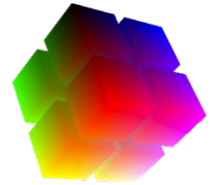
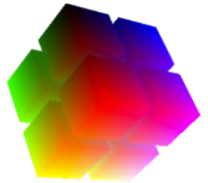
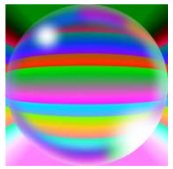
HEIF 解説
～コンテナ構造の実際～

2018/01/25(Thu) “よや” yoya@awm.jp



自己紹介 (@yoya)

- ImageFlux 開発の手伝いをしてます
  - 画像バイナリ調査とプロトタイプ作りがメイン業務
    - (C言語こわい。。Golang 怖くない。でも cgo チョットコワイ。。。)
  - 今は ImageFlux - HEIF 対応の調査研究中 (今日はその情報共有)
- 趣味は ImageMagick の追っかけ
  - http://d.hatena.ne.jp/yoya/searchdiary?word=ImageMagick

目次

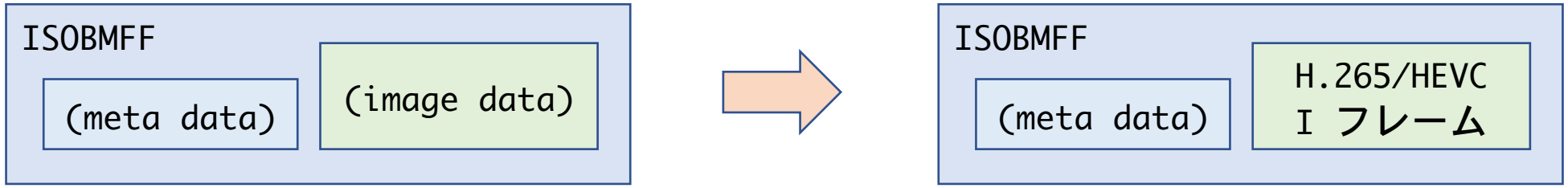
- HEIF とは
- HEIF 利用環境
- HEIF 画像サンプル
- HEIF 関連ツール、ライブラリ
- HEIF のコンテナ構造
- HEIF 最近の話題

参考リンク

- Nokia Technologies
  - http://nokiatech.github.io/heif/index.html
  - https://github.com/nokiatech/heif
- MPEG
  - https://mpeg.chiariglione.org/standards/mpeg-h/image-file-format
- Apple
  - https://developer.apple.com/videos/play/wwdc2017/513/
  - https://developer.apple.com/videos/play/wwdc2017/511/
  - https://developer.apple.com/videos/play/wwdc2017/507/

HEIF (ヒーフ)とは？

- HEIF は画像を格納するコンテナ仕様
  - JPEG2000 の後継
    - JPEG2000の情報要素を流用するので、知ってると楽
    - JPEG2000 のコンテナ構造は **ISOBMFF** (QuickTime 形式の後継)
  - 主に、画像圧縮コーデックは **H.265/HEVC** Intra フレーム
    - HEVC を入れる事を示す為に、拡張子は .heic
    - 一応、任意の圧縮形式を入れられる(はず)



HEIF を盲目的に褒めてみる

- HEIF は **ISOBMFF** による柔軟性と **H.265/HEVC** の高圧縮性能とダイナミックレンジを持ち合わせ、既存の全て画像フォーマットに取って代わるポテンシャルを期待される
  - JPEG より圧縮が効く 。同じ位の画質で半分のサイズになると言われる。
  - PNG と同じく可逆圧縮(Lossless)できる
  - GIF と同じくアニメーションできる
  - TIFF と同じく複数の画像をメタデータ付きで保存できる
  - RAW/DNG 程ではなくとも、必要十分な色域 (HDR 的な意味で)

HEIF を盲目的には褒めない

- HEIF は ISOBMFF による柔軟性と H.265/HEVC の高圧縮性能とダイナミックレンジを＜略＞？？？
  - JPEG より圧縮処理が圧倒的に重たいよね。。
  - PNG と違って YUV かつ 420 で保存するので(現状では)劣化する
  - GIF と同じくアニメーションできるけどブラウザでさくさくインライン表示できなきゃ意味ない。(言いがかり気味…)
  - macOS, iOS 共に今のところ YUV420 で 8bit しか表示できない？？

HEIF FAQ的な (1/2)

- HEIF の拡張子は？
  - 中に H265/HEVC を入れる場合は .heic 。アニメーションは .heics
    - H.264/AVC だと .avci / .avcs（何故かこれだけ s をつけても4文字）
    - 任意のコードで .heif / .heifs
- HEIF の読みは？
  - Apple の WWDC2017 で発音していた「ヒーフ」が定着しそうです
    - 開発元の Nokia テクノロジーは「へイフ」と呼んで欲しいらしいが。。
- HEIF の開発者は？
  - Nokia Technologies の emre.aksu 氏。今は research leader
    - 何人も該当者いるけど、うち１人挙げるなら多分この人
    - ISO 提案ドラフト authorの一人で参照実装(nokiatech/heif)のプログラマ

HEIF FAQ的な (2/2)

- 画質指定(JPEG でいう quality)は？
  - x265 だと crf パタメータで調整できる。(他にも色々ある)
- Lossless(可逆圧縮) もできる？
  - HEVC ができるので出来る
    - 。。と言いたいけど RGB⇔YUV 変換するので完全には無理
    - あと YUV444 対応してるかどうかも問題 (色の情報が落ちるかも)
- 透明度も持てる？
  - Auxillary role で対応
  - 同じ要領で、depthマップも持てる
- アニメーションできる？
  - できるはず。でも未調査。m(_ _)m

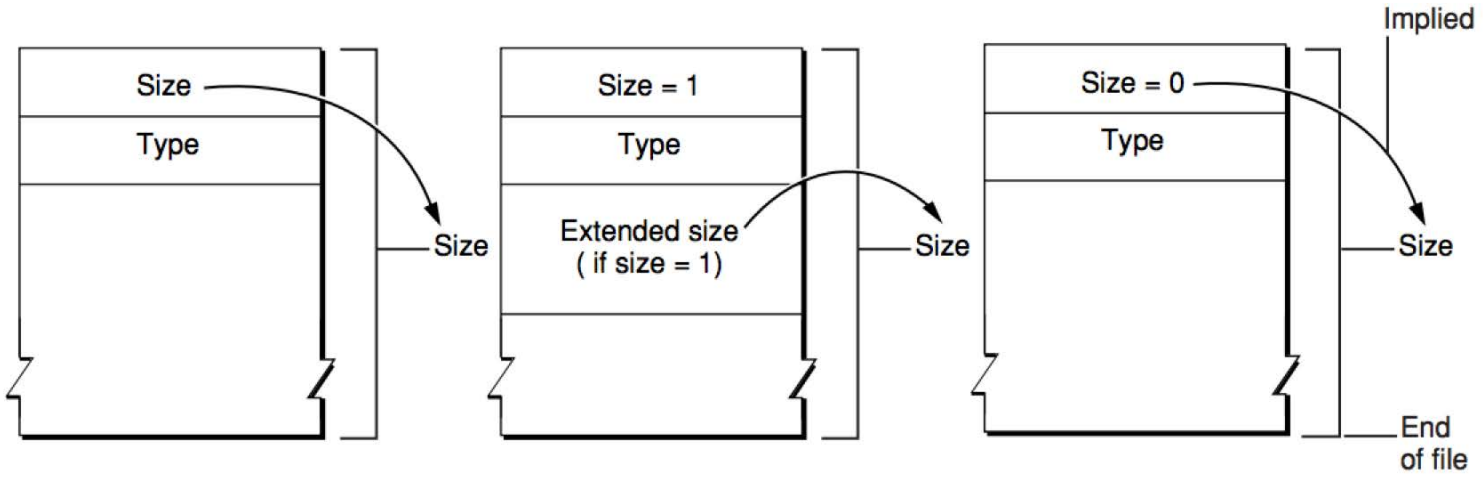
HEIF の基本

- 画像の Role(役割)に名前がついている
  - HEIF は複数の画像や補助イメージを入れられるので (いわゆる MiAF)

Roles	説明
Primary	代表する画像
Master	フル解像度
Thumbnail	解像度小。サムネ
Auxillary	透明度イメージ、深度マップ
Hidden	表示しない
Derived	グリッドやオーバーレイ加工など
Equivalent	その他のイメージ

