Video Theory II

Technology For Performers

Frame Rate
(continued)

A quick refresher…

- **Frame Rate** (fps, Hz)
  - The number of still images played each second in order to create the illusion of movement in film
  - On a film camera/projector, the number of times per second that the shuttle advances one frame
  - Above 10-12 fps we see *motion*
- Original film standard frame rate = 24 fps

A quick refresher…

- **Flicker Rate** (fps, Hz)
    - The number of times per second that the shutter on a film projector opens, revealing light. A higher flicker rate creates the illusion of a higher frame rate without actually using more film
- Typical film flicker rate = 48hz, 72hz
    - Note that this is exactly 2x or 3x our base frame rate of 24 fps
- By flashing each frame 2 or 3 times we satisfy the **flicker fusion threshold** - the point at which our eye no longer perceives the ‘flickering’ of light caused by the shutter opening and closing

A quick refresher…

- Along comes TV, requiring new standards
- Enter **interlacing**, a way of ‘optimizing’ for television broadcast
    - Interlacing divides the screen up into 2 fields (sets of slices), which refresh fast enough to satisfy the flicker fusion threshold
    - These fields refresh at a rate 60Hz (which coincides with AC power in North America)
    - This gives us the frame rate of **60i** (60 interlacing fields per second), which is sometimes referred to as 30 fps (since we see 30 full frames per second)

A quick refresher…

- Along comes **colour** TV, ruining everything
- To prevent the chroma carrier frequency (for colour information) from interfering with the audio, the frame rate is slowed down 0.1%
- This gives us the NTSC standard of **59.94i** or **29.97 fps**

So, what happens when we want to transfer something that was shot on film (24 fps) to NTSC (29.97 fps) so that we can play it on TV?

