

Interlace, Interleave, en Field Dominance

Videomonitoren en televisies zijn “raster-based display systems”: elk beeld (frame) wordt lijn per lijn om het scherm gezet (625 lijnen in [pal](#) en [secam](#), 525 in [NTSC](#)).

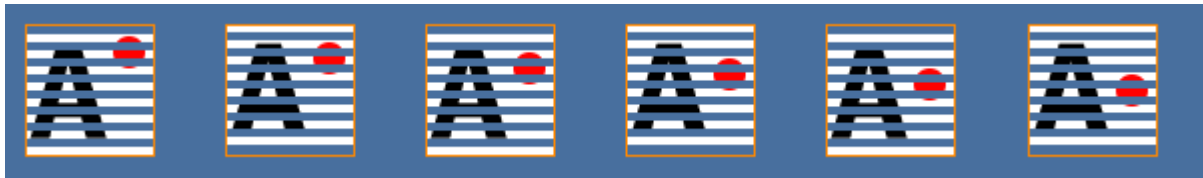
Progressive vs. Interlaced

Progressive scanning is makkelijk. Elk frame wordt volledig opgenomen/weergegeven. Dit is het geval in elke film-camera (waar zelfs geen lijnen aan te pas komen), en in videocamera's die opnemen in progressive scan. De frame rate duidt het aantal frames per seconde aan (24/25/30).



Hierboven 3 elkaar opvolgend frames (progressive scan)

Diezelfde drie frames weggeschreven als interlaced video geeft het volgende resultaat (we gaan voor de rest van de uitleg uit van een PAL-video signaal, dus 25fr/sec).



In plaats van uit volledige frames bestaat het signaal uit *fields*, waarbij een *field* op $1/50$ sec (de helft van $1/25$ sec) de helft van het beeld op de helft van het totaal aantal beschikbare lijnen wegschrijft, en de volgende $1/50$ sec (nogmaals de helft van $1/25$ sec) de ander helft van het beeld op de andere helft van het totaal aantal lijnen wegschrijft. Op $2 \times 1/50$ sec ofwel $1/25$ sec is het volledige beeld opgenomen/weggeschreven. De ene helft van de lijnen noemen we *upper fields*, de andere helft *lower fields*.

Het zou duidelijk moeten zijn dat de optelsom van twee fields niet hetzelfde is als één frame progressive scan. Een beetje vereenvoudigd kan je stellen dat bij progressive scan elk frame een foto is van $1/25$ seconde, terwijl interlaced video twee foto's door mekaar mengt die met een tussentijd van $1/50$ sec na mekaar zijn opgenomen.

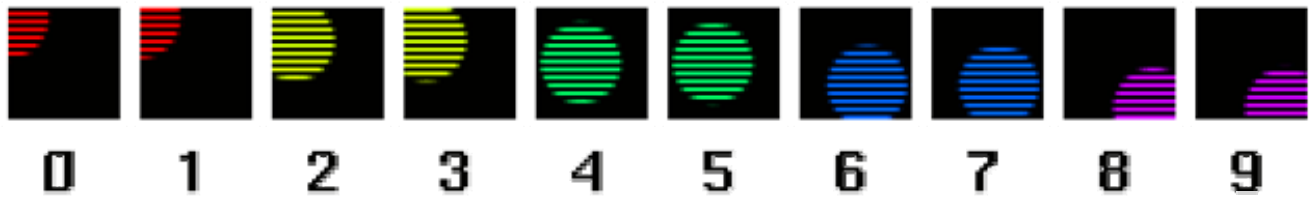
Nemen we een voorbijvliegende bal als voorbeeld. En we bekijken fields 0 tot 9. Dan ziet dat er in interlaced video als volgt uit :