ISOBMFF について

- MPEG の規定するコンテナ(バイナリ)形式
  - https://en.wikipedia.org/wiki/ISO_base_media_file_format
- QuickTime の後継なので、こちらで勉強しても良い
  - https://developer.apple.com/standards/qtff-2001.pdf

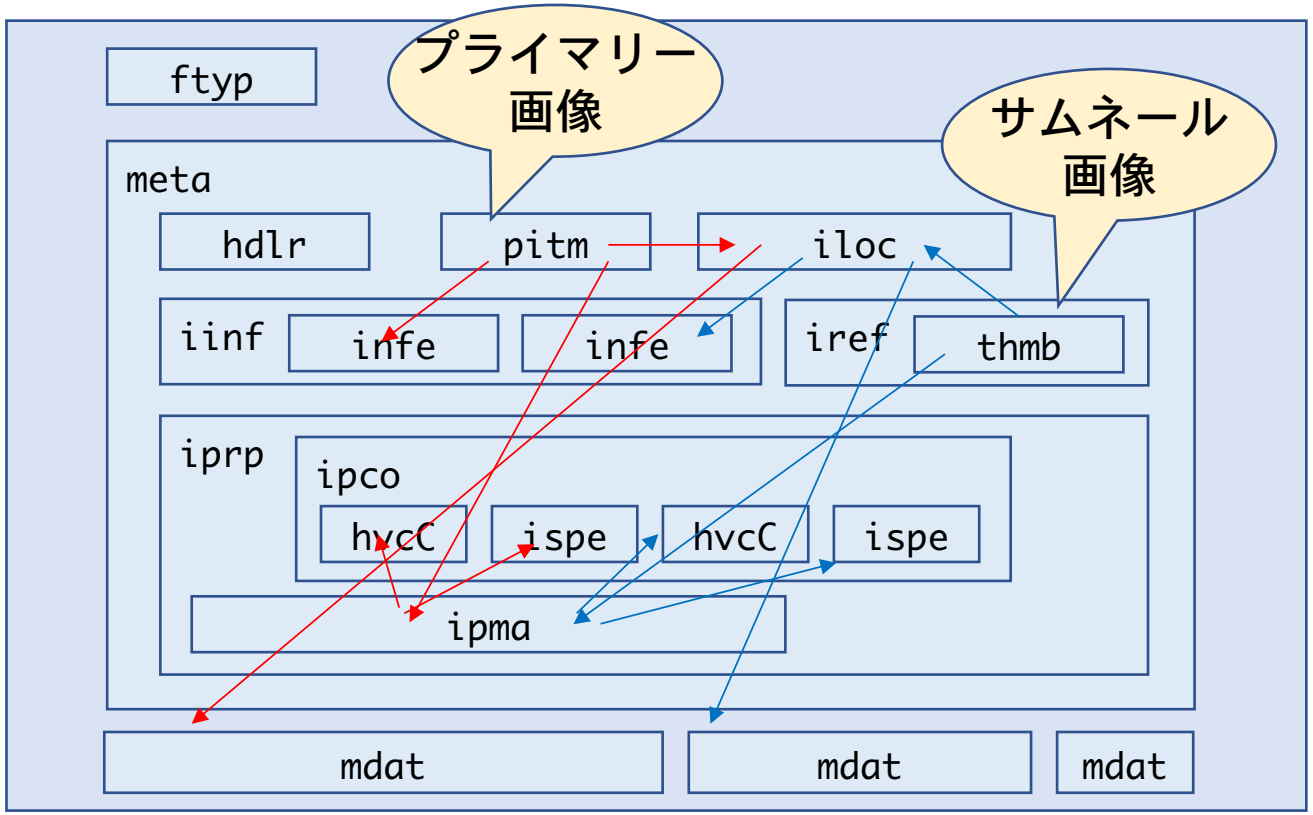


ISOBMFF の柔軟性 (1/3)

- ISOBMFF (ISO base media file format)

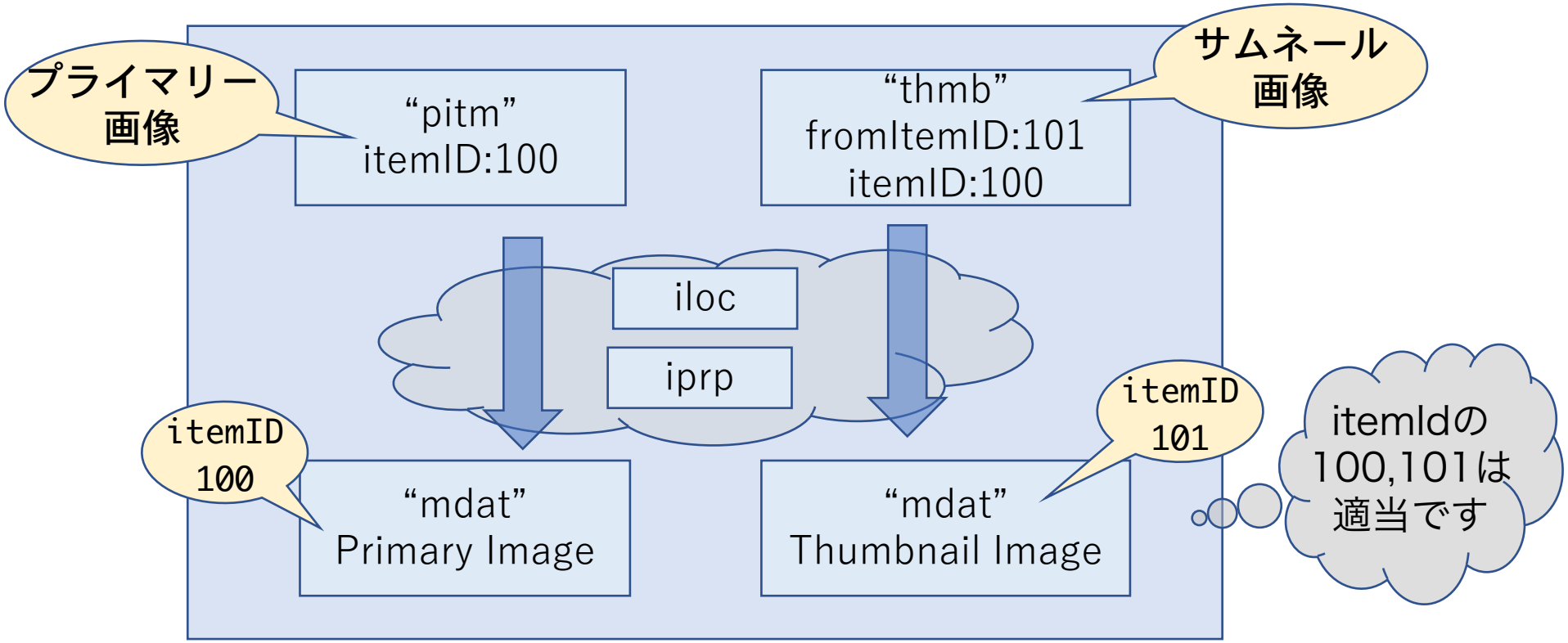
nokiatech/heif/images/
autumn_1440x960.heic
のコンテナ構造
(詳しくは後で)