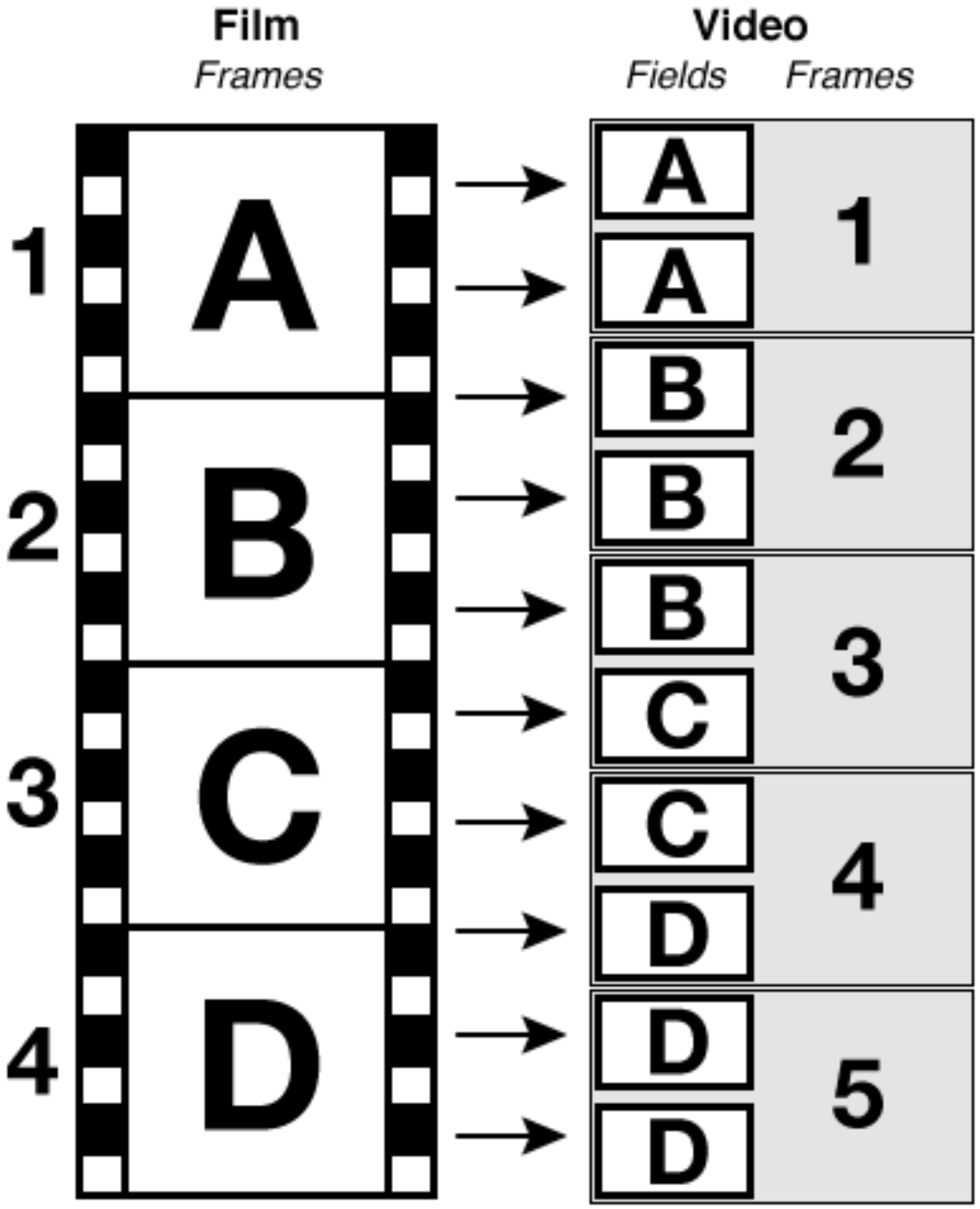
Telecine

- **Telecine** is the process of transferring film to video
    - It is also the name of the piece of equipment used for this transfer
- This process uses a technique called **pulldown** to convert the frame rate from 24 fps to 29.97 fps
    - There a few variations of the the pulldown technique, but the standard is called a **2:3 pulldown**

