Negotiation of Generic Image Attributes in SDP

draft-johansson-mmusic-image-attributes-01

ERICSSON
TAKING YOU FORWARD

- Problem statement
- Objective of draft
- Existing solutions
- Observations/issues sofar
- WG item ?
- Current solution (for those interested)

What is the problem ?

- Only a limited set of picture sizes specified in standards/payload formats
  - Coarse granularity
    - (SQCIF, QCIF, QVGA, CIF, 4CIF...)
  - Rescaling needed in the receiver to fit the video image into a (portion of) the screen.
    - Introduces distortion (blurring)
    - Increases complexity
    - Upscaling → Unnecessary low quality
    - Downscaling → Waste of bitrate, over allocation of memory
- LS from 3GPP to MMUSIC
  - https://datatracker.ietf.org/documents/LIAISON/file541.doc

Objective of this draft

- Make it possible to negotiate a desired image size on the receiver display
  - Lower need to allocate abundant memory
  - Reduce/remove the need to rescale the image
    - Optimum quality for given bitrate / image size can be achieved
    - Less image distortion (e.g blurring)
- Generic, should (preferably) not be codec dependent

http://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_47/Docs/S4-080009.zip

Objective of this draft

- Interoperate well with related parameters in e.g payload formats
- Support:
  - Asymmetric setup, a very likely scenario.
  - SAR (Sample Aspect Ratio) → make it possible to compensate for other than 1:1 sample aspect ratios on the receiver display (optional)
  - Frame rate ?
- An outlined solution exist

Already existing...

- sprop-parameter-sets (H.264)
  - Offer (Alice) signals e.g the image size for the media from Alice to Bob, i.e not what Alice wish to receive.
    - Contrary to ”normal” offer/answer
  - Bob is assumed to handle this → sprop-parameter-sets cannot be used to avoid rescaling of image
  - Specification of image size, not negotiation
  - Subject to change ? (RFC3984bis)
- CUSTOM x,y,MPI (H.263)
  - specified in fmtp line along with other picture sizes that the offerer wish to receive with a given max frame rate
  - `a=fmtp:xx CUSTOM=640,480,2;CIF=1;QCIF=1`
  - answerer SHALL NOT modify any parameters

Some observations/issues

- Draft lacks a proper requirement specification
- Interaction with other parameters must be clarified
  - Relation to e.g profile-level-id or max resolution
  - Interaction with `a=framerate`, `a=quality`, `a=orientation`
  - frame rate range → may be better to remove it from draft esp. as the formulation does not map well to e.g MB/s in H.264.
  - Possible conflict with sprop-parameter-sets → (Selectively) ignore sprop-parameter-sets if imageattr is used ?
- Rescaling or cropping on sender side
  - Encoders are not mandated to rescale image → Assume that offer of video fixes this issue?

Some observations/issues

- Asymmetry introduces problems..
  - Separate RTP streams 2nd offer/answer
  - Solution: Bake send and receive directions into the same a=imageattr ? (Randell Jesup)
- SAR references to H.264 spec
- Is video the only application ?
  - Whiteboard presentation with optimized quality?
- ABNF syntax needs to be corrected...
- Only point-to-point considered
- Any better name for the attribute (img, image) ?

Working group item ?

- Author would like to propose this as a working group item
  - 3GPP SA4 wants to use something like this as hinted by the LS
  - The identified issues are not any show-stoppers IMHO

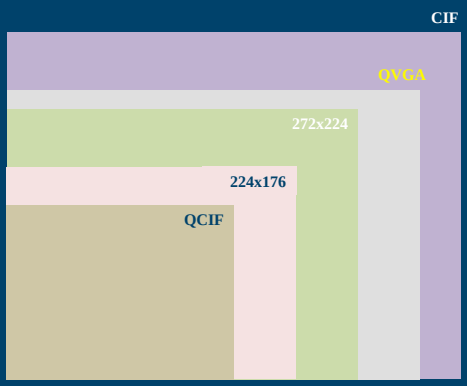
Solution outline

An outline of the latest draft version for the interested reader.

