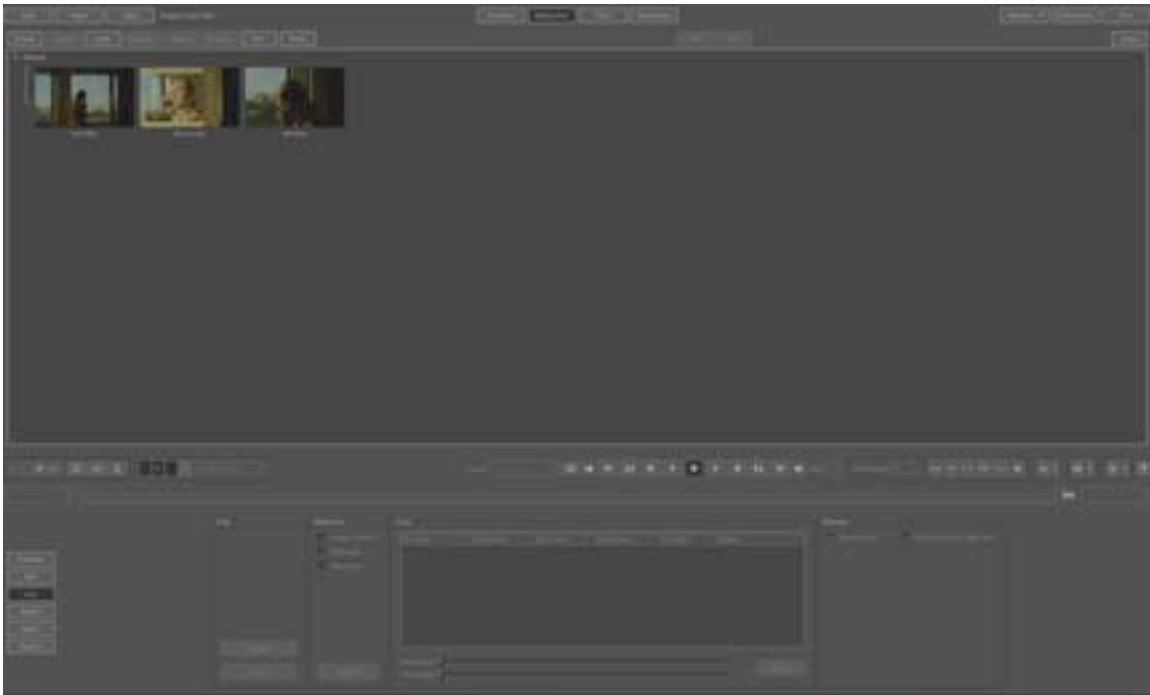
Edit points may be manually identified by navigating to the appropriate frame and pressing the "Insert" button to add an edit point. If an edit point already exists at that frame the buttons text changes to "Remove" and the edit point is deleted.

This Insert/Remove button can be used to both correct the automatically detected edit points and to manually cut up a clip.

Committing

Once you are happy that the sub-clips listed in the cuts list are correct, press the "Commit" button to create the sub-clips. Note: once the sub-clips have been created there is no way of reverting back to the original clip - so make sure the cuts list is correct before committing to it.

The player window should revert back to the Media Bins view. The original clip will have been removed and the new sub-clips added:



(If you entered the cut-tool via the timeline then upon committing the view returns to the timeline with the clip you have just cut replaced by the sub-clips).

A new sequence clip is created which contains the sub-clips correctly inserted into it in such a way that it exactly mimics the original clip. This sequence clip can be dropped into the timeline, or exported, for when you need to see the sub-clips in context. However, its important to remember that whilst this sequence clip outwardly has the same appearance as the original clip it is infact a collection of numerous independent smaller clips each of which can be worked upon individually and in isolation.

The image below shows the sequence clip loaded into the timeline.



When the sub-clip are created they inherit certain information from the "parent" clip. Specifically this information is: ROI, defect map mode, colour space, deinterlace settings and grain parameters (but not any degrain/regrain effects). If a clip contains any more information than the above then the user is warned when the clip is loaded into the cut tool that such information will be lost when the clip is cut.

Automatic edit point detection criteria

The automatic analysis can use 3 different criteria to identify edit points. These 3 criteria, which can be used in any combination, are:

1. Image content - adjacent frames are analysed and, if their content differs significantly, and edit point is assumed to occur.
2. Timecode - a discontinuity in the timecode information between adjacent frames is assumed to be an edit point.
3. Edgecode - a discontinuity in the edgecode information between adjacent frames is assumed to be an edit point.

Specify which method(s) you wish to use by the checkboxes above the "Detect" button before running the automatic detection.

Not all image formats have timecode and/or edgecode information embedded within them. If this information is not available for a particular clip then the relevant options are disabled. Image content analysis is always available.

Where applicable it is usually quicker to look for edit points based on timecode and/or edgecode discontinuities since this only involves minimal processing of the frame headers and not a detailed analysing of the full resolution frame content itself. However, just because a clip has timecode/edgecode information present does not automatically mean it can be used to detect edit points since such information may have been overwritten at some earlier point in the scanning/preparation of the clip resulting in the the original discontinuities being "lost".

In order to use the timecode embedded within a clip it is imperative that the frame rate of the project into which the clip is imported matches the frame rate used in the timecode. If there is a mismatch, say the clip's

timecode is based on 25fps but its imported into a 24fps project, then the timecode of some frames will be invalid. If this occurs then the automatic detection of the edit points will abort.

The tutorial sequence has a timecode embedded within it and this can be used to detect edit points. This sequence was shot at 25fps and so the clip must be used in a project with a matching frame rate. Whilst the sequence also has an edgecode, this information has been "flattened" at some time in its life prior to being used in PFClean.