柔軟というか
複雑というか



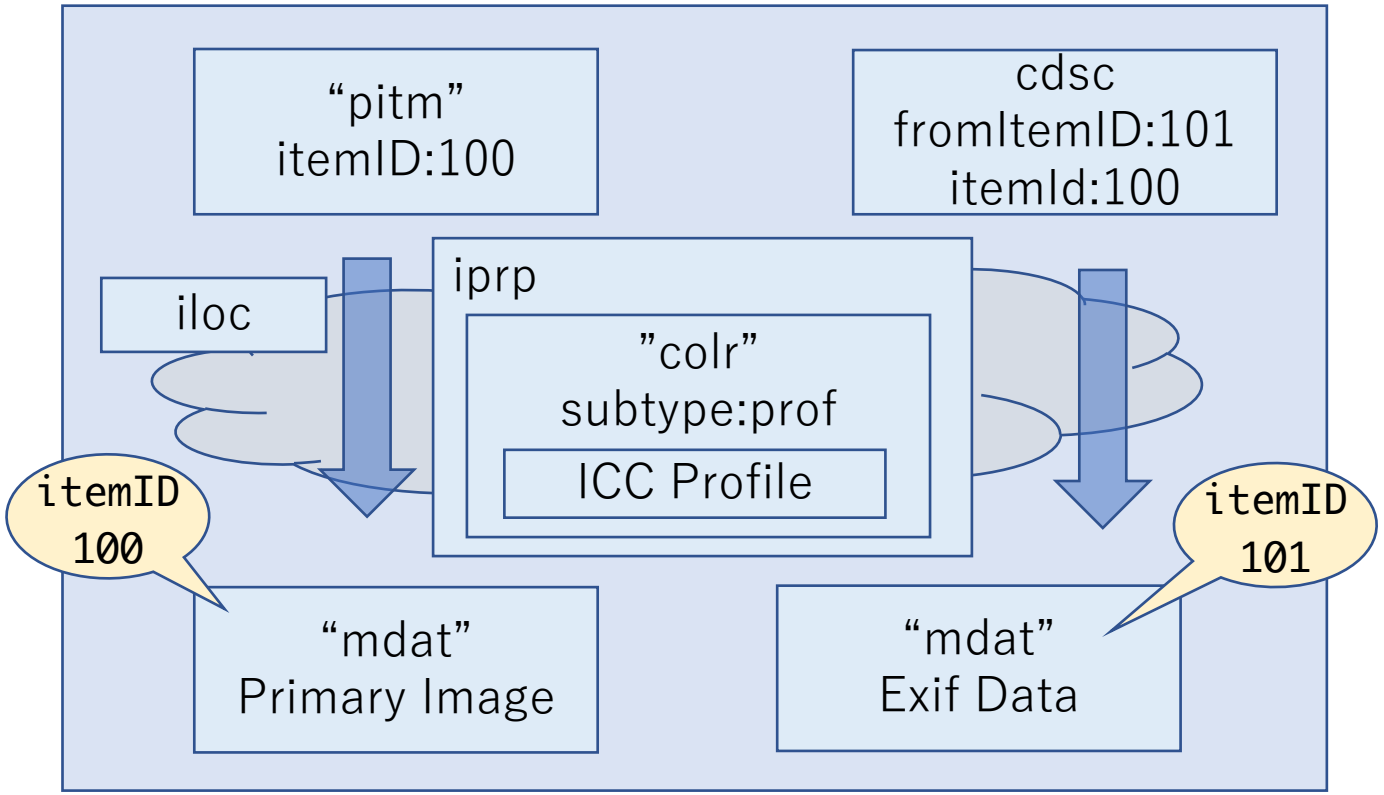
ISOBMFF の柔軟性 (2/3)

- サムネール画像つき



ISOBMFF の柔軟性 (3/3)

- メタデータ(ICC プロファイル、Exifデータ)

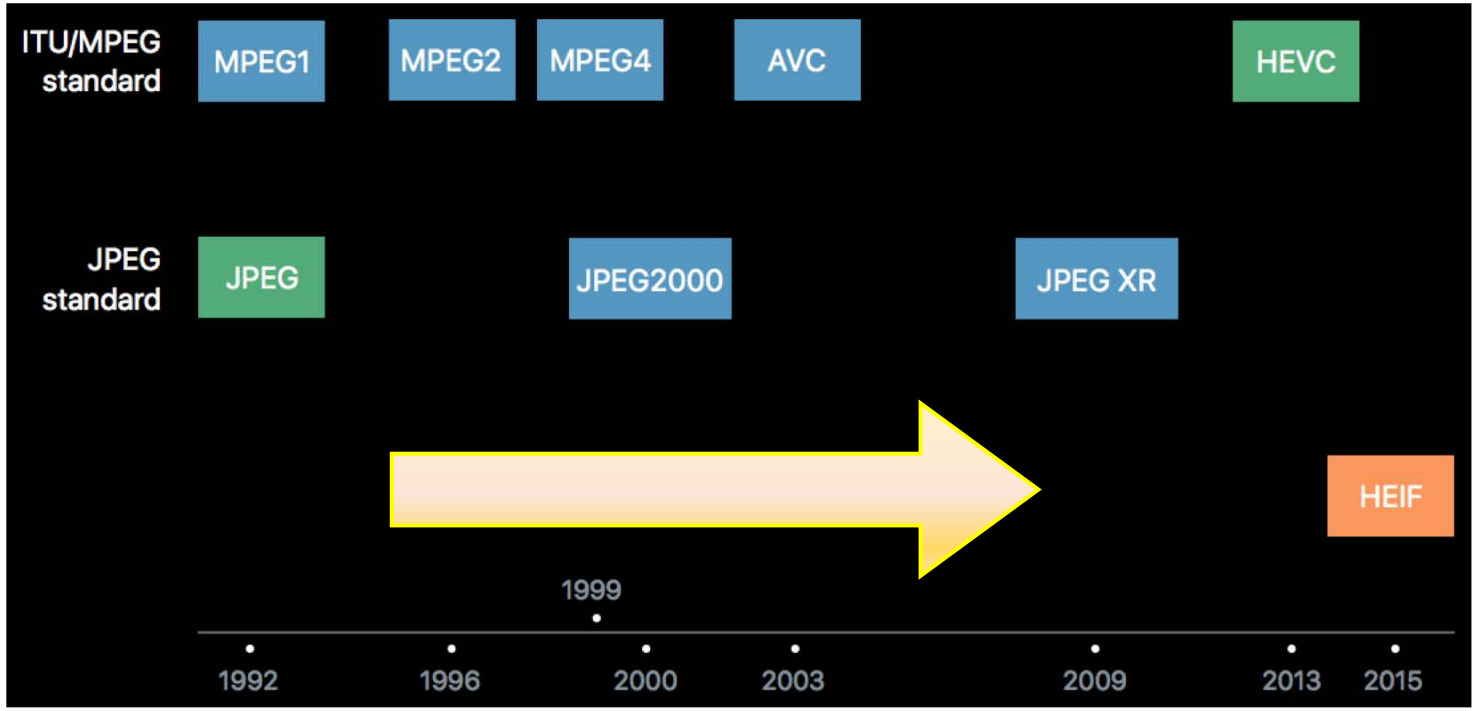


ISOBMFF の柔軟性（？？？）

- Box ごとに色んな仕様書を漁らないと分からないので、昔から MPEG を追ってない人には辛いかも。
- ptim, infe, ipma, => ISO/IEC 14496-12 (ISOBMFF)
- iloc => T.807 (JPEG2000)
- elst, csmi, mint, Exif, hvcC => ISO/IEC CD 23008-12

H.265/HEVC の高圧縮性

- MPEG1 から H.265/HEVC まで、格段と圧縮性能が上がってる
  ただし、圧縮サイズを半分にする為に処理負荷が10倍になったりする



© https://developer.apple.com/videos/play/wwdc2017/513/

H.265/HEVC I **フレーム**の圧縮アルゴリズム

- Angular予測フィルタ (エッジの向きを33方向で選択)

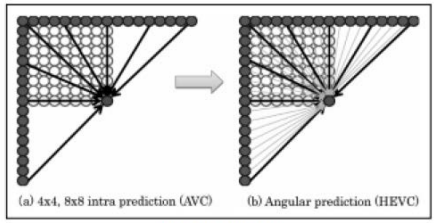
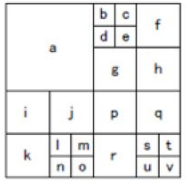


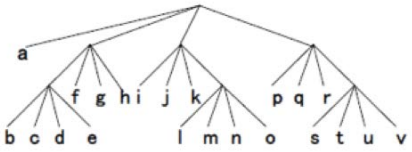
図2 Angular予測

© https://www.jstage.jst.go.jp/article/itej/67/7/67_537/_pdf

- ブロック分割
  - 画像のエリア別に圧縮パラメータを変える
  - JPEG はこれが出来ないのが大きな弱点



- 整数DCT (JPEG は 実数計算)
  - なお、整数DST 、skip(直交変換なし)も選択できる



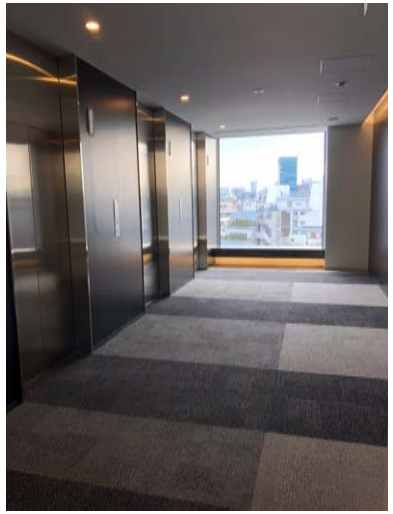
© http://independent.academia.edu/ShevachRiabtsev

- 算術符号 CABAC (ハフマン符号圧縮より強い)
  - コンテキスト適用処理

H.265/HEVCのHDR対応 (1/5)

- HDR (High Dynamic Range)
  - 主に露出調整と色域が絡む
- 暗い場所と明るい場所を同時に撮りたい
  - ＞色深度8ビットじゃ足りない

明るい場所の階調が潰れる



暗い場所の階調が潰れる

H.265/HEVCのHDR対応 (2/5)

- Deep Color
  - 既存の画像ファイルは8bit 深度が殆ど
    - 2^(8*3) = 約1678万色。いわゆる True Color
  - 10bit,16bit 等、それ以上が Deep Color
- HEVC は 10bit, 12bit に対応
  - 10bit は 2^(10*3)=約10億色
  - 12bit は 2 ^(12*3)=約687億色
- ただしプロファイル次第

プロファイル	色ビット深度	色差サンプリング
Main	8ビット	YUV420
Main 10	8～10ビット	YUV420
Main 12 (?)	8～12ビット	???