2:3 Pulldown

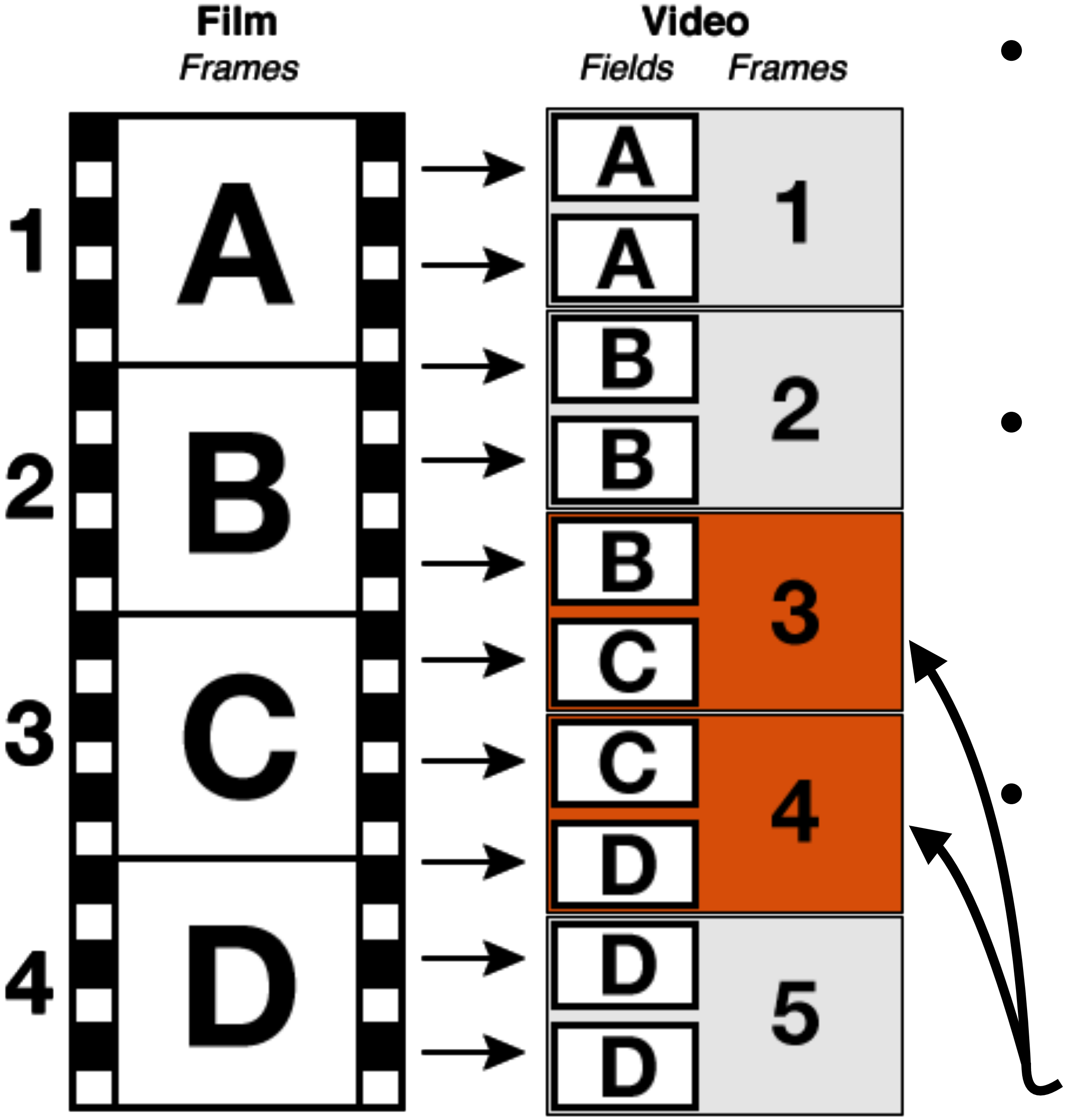
- The 2:3 pulldown process starts with the film frame rate being slowed down by 0.1% to **23.976 fps**
    - This gives us a nicer ratio to work with:
        - 23.976/29.97 = 4/5
        - Now we just need to fit 4 frames of 23.976 fps film into 5 frames of 29.97 fps video