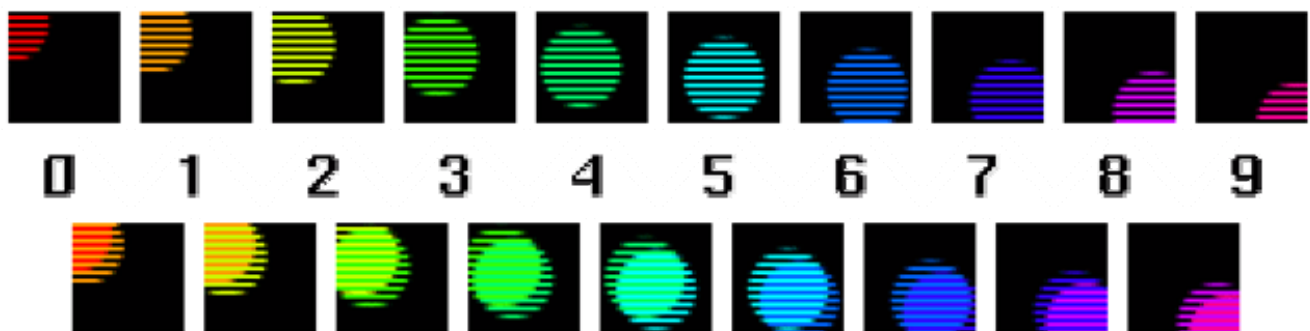
en in progressive scan als volgt



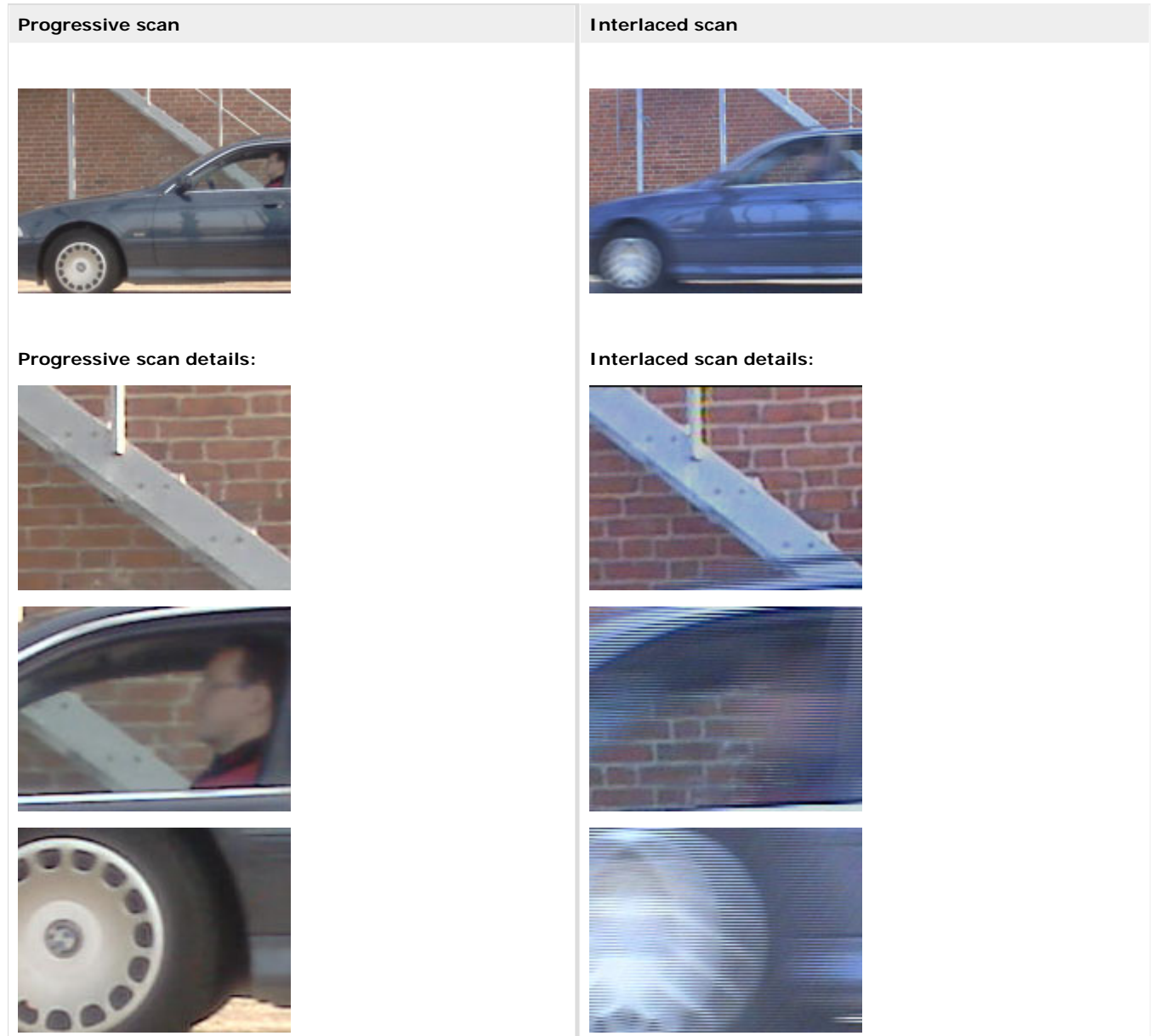
maar nooit als :



