

IEEE 1722 Control Format Generic Image Sensor Transport

2020-07-22, Rev 3

Edited by Yong Kim

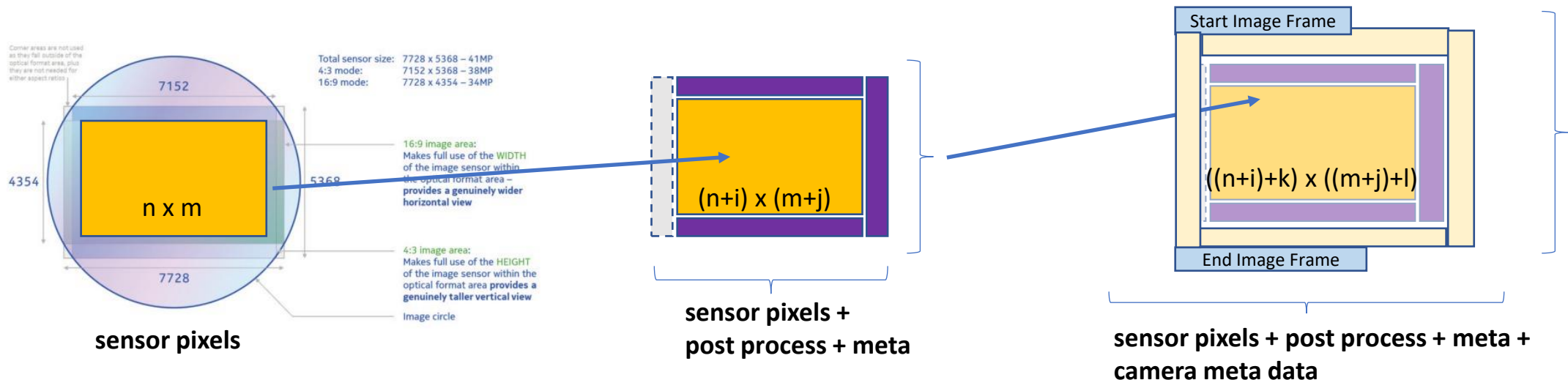
yongkim at axonne dot com

Motivation

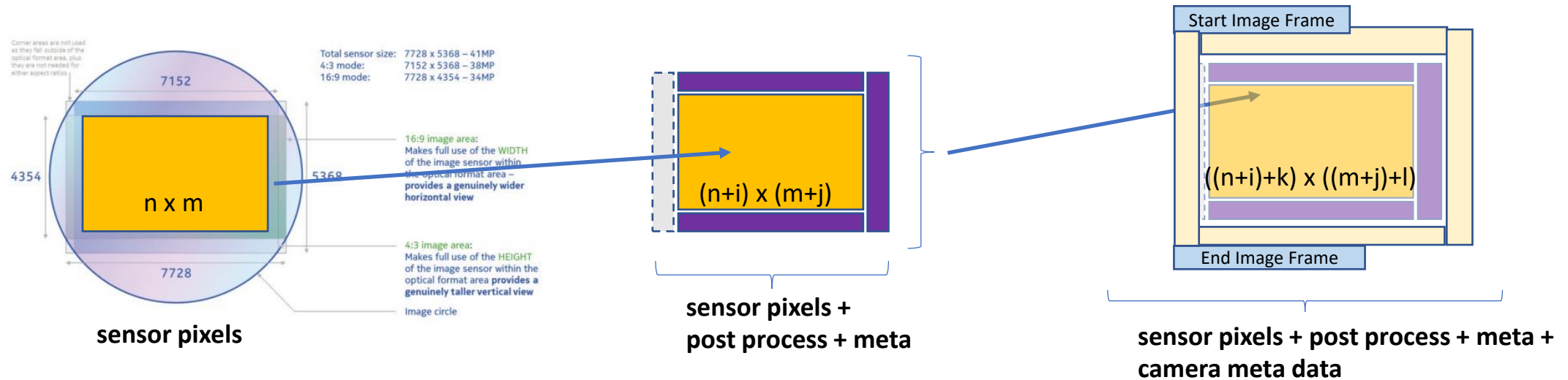
- One of the major justifications of automotive Multi-gig Ethernet PHY (IEEE 802.3ch) is network connection of high bandwidth autonomous drive (AD)/advanced driver assistance system (ADAS) sensors, such as uncompressed AD and parking cameras.
- Image sensors outputs $n \times m$ matrix of n pixels per line \times m lines.
- Image sensors meta-data outputs, often, are embedded in the extra row or columns.
- There is a need to transport image sensor outputs, row by row, over high-speed serial interface; P2P, or ideally real P2MP network, transport protocol that supports time sensitive networking, i.e. IEEE 1722 AVTP.
- Avoid pre-standard and proprietary implementation islands that form once system productions start; Avoid artificial turf defense when these form. There is no intrinsic value to having multiple different but functionally equivalent approaches.
- Avoid compatibility issues, implement transparent transport, i.e. bits in, bits out, layered.
- Stated goal of 1722b – serve the industry and perform necessary revision quickly – fits well with a goal of this proposal.