2:3 Pulldown



- To accomplish this, we split the frames across our interlacing fields as follows:
    - The first *frame* of film occupies the first and second *fields* of video (1 frame of video)
    - The second *frame* of film occupies the third, fourth and fifth *fields* of video (1.5 frames of video)
    - The third *frame* of film occupies the sixth and seventh *fields* of video (1 frame of video)
    - The fourth *frame* of film occupies the eighth, ninth and tenth *fields* of video (1.5 frames of video)
- We have now spread 4 frames of film across 5 frames of video.

Cadence



- This method is called 2:3 pulldown because of its 2-3-2-3 pattern, or **cadence**.

- Although this 2:3 method is the ‘classic’ approach, it has its problems.

- Every 4th and 5th video frame we have a combination of 2 film frames. These are referred to as **dirty frames**.

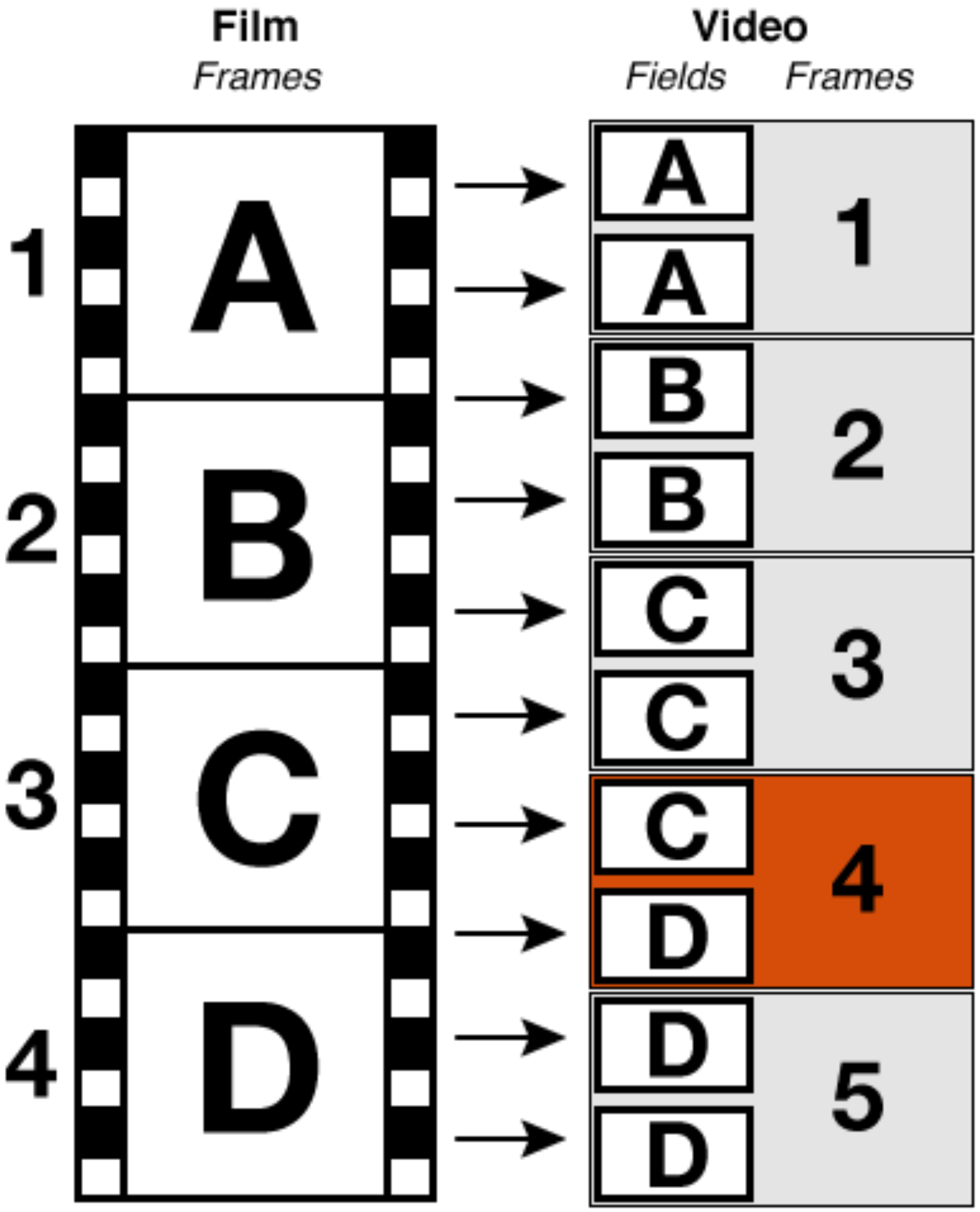
Advanced Pulldown

- Modern technology allows us to buffer more than one frame, which means we can make use of other cadences. Sometimes this technique is called *advanced pulldown*.

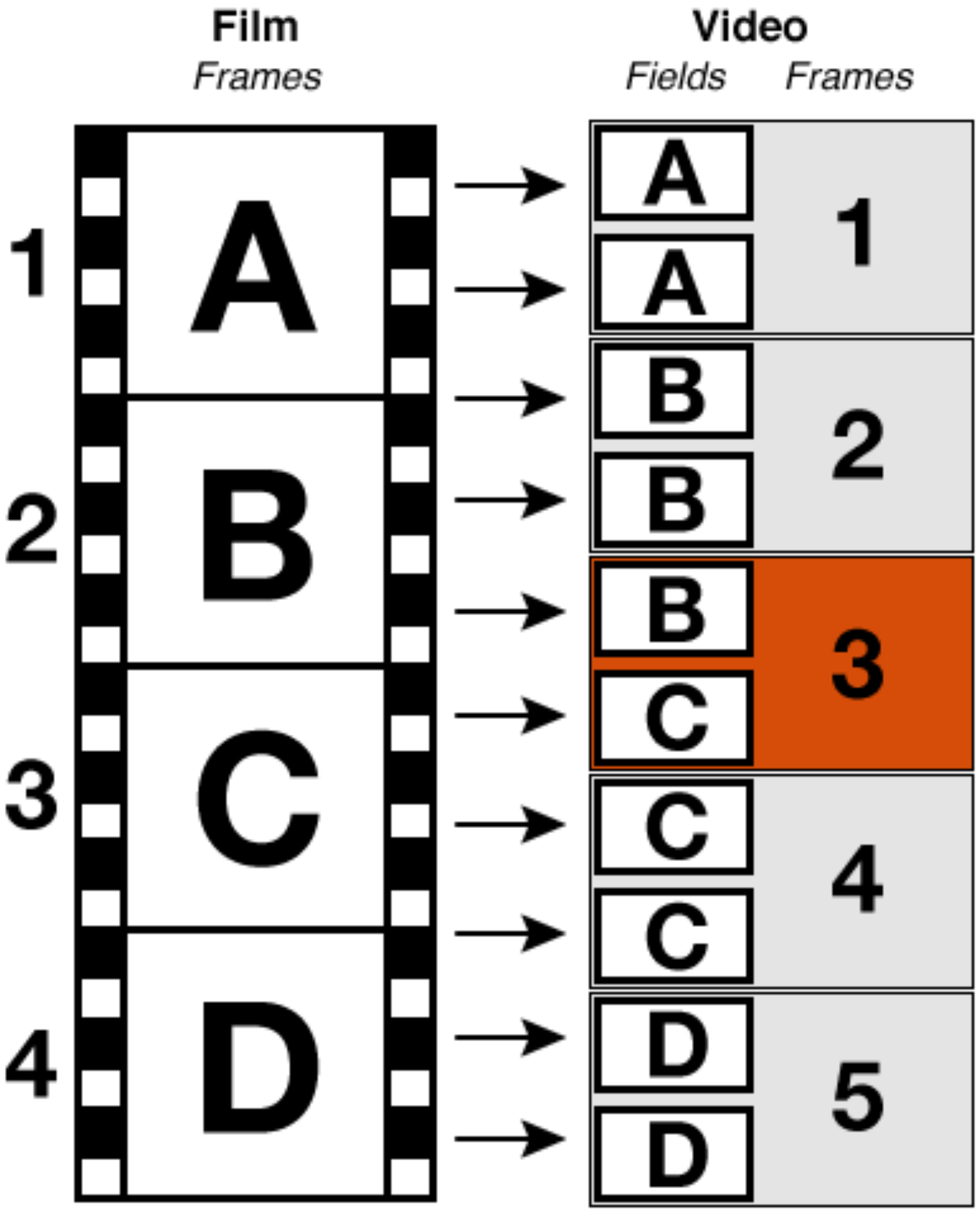
- The 2-3-3-2 or 2-2-3-3 cadences produce 4 clean frames and only 1 dirty frame every time the pattern cycles