H.265/HEVCのHDR対応 (3/5)

- プロファイル種別の参考資料

Feature support in some of the video profiles[13]

Feature	Version 1		Version 2						
	Main	Main 10	Main 12	Main 4:2:2 10	Main 4:2:2 12	Main 4:4:4	Main 4:4:4 10	Main 4:4:4 12	Main 4:4:4 16 Intra
Bit depth	8	8 to 10	8 to 12	8 to 10	8 to 12	8	8 to 10	8 to 12	8 to 16
Chroma sampling formats	4:2:0	4:2:0	4:2:0	4:2:0/4:2:2	4:2:0/4:2:2	4:2:0/4:2:2/4:4:4	4:2:0/4:2:2/4:4:4	4:2:0/4:2:2/4:4:4	4:2:0/4:2:2/4:4:4
4:0:0 (Monochrome)	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
High precision weighted prediction	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chroma QP offset list	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cross-component prediction	No	No	No	No	No	Yes	Yes	Yes	Yes
Intra smoothing disabling	No	No	No	No	No	Yes	Yes	Yes	Yes
Persistent Rice adaptation	No	No	No	No	No	Yes	Yes	Yes	Yes
RDPCM implicit/explicit	No	No	No	No	No	Yes	Yes	Yes	Yes
Transform skip block sizes larger than 4x4	No	No	No	No	No	Yes	Yes	Yes	Yes
Transform skip context/rotation	No	No	No	No	No	Yes	Yes	Yes	Yes
Extended precision processing	No	No	No	No	No	No	No	No	Yes

© https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding#Profiles

■表1. 主なHEVCプロファイル

general_profile_idc	プロファイル名	内容
1	Main	4:2:0 8 bit映像符号化
2	Main 10, Main 10 Still Picture（策定中）	4:2:0 8-10 bit映像・静止画符号化（静止画は策定中）
3	Main Still Picture	4:2:0 8 bit静止画符号化
4	Format range extensions (RExt)	レンジ拡張（4:0:0/4:2:2/4:4:4、12/14/16 bit等）
5	High throughput	ハイスループット拡張
6	Multiview Main (MV-HEVC)	マルチビュー拡張
7	Scalable Main & Scalable Main 10（SHVC）	スケーラブル拡張
8	3D Main（3D-HEVC）	3D拡張
9	Screen content extensions (SCC)	スクリーンコンテンツ拡張
10	Scalable format range extensions	スケーラブルレンジ拡張

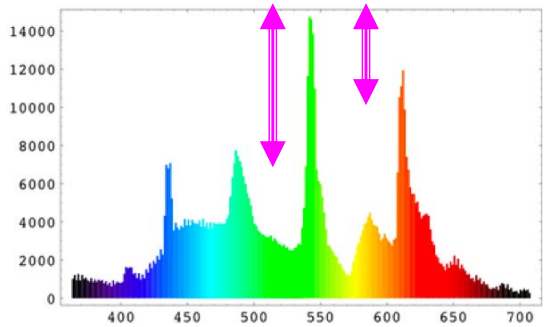
（筆者作成）

© https://www.ituaj.jp/wp-content/uploads/2017/06/2017_06-07-Spot-HEVC.pdf

H.265/HEVCのHDR対応 (4/5)

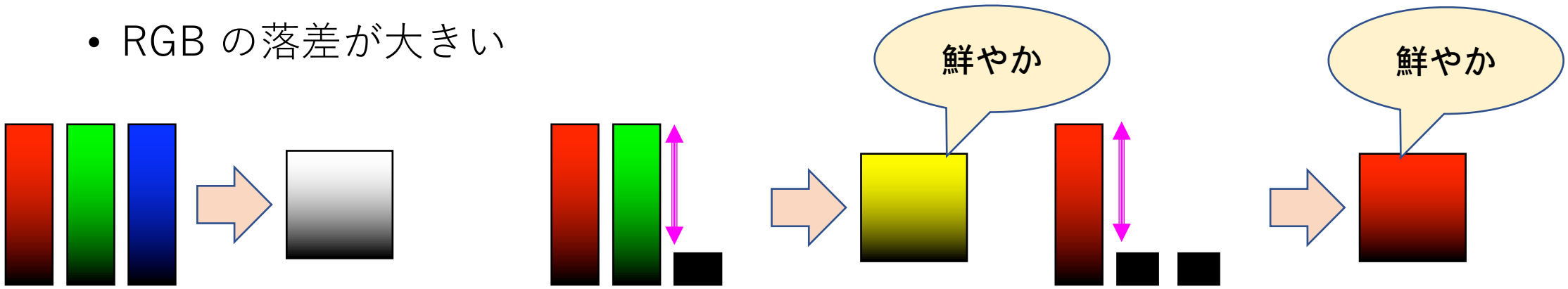
- コントラストが上がると、より鮮やかな色が表示できる

- 鮮やかな色とは？
  - スペクトルが尖ってる
    - レーザーの単色光とか



© http://hidjikken.blog.shinobi.jp/分光器の波長較正/

  - RGB の落差が大きい

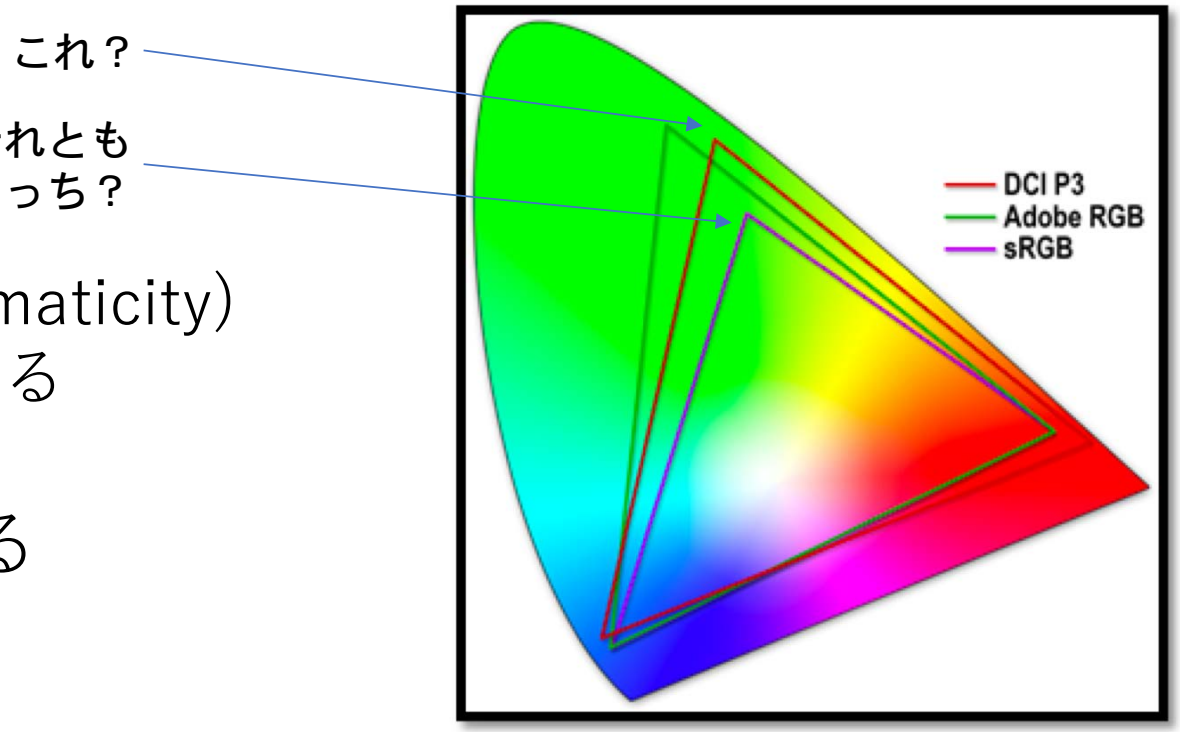


H.265/HEVCのHDR対応 (5/5)