Cameras used in ADAS and Autonomous Drive

- 1 MP, 1.3MP, 2MP, and 8 MP, often leveraged from consumer electronics R&D, e.g. cell phones.
 - The same image sensors but instead of RGBG (Red, Green, Blue, Green) filter lens, it may use alternate color filters, RCCC (Red, Clear, Clear, Clear), or other combinations that are optimized for machine vision and object detection.
 - Camera sensor wrapped around camera system SoC may outputs to a number of standard interfaces.
 - Many of the ADAS/AD camera system leverages the mobile camera (integrated Sensor + system SoC) R&D.
- Transport Considerations
 - Camera sensors output what it sees in various configurable formats
 - Camera system SoC output what it processes in various configurable formats
 - Transport should just transport bits, bytes. In camera terms, [frame start] [lines]...[lines][frame end].



1722b to transport image data - image sizes

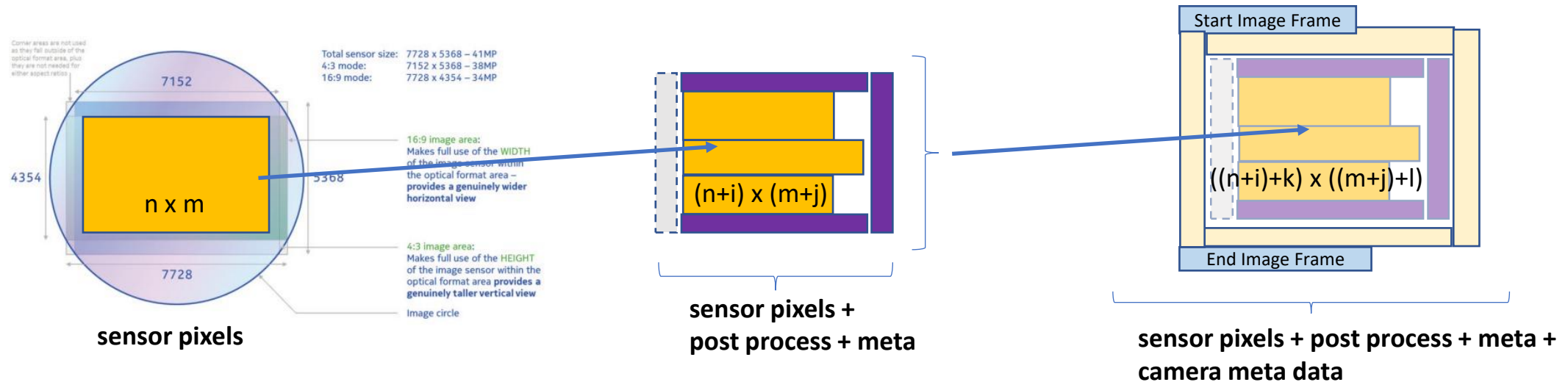


- Image sensor + lens + filter
- Just accept $n \times m$ pixels of some resolution
- No need to know, don't care about content (from transport perspective)

- Pixels has meaning (color space, depth, HDR processed, etc. whatever).
- Added pixel space may be added for meta data that increases $n \times m$ to be a bit bigger.
- No need to know, don't care about content (from transport perspective)

- Transparent transport of the previous content.
 - Start Image Frame
 - Line by line
 - End Image Frame.
 - May have added metadata but not distinguishable.

1722b to transport image data - discuss



- Image sensor or lens filter
- Just accept $n \times m$ pixels of some resolution
- No need to know, don't care about content (from transport perspective)

- Each image is on its own.
- Frame, Line, .. Line, Frame
- Don't assume fixed image size/depth.
- Each image type is separate stream (e.g allow for interspersed raw and HDR images, or dual image sensors onto single transport.)

- Each frame is on its own.
- FS, 'embedded', ... , line, ..., embedded', FE
- Don't assume fixed image size/depth.
- Transparent transport of packet header – issues?

1722b to transport image data – requirements

Background

- Video Stream format is optimized for rendering device, e.g. TV
 - Presentation Time → renderer's use.
- Example: Brake actuator needs presentation time – currently in time-sensitive header.
- ACF format allows for input data timestamp, e.g. Lin message time stamp, camera shutter.
 - ACF time → sensor input's use.
- Example: Image sensor needs input time – currently only in ACF

Some Requirements for potential 1722 transport.

- Format with input time (e.g. as in existing ACF sub-type message timestamp)
- Raw video format that supports variable length lines (i.e multiple rectangles in a single frame).

1722b to transport image data – considerations

- Raw image : e.g. not i.e., 4:4:4, 4:4:4:4, ...,
- ‘Improved image’: e.g. 8b(4:2:2), 10b(4:4:2), 12b(4:4:4), 16b(5:6:5)...
- Resolutions: e.g. 1MP, 1.3MP, 2MP, 4MP, 8MP,...

- Line Lengths:

pixel/line	+8	12 bit/pixel in bytes	16 bit/pixel in bytes
1280	1288	1932	2576
1920	1928	2892	3856
2560	2568	3852	5136
3840	3848	5772	7696

- Should allow for transparent transport of a line of pixel bytes in multiple Ethernet frames and reduce max latency for other TSN flows w/o use of preemption.
- Be aware of robustness of CRC32 and payload – avoid use of jumbo.
- For those who loves linear math and error functions, see referenced in: http://www.ieee802.org/3/bj/public/jul12/cideciyan_01_0712.pdf