- This helps when converting the interlaced video *back* to a **progressive** format, because the dirty frame can simply be dropped without losing any content

Cadence



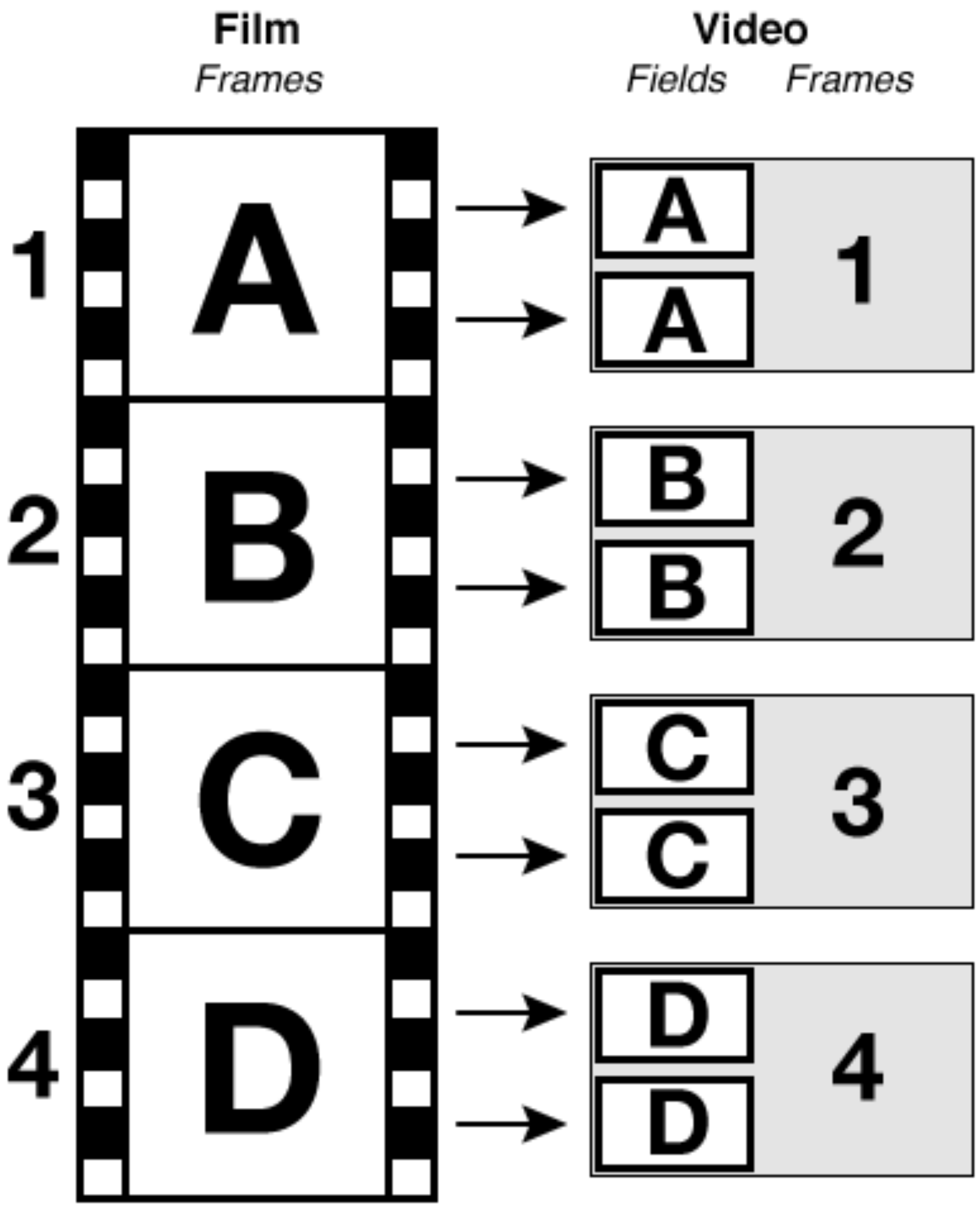
2-2-3-3 Pulldown



2-3-3-2 Pulldown

So… what happens when we want to transfer something that was shot on film (24 fps) to PAL (25 fps) so that we can play it on TV somewhere else in the world?

24 @ 25



- The PAL standard of 25 fps is so close to the film standard of 24 fps that we just just **speed up** the frame rate by **4%**
- To retain sync, the audio must also be sped up by 4%
  - This results in a pitch increase of 0.679 semitones.
  - This is enough to be noticeable, so it is often corrected afterwards using pitch-shifting
- The interlacing process is simplified as well. A **2:2 pulldown** technique is used and each frame is simply split into two fields

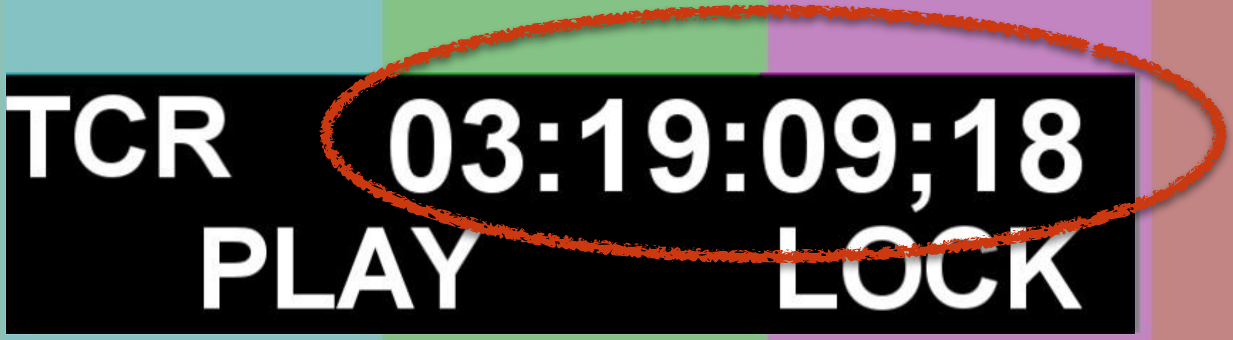
TI:ME:CO:DE

Timecode

- The entrance of electronic videotape editing brought about the need for some more organizational tools
- Enter **timecode**
  - Standardized by the Society of Motion Picture and Television Engineers (SMPTE) in the late 1960's
  - SMPTE timecode assigns each individual frame with its own unique identifier
  - The format is HH:MM:SS:FF (hours, minutes, seconds, frames)

Timecode

- You’ve seen timecode before, in Final Cut Pro or maybe even in Pro Tools
- Timecode also appears as a ‘burn-in’ on in-production versions of films so that **sync** can be maintained throughout a production

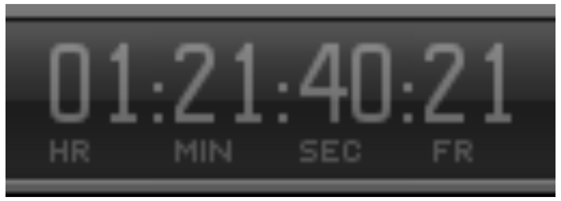


timecode!