- RGB:(0,255,0)の緑は具体的にどんな色？

- HDR 色域対応
  - sRGB より広い色表現
  - R,G,B の3刺激値 (primary chromaticity)
    右図の三角形の頂点を広げられる

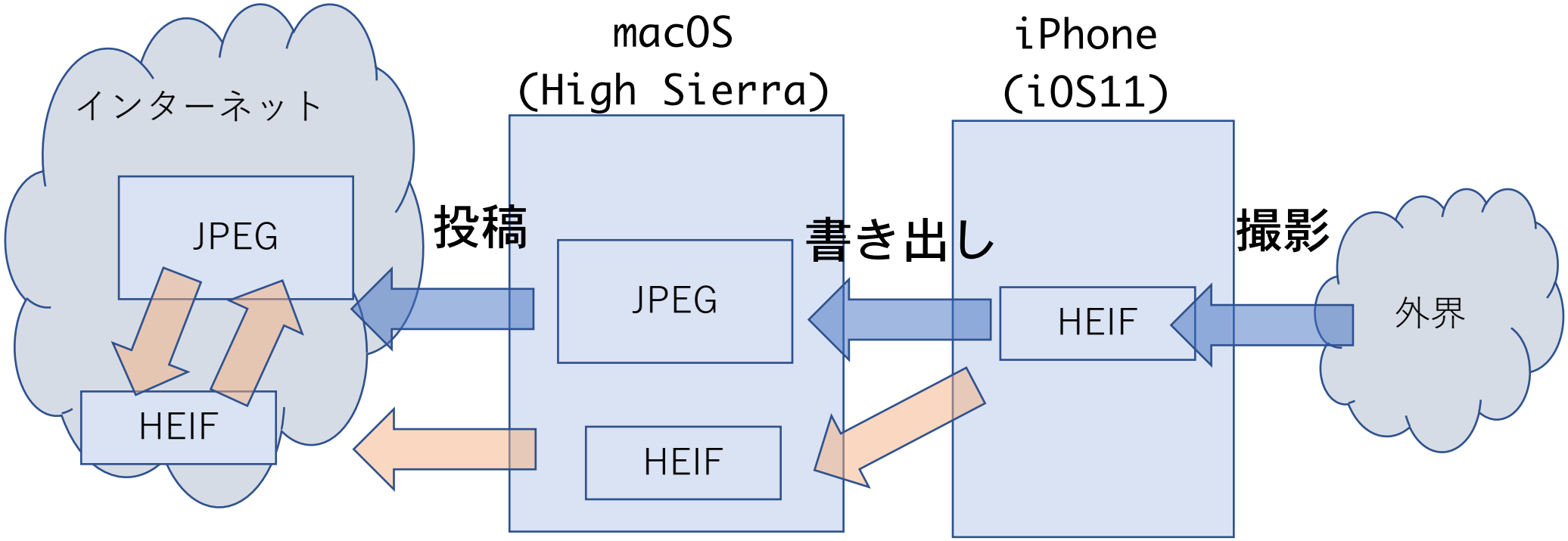
- 色域を広げると階調が荒くなる
  - > さっきの Deep Color が必要



© https://www.reddit.com/r/apple/comments/5287c3/lets_talk_about_that_new_display_in_iphone_7/#d7i93i7

HEIF 利用環境(1/3)

- 現状は青矢印、将来は赤矢印？(かもしれない)



HEIF 利用環境 (2/4)

- 今のところ Apple の macOS/iOS の最新版のみ
- 機種(に載るチップ)にもよる

iOS11
対象外

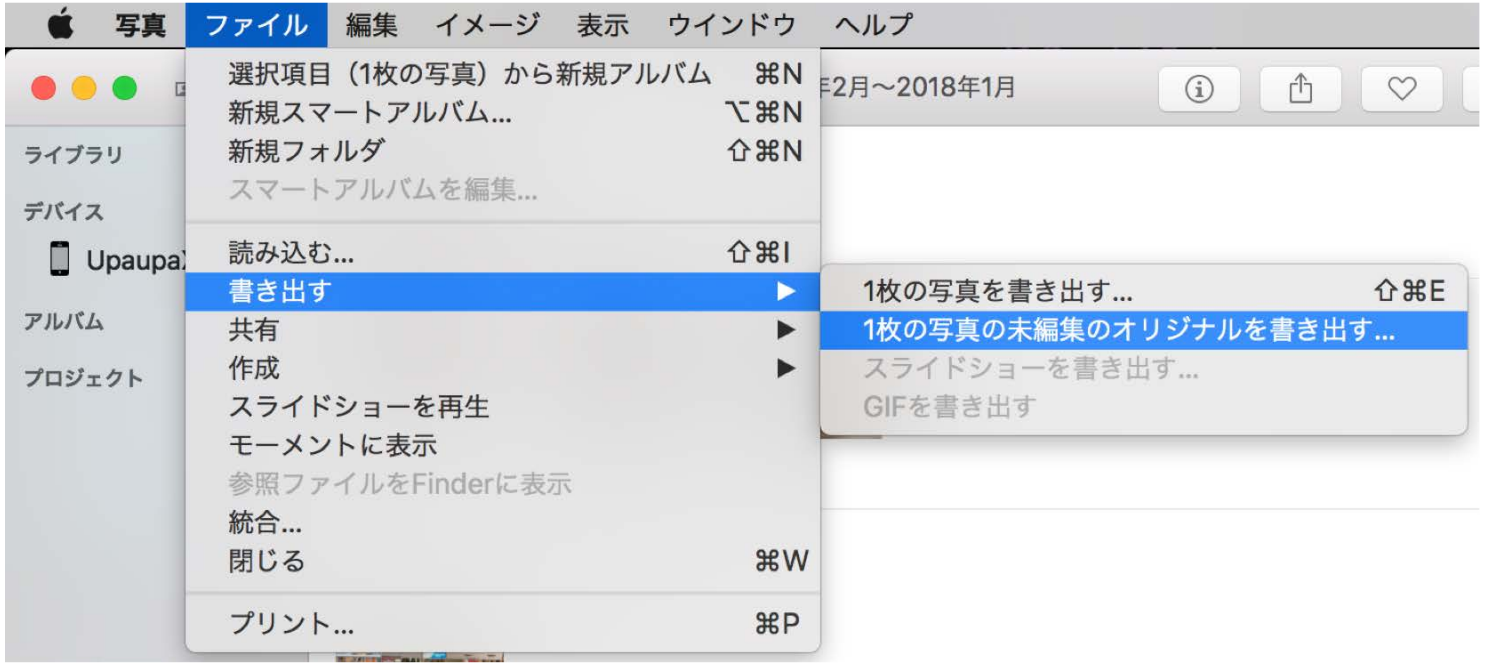
	iPhone7 以降	iPhone6,SE 以降	iPhone5s 以降	iPhone5c 以前
Encode	H/W	-	-	-
Decode	H/W	H/W	S/W	-

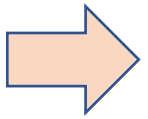
設定>カメラ>
フォーマット

- ただし、HEIF 対応アプリ(標準カメラ等)で撮影しないと駄目
- MacBook 等の写真アプリで吸い出すと JPEG に変換

HEIF 利用環境 (3/4)