ERICSSON
TAKING YOU FORWARD

Different image sizes

- Offer the image sizes in the figure
  - Equal preference unless specified (q parameter)
- Answerer picks the desired image size



Offer:

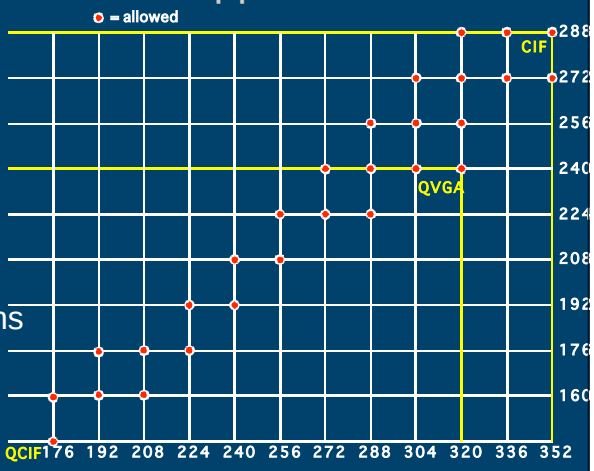
```
a=imageattr:97 [x=352,y=288] [x=320,y=240] [x=272,y=224] \
   [x=224,y=176] [x=176,y=144]
```

Answer:

```
a=imageattr:97 [x=272,y=224]
```


Different image sizes with supported PAR

- Support ranges with limitation in possible picture aspect ratios (PAR)
- Eliminates problem with odd combinations



Offer:

```
a=imageattr:97 [par=[1.1,1.3],x=[176:16:352],y=[144:16:288]]
```

SAR

- SAR = Sample Aspect Ratio
  - Makes it possible for encoder to compensate for the receiver display sample aspect ratio
  - Reduces/eliminates aspect ratio distorsion
  - Reference to table E1 in H.264 standard (is this a good idea?)

```
a=imageattr:97 sar=[1:16] [x=800,y=640] [x=480,y=320]
```


Frame rate

- Makes it possible to define a supported frame rate range for the given image size range
- Quesionable if frame rate should be specfied in this draft

```
a=imageattr:97 [fr=[5,15],x=800,y=640] [fr=[10,25],x=480,y=320]
```

Support for asymmetry

- Very likely that two endpoints desire different image size.
- Asymmetry supported by means of incomplete formulation
- Introduces need for a 2nd offer/answer
- Likely to change...

```
1st offer (Alice to Bob):
m=video 12340 RTP/AVP 97
a=rtpmap:97 H264/90000
a=imageattr:97 [x=[480:16:800],y=[320:16:640]] /
   [x=[176:8:208],y=[144:8:176]]
a=sendrecv
m=video 12342 RTP/AVP 98
a=rtpmap:98 H264/90000
a=imageattr:98 *
a=recvonly
```

This is the mage sizes I support in A →B direction

Tell me.. What do you support in B→A direction

Support for asymmetry (cont..)

```
1st answer (Bob to Alice:
m=video 12340 RTP/AVP 97
a=rtpmap:97 H264/90000
a=imageattr:97 [x=480,y=320]
a=recvonly
m=video 12342 RTP/AVP 98
a=rtpmap:98 H264/90000
a=imageattr:98 sar=[1:16] [x=[480:16:800],y=[
       [x=[176:8:208],y=[144:8:176]]
a=sendonly

2nd offer (Alice to Bob):
m=video 12340 RTP/AVP 97
a=rtpmap:97 H264/90000
a=imageattr:97 [x=480,y=320]
a=sendonly
m=video 12342 RTP/AVP 98
a=rtpmap:98 H264/90000
a=imageattr:98 sar=2 [x=800,y=640]
a=recvonly
```

I want 480x320 image size on my display

I support the following in the direction B→A

I want 800x640 image size on my display and compensation for sample aspect ratio 12:11