Timecode

- There are timecode standards for each frame rate that we have discussed, and most of them are what we refer to as **Non-Drop Frame** timecode

- **Non-Drop Frame** timecode functions as you would expect:
    - Remember our format…



    - The *frames* field of the timecode starts at 0, counts up as many fps as are in that particular standard, then resets to 0
        - Don’t forget that ‘0’ counts as a frame. If we are at 30 fps, the numbers will count from 0-29.

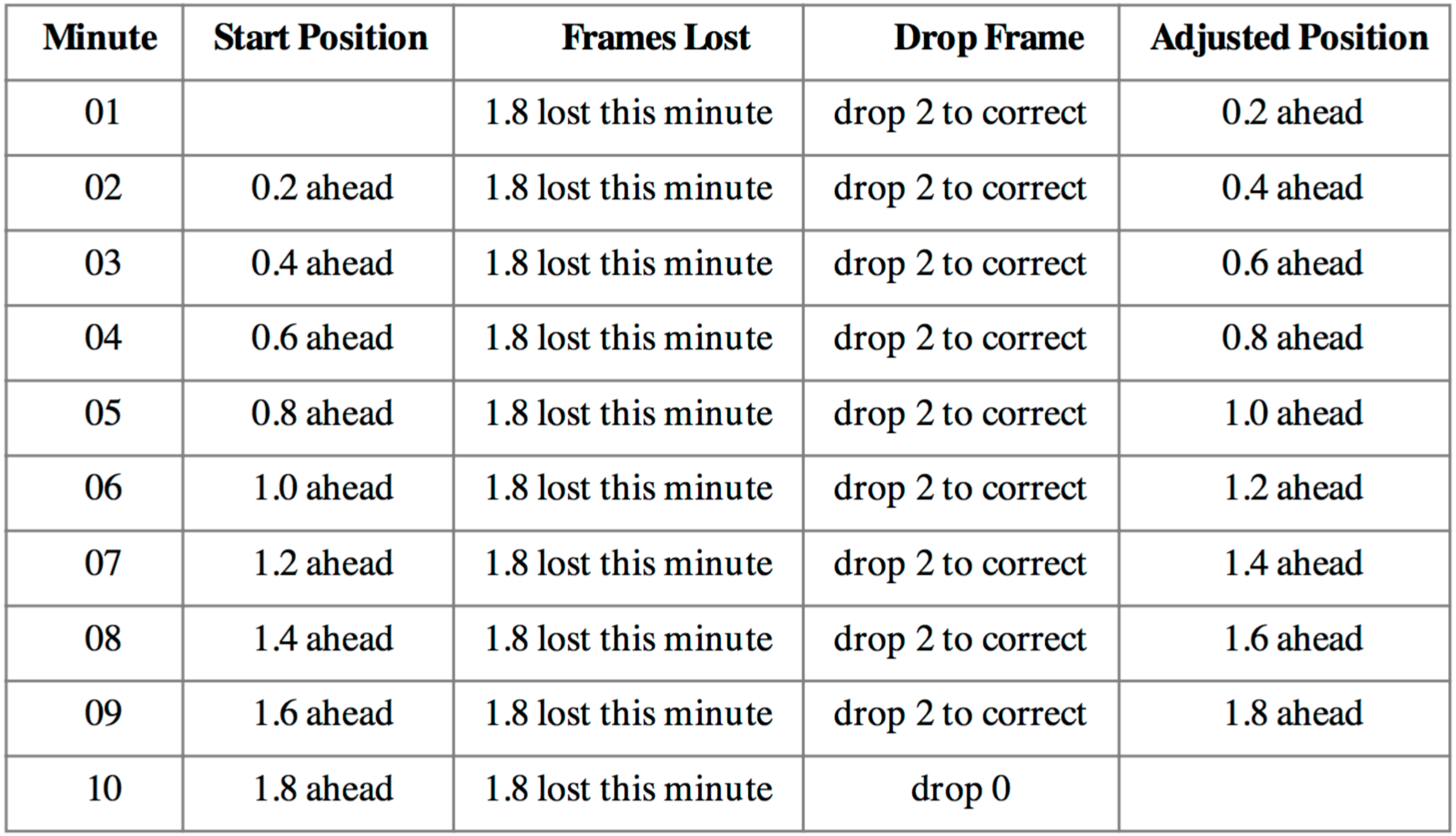
Drop Frame

- **Drop frame** works *almost* the same way
- Remember that time we slowed down 30 fps by 0.1% so that we could make colour TV broadcasts work? Well, it turns out that the slight slowing down means we fall *out of sync* with real time
  - In Non-Drop Frame 30 fps, an hour of footage contains 108,000 frames
  - Slow that down by 0.1% and we've only made it through 107,892 frames at the 1 hour mark (real time)
  - This difference accounts for 3.6 seconds of footage
- This is an important consideration for broadcast - you don't want the end of your program material chopped off!
- The solution?

Drop Frame

- Every *minute*, we skip ahead two frames in the timecode counter (frames 00 and 01 are not counted)
    - This almost fixes the problem, but actually overcompensates slightly, so…
- Every *10 minutes* (10, 20, 30, 40, 50, and 60 minutes) we don't skip any frames in the timecode counter