- HEIF のまま吸い出す方法 (macOS High Sierra で)
  - 写真アプリ>ファイル>書き出す>オリジナルを書き出す



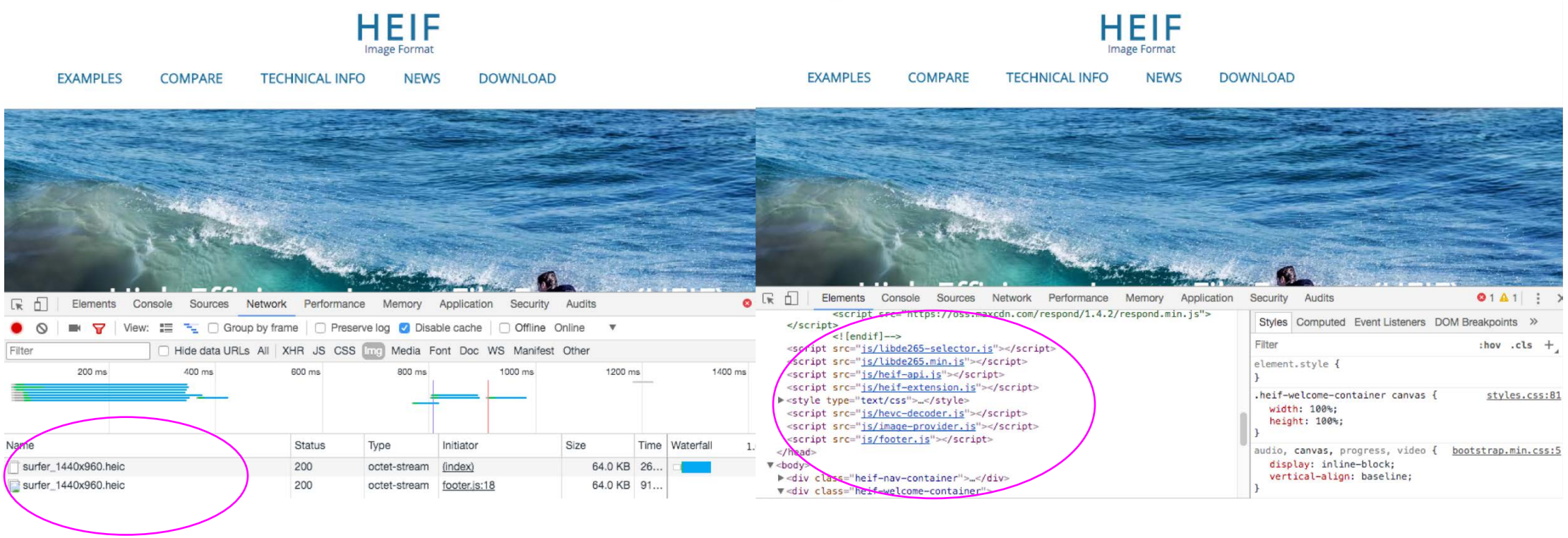


IMG_0001.HEIC

HEIF 利用環境 (3/4)

- JavaScript でブラウザ表示できる

http://nokiatech.github.io/heif/

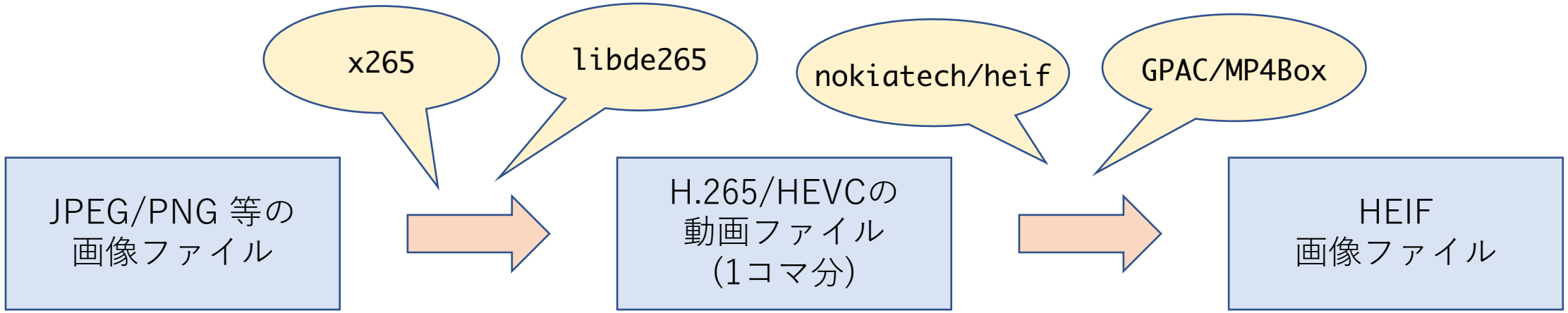


HEIF 画像サンプル

- Nokia Technologies
  - https://nokiatech.github.io/heif/examples.html
  - https://github.com/nokiatech/heif/tree/gh-pages/content/images
- GPAC
  - http://download.tsi.telecom-paristech.fr/gpac/MPEG/ISOBMFF-Conformance/heif/
    - 各ファイルの説明: http://download.tsi.telecom-paristech.fr/gpac/MPEG/ISOBMFF-Conformance/HEIF_conformance.xlsx
  - http://download.tsi.telecom-paristech.fr/gpac/MPEG/IFF/Conformance/

HEIF 変換/解析ツール

- 変換ツールいろいろ



- http://x265.org/
- http://www.libde265.org
- http://nokiatech.github.io/heif/
- https://gpac.wp.imt.fr/mp4box/

HEIF 変換 (ffmpeg と MP4Box) (1/2)

- x265 のフロントエンドとして ffmpeg を使う
  - (log-level オプションがないと大量にメッセージでる)

```
% ffmpeg -i input.jpg -loglevel warning -pix_fmt yuv420p ¥
        -vcodec libx265 -f hevc -x265-params log-level=1 ¥
        -crf 12 -preset slower tmp.hevc
%
% MP4Box -add-image tmp.hevc -quiet -ab heic -new output.heic
%
```

  - crf の値で画質とファイルサイズのトレードオフを調整できる
    - 小さいと画質が良くなり、大きいとファイルサイズが減る

HEIF 変換 (ffmpeg と MP4Box) (2/2)

- 64x64 が最小単位。(それ未満はエンコードできない)

```
% convert logo: -resize 63x63 63x63.jpg
% ffmpeg -i 63x63.jpg -loglevel warning -pix_fmt yuv420p ¥
        -vcodec libx265 -f hevc -x265-params log-level=1 ¥
        -crf 12 -preset slower tmp.hevc
x265 [error]: Picture size must be at least one CTU
x265 [error]: Picture width must be an integer multiple of the
specified chroma subsampling
x265 [error]: Picture height must be an integer multiple of the
specified chroma subsampling
```

- YUV420 (今のところ iOS は 444 NG)だと奇数サイズNG

HEIF 解析ツール (1/2)

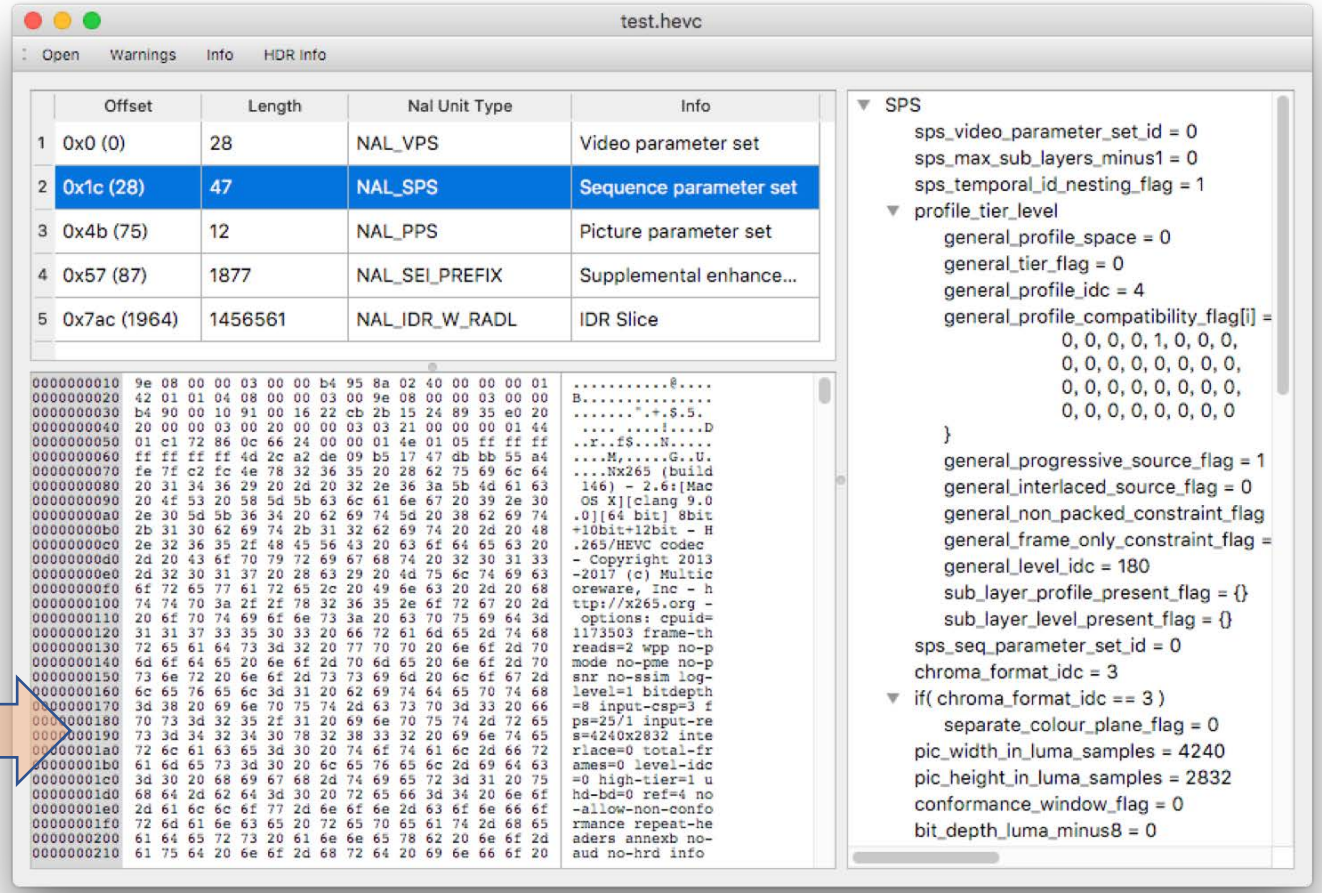
- GPAC/MP4Box が便利
  - 情報要素を XML で分解します

```
% MP4Box -std -diso  out_t.heic
<?xml version="1.0" encoding="UTF-8"?><!--MP4Box dump trace--
><IsoMediaFile xmlns="urn:mpeg:isobmff:schema:file:2016"
Name="out_t.heic"><FileTypeBox Size="24" Type="ftyp"
Specification="p12" Container="file" MajorBrand="mif1"
MinorVersion="0"><BrandEntry AlternateBrand="mif1"/><BrandEntry
AlternateBrand="heic"/></FileTypeBox>
＜略＞
```

