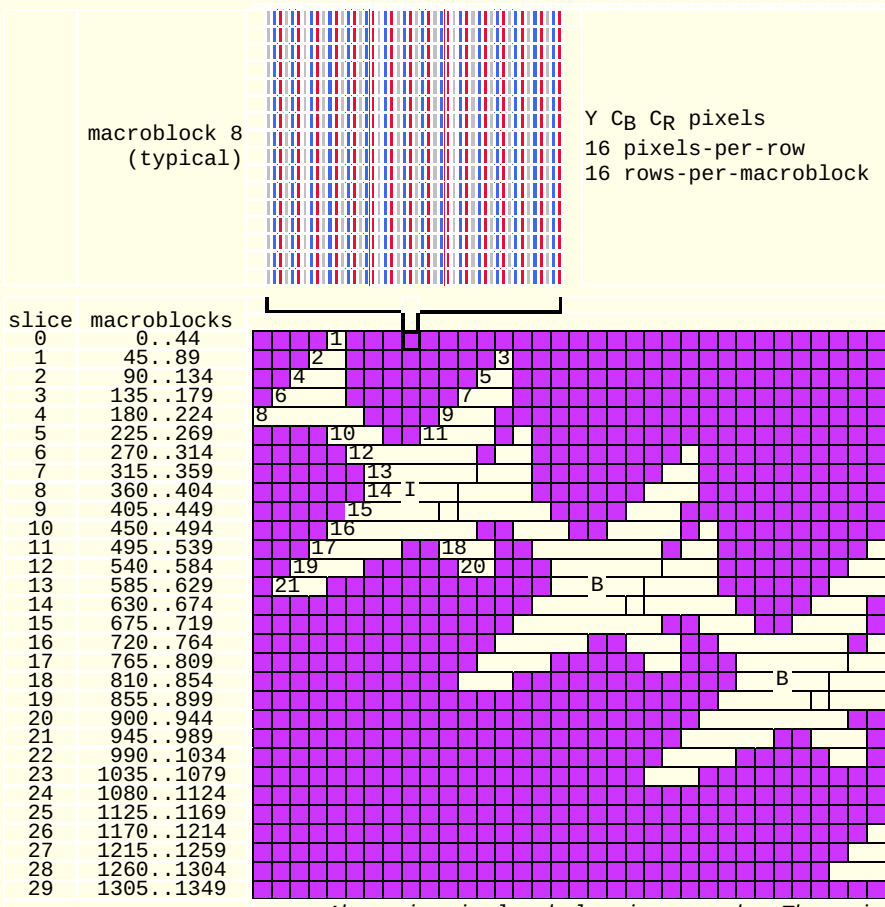
MPEG-2 Streams — *visualizing the physical properties of 234 milliseconds (7 frames) of an 'NTSC' video for example.*

There are more structures and headers than are shown but what's shown is what matters most. The start codes shown have been extracted from actual streams. In the lower part of the page, the pictures are encoded, muxed with audio and subtitles, and carried by a packetized elementary stream (PES). The illustrations show where and when the various headers are added.



The image above is meant to illustrate a composite of 4 consecutive pictures that are displayed I-B-B-P. The composite image supposes that each picture contains 21 slices that comprise the image of a bird in flight. The 21 slices comprising the bird image defines a so-called *region of interest* (ROI). There can be many ROIs. ROIs must have the same macroblock size as the picture for which they are destined. The slices that comprise an individual ROI may but usually won't align with the origin picture's macroblocks. ROIs can overlap. An ROI can encompass an entire origin picture.

As a practical matter, ROIs are first identified as macroblocks within a *destination* picture, then an optical match is sought within other pictures. If a match is found, the picture containing the match is the *origin* picture. The match will have an origin-to-destination x- y-pixel offset that must align with the destination's macroblocks but will generally not align with the origin's macroblocks. That x- y-pixel offset is the slice's *motion vector*.

For each slice in an ROI, an MPEG-2 encoder records the destination's left-most macroblock number, plus the slice's motion vector, plus a discrete cosine transform (DCT) array of the slice's pixel color fixups.

When it renders the destination picture, an MPEG-2 decoder *references* the origin picture, which it has already rendered, extracts the ROI's slices, copies the ROI slices into the destination picture, and applies the color fixups.

MPEG-2 has the following rules regarding origin pictures:

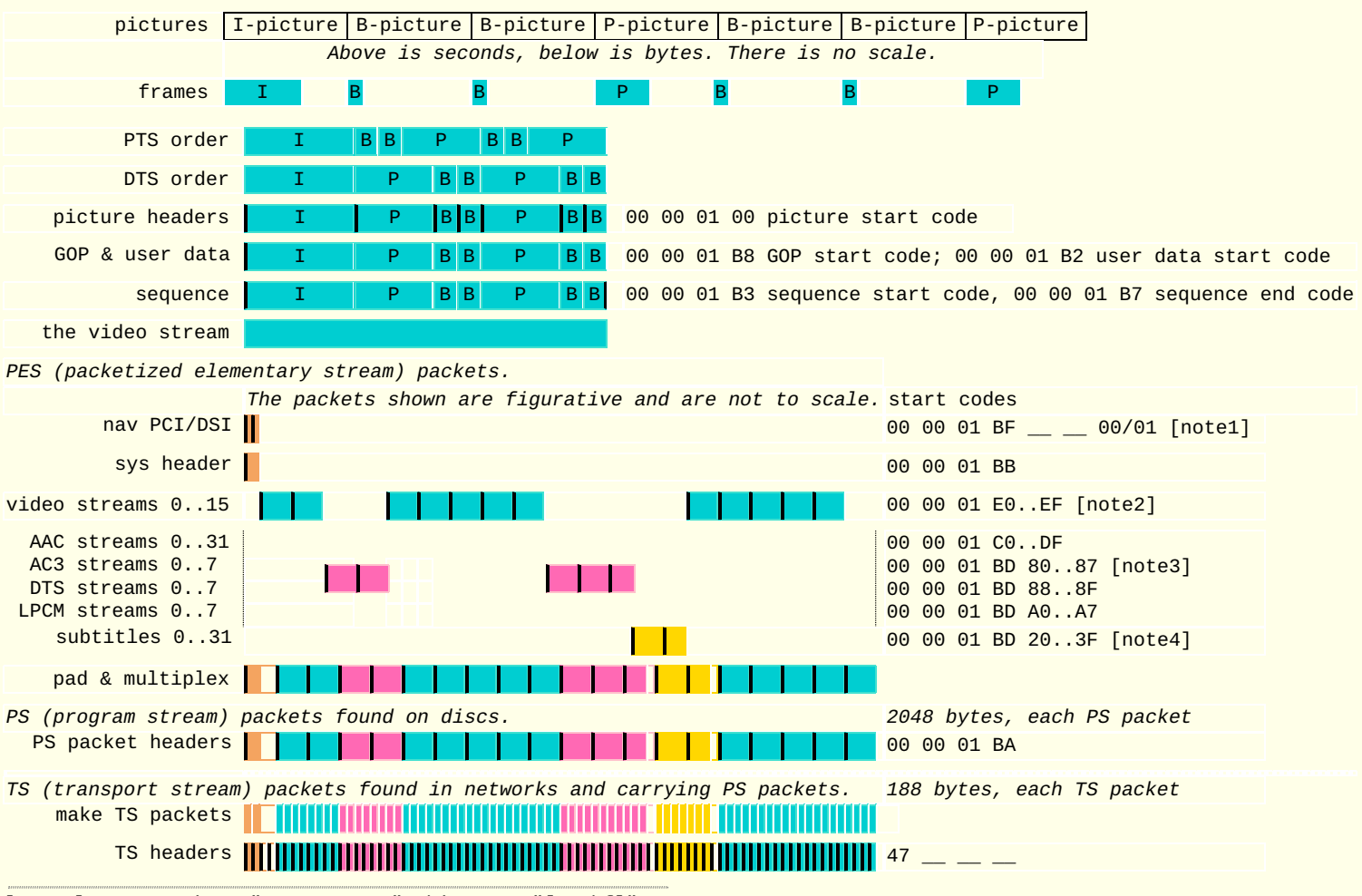
- An I-picture contains only native macroblocks and-or I-slices. I-slices are slices derived from ROIs solely from within the destination picture, an image of links in a chain link fence for example. I-slices are also called *intra-slices*.
- A P-picture contains native macroblocks, and-or I-slices, and-or P-slices. P-slices are slices derived from ROIs within a past I- or P-picture. P-slices are also called *inter-slices*.
- A B-picture contains native macroblocks, and-or I-slices, and-or P-slices, and-or B-slices. B-slices are slices derived from ROIs either from within a future I- or P-picture or that are interpolated from two origins: a P-slice and a future I- or P-picture. Like P-slices, B-slices are also called *inter-slices*

Thusly, I-pictures are aggregations of native macroblocks and-or I-slices, P-pictures are aggregations of native macroblocks and-or I-slices and-or P-slices, and B-pictures are aggregations of native macroblocks and-or I-slices and-or P-slices and-or B-slices.

In MPEG-2,

- all P-slices in an individual destination picture must derive from a single origin picture, and
- all B-slices in an individual destination picture must derive from a single origin picture or from a single origin pair.

- A frame containing a coded I-picture is called an *I-frame*.
- A frame containing a coded P-picture is called a *P-frame*, though technically, the P-frame MPEG tag can be applied even to P-frames that contain *only* I-slices.
- A frame containing a coded B-picture is called a *B-frame*, though technically, the B-frame MPEG tag can be applied even to B-frames that contain *only* I-slices or *only* P-slices.
- A field containing a coded I-scan or coded I-half-picture is called an *I-field*.
- A field containing a coded P-scan or coded P-half-picture is called a *P-field*.
- in MPEG-2, the first field must be an I-field, but the second field can be an I-field or it can be a P-field that references the first field.



[note1] FFmpeg cites "00 00 01 BF" this way: "[0x1bf]".

[note2] FFmpeg cites "00 00 01 E0" this way: "[0x1e0]".

[note3] FFmpeg cites "00 00 01 BD 80" this way: "[0x80]".

[note4] FFmpeg cites "00 00 01 BD 20" this way: "[0x20]".