    - It is important to note - the video frames themselves are not skipped. In fact, nothing about the video itself changes, the frames are just assigned different timecode values
    - This is kind of like leap years on our calendar. Every four years we 'add' a day to compensate for our counting system having a small amount of error built-in

Drop Frame

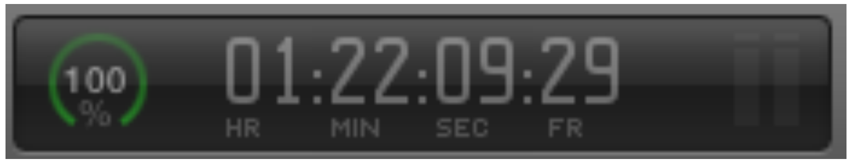


Minute	Start Position	Frames Lost	Drop Frame	Adjusted Position
01		1.8 lost this minute	drop 2 to correct	0.2 ahead
02	0.2 ahead	1.8 lost this minute	drop 2 to correct	0.4 ahead
03	0.4 ahead	1.8 lost this minute	drop 2 to correct	0.6 ahead
04	0.6 ahead	1.8 lost this minute	drop 2 to correct	0.8 ahead
05	0.8 ahead	1.8 lost this minute	drop 2 to correct	1.0 ahead
06	1.0 ahead	1.8 lost this minute	drop 2 to correct	1.2 ahead
07	1.2 ahead	1.8 lost this minute	drop 2 to correct	1.4 ahead
08	1.4 ahead	1.8 lost this minute	drop 2 to correct	1.6 ahead
09	1.6 ahead	1.8 lost this minute	drop 2 to correct	1.8 ahead
10	1.8 ahead	1.8 lost this minute	drop 0	

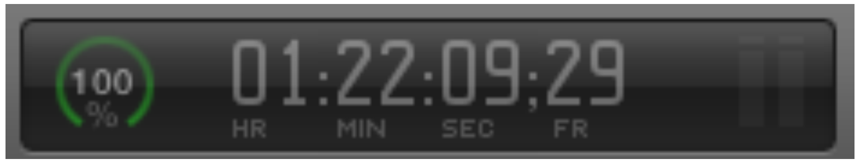
- Using this approach, Drop Frame timecode lines up with ‘real’ time every 10 minutes
- As an added bonus,10 minutes of NTSC video contains exactly 17,982 frames which means we have a nice clean frame boundary at each 10 minute marker

Timecode

- In Final Cut Pro X:
- Non-Drop Frame timecode is displayed:
  - HH:MM:SS:FF



- Drop Frame timecode is displayed:
  - HH:MM:SS;FF



- *Note the semicolon between seconds and frames, used only in Drop Frame

Questions?