HEIF 解析ツール (2/2)

- H.265/HEVC の調査
- HEVCESBrowser が便利
  - Xcode で簡単ビルド

```
% ./hevcesbrowser test.hevc
```



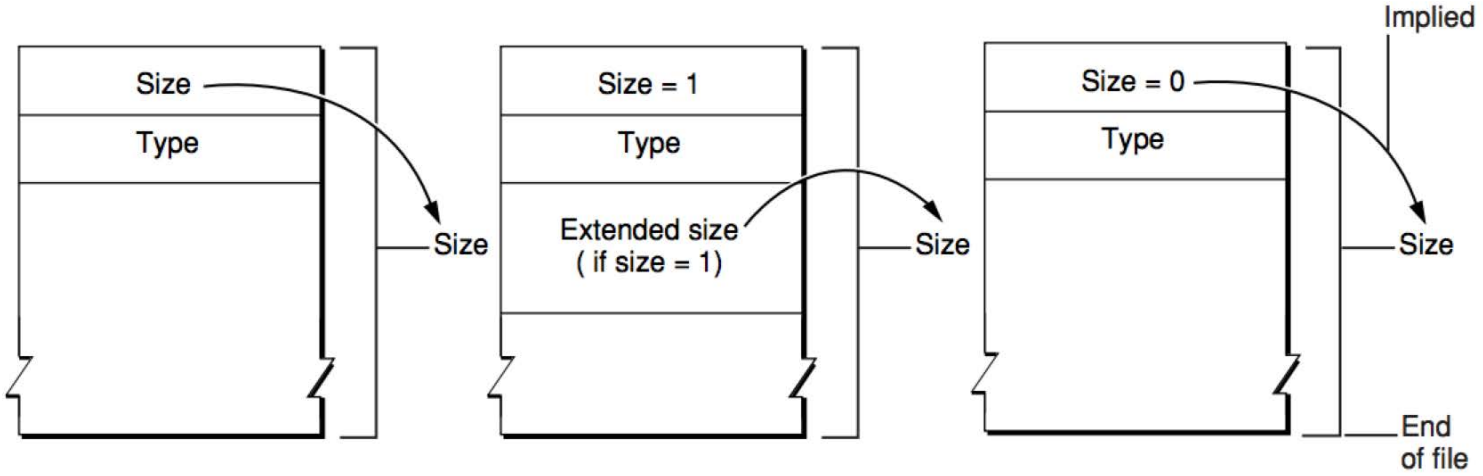
- https://github.com/virinext/hevcesbrowser

バイナリ話

ここから本題

HEIF コンテナ構造

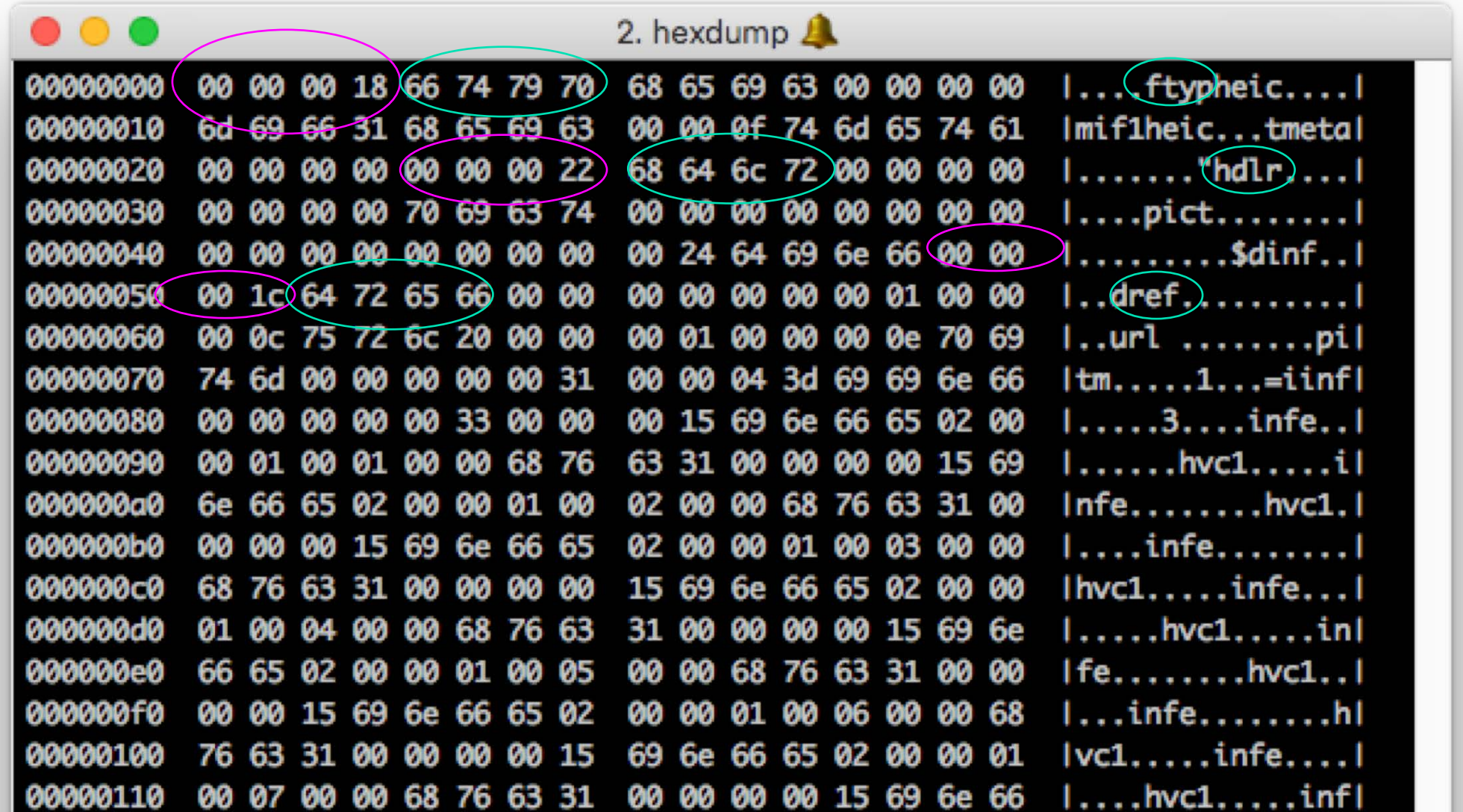
- ISOBMFF は簡単にバイナリ分解できる
  - Size + Type + Payload



© https://developer.apple.com/standards/qtff-2001.pdf

(親になるコンテナボックスがちょっと面倒)