Als we een momentopname van de monitor zouden maken, of een still uit een videobeeld distilleren zonder met de interlacing rekening te houden, dan zouden we dus het volgende zien (onderste lijn) :



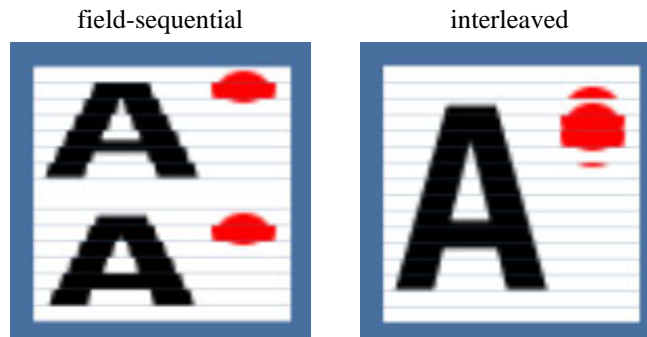
Of een voorbeeld uit de echte wereld. Omwille van wat hierboven staat zou het duidelijk moeten zijn dat het effect van interlacing het duidelijkst is bij bewegende objecten (of een bewegende camera uiteraard) :



Framing en Interleaving

Een interlaced video stream (zoals bv DV of Betacam) wordt bij het inladen in de computer *ge-deïnterlaced*, en als zelfstandige frames behandeld/bewerkt in FCP, AVID, PREMIERE, ... , om pas bij het uitladen, of naar de video monitor sturen, terug in interlaced video omgezet te worden.

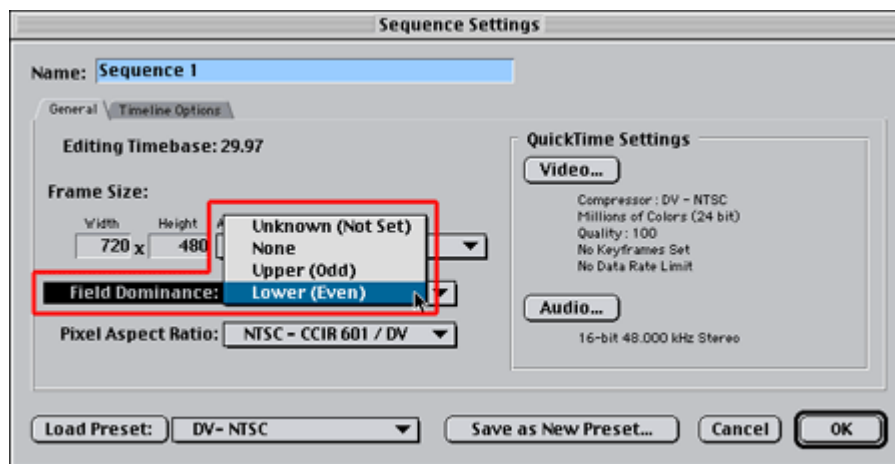
Dat wil niet zeggen dat de indeling als interlaced verloren gaat. De *upper en lower fields* zitten in de computer als field-sequential of interleaved opgeslagen, het montageprogramma en de grafische kaart zorgen er voor dat de beelden op de juiste manier op de monitor worden weergegeven (dit alles gebeurt nagenoeg zonder enige tussenkomst van de gebruiker).



Field Dominance

Als dus een videostream gedigitaliseerd wordt, zal de gebruiker een keuze moeten maken moet elk frame beginnen met Upper field eerst, of met lower field eerst (eerst zijnde lijn 1 en om het nog moeilijker te maken soms ook lijn 0). Die keuze noemt men de *field dominance*.

Hieronder een voorbeeld uit Final Cut Pro sequence settings :



De field dominance van je video stream is belangrijk, omdat deze keuze behouden moet blijven doorheen alle bewerkingen van het gegeven beeld (digitalisatie, montage, export/import als Quicktime, exporteren/importeren van frames naar teken- of andere programma's, en uiteindelijk de playout).

Tot slot misschien het belangrijkste :

Alle DV streams zijn lower-field-first.

ter info voor de techneuten onder ons : ["Definitions: F1/F2, Interleave, Field Dominance, and More"](#)