HEIF コンテナ構造


2. hexdump
00000000 00 00 00 18 66 74 79 70 68 65 69 63 00 00 00 00 |....ftypheic....|
00000010 6d 69 66 31 68 65 69 63 00 00 0f 74 6d 65 74 61 |mif1heic...tmeta|
00000020 00 00 00 00 00 00 00 22 68 64 6c 72 00 00 00 00 |.......'hdlr....|
00000030 00 00 00 00 70 69 63 74 00 00 00 00 00 00 00 00 |....pict........|
00000040 00 00 00 00 00 00 00 00 00 24 64 69 6e 66 00 00 |.........$dinf..|
00000050 00 1c 64 72 65 66 00 00 00 00 00 00 00 01 00 00 |..dref..........|
00000060 00 0c 75 72 6c 20 00 00 00 01 00 00 00 0e 70 69 |..url ....pi|
00000070 74 6d 00 00 00 00 00 31 00 00 04 3d 69 69 6e 66 |tm.....1...=iinf|
00000080 00 00 00 00 00 33 00 00 00 15 69 6e 66 65 02 00 |.....3....infe..|
00000090 00 01 00 01 00 00 68 76 63 31 00 00 00 00 15 69 |......hvc1.....i|
000000a0 6e 66 65 02 00 00 01 00 02 00 00 68 76 63 31 00 |nfe.........hvc1.|
000000b0 00 00 00 15 69 6e 66 65 02 00 00 01 00 03 00 00 |....infe........|
000000c0 68 76 63 31 00 00 00 00 15 69 6e 66 65 02 00 00 |hvc1.....infe...|
000000d0 01 00 04 00 00 68 76 63 31 00 00 00 00 15 69 6e |.....hvc1.....in|
000000e0 66 65 02 00 00 01 00 05 00 00 68 76 63 31 00 00 |fe........hvc1..|
000000f0 00 00 15 69 6e 66 65 02 00 00 01 00 06 00 00 68 |...infe........h|
00000100 76 63 31 00 00 00 00 15 69 6e 66 65 02 00 00 01 |vc1.....infe....|
00000110 00 07 00 00 68 76 63 31 00 00 00 00 15 69 6e 66 |....hvc1.....inf|

HEIF コンテナ構造

- 手作業で分解するの面倒なのでツール作成
  - https://github.com/yoya/IO_HEIF (composer 対応)

```
% composer require yoya/io_heif
% php vendor/yoya/io_heif/sample/heifdump.php -f test.heic
type:ftyp(offset:0 len:28):File Type and Compatibility  major:mif1 minor:0
alt:mif1, heic, hevctype:meta(offset:28 len:512):Information about items
version:0 flags:0    type:hdlr(offset:40 len:33):Handler reference
version:0 flags:0      componentType:^@^@^@^@      componentSubType:pict
＜略＞
```

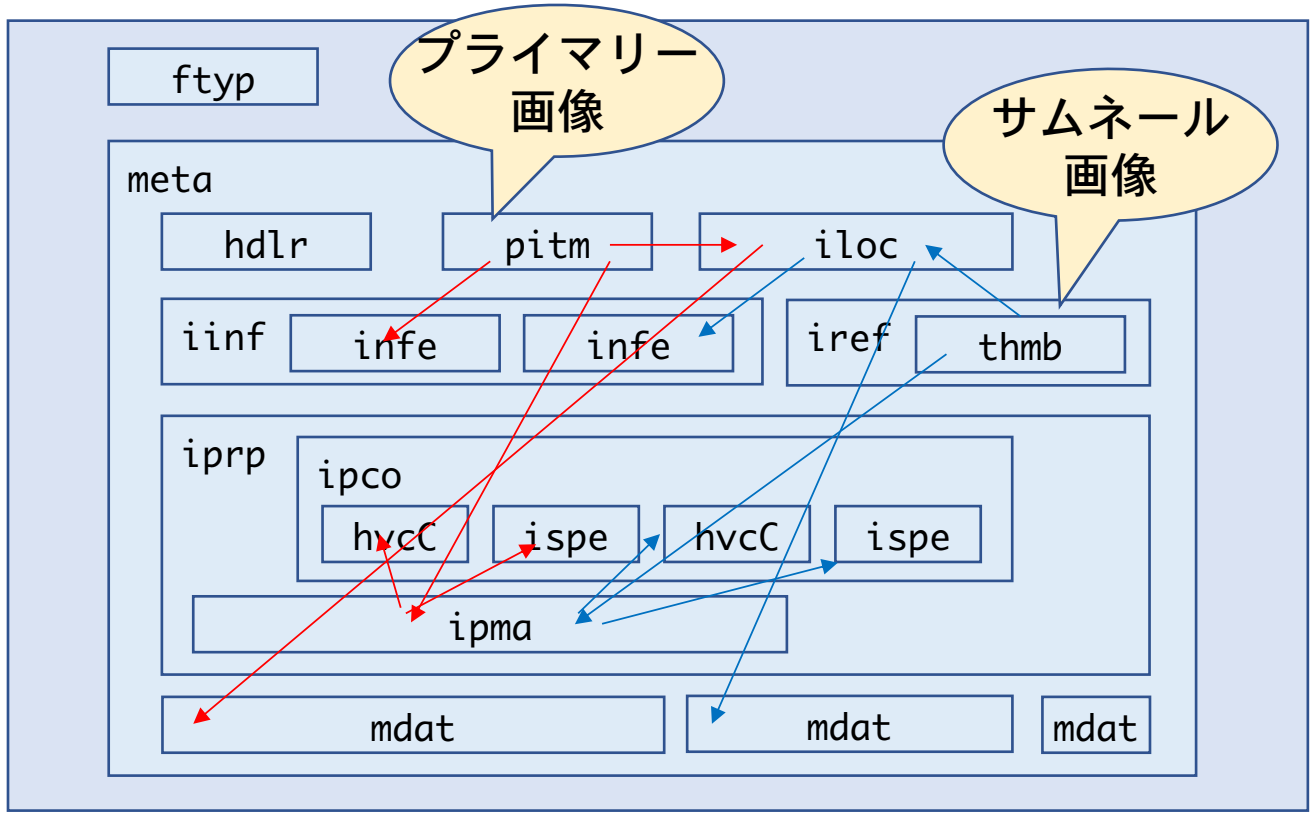
HEIF コンテナ構造

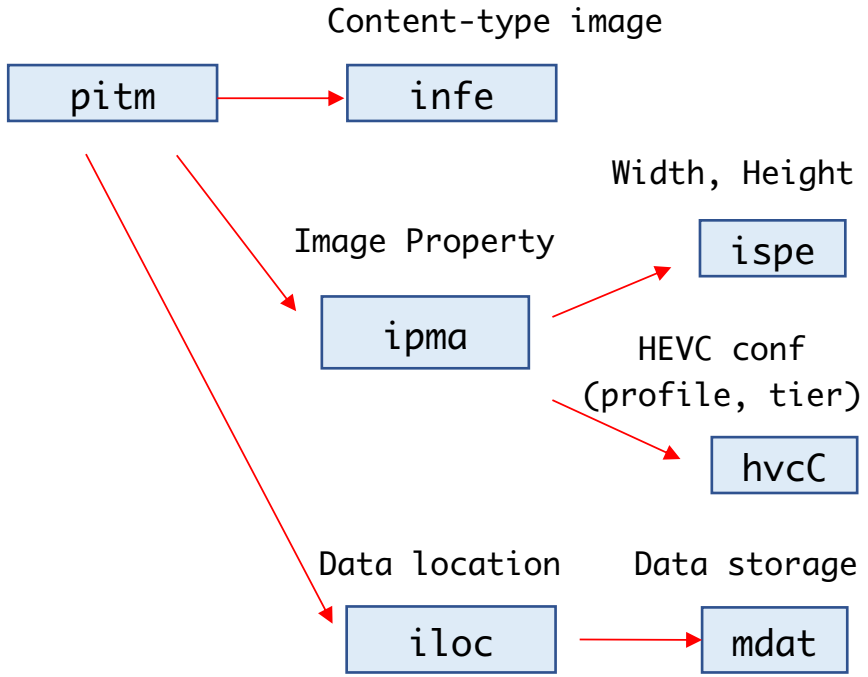
- -h で 16進数表示

```
% php vendor/yoya/io_heif/sample/heifdump.php -h -f test.heic
type:ftyp(offset:0 len:28):File Type and Compatibility
  major:mif1 minor:0  alt:mif1, heic, hevc
             0  1  2  3  4  5  6  7   8  9  a  b  c  d  e  f
0123456789abcdef
0x00000000  00 00 00 1c 66 74 79 70  6d 69 66 31 00 00 00 00      ftypmif1
0x00000010  6d 69 66 31 68 65 69 63  68 65 76 63              mif1heichevc
type:meta(offset:28 len:512):Information about items  version:0 flags:0
             0  1  2  3  4  5  6  7   8  9  a  b  c  d  e  f
0123456789abcdef
0x00000010                                    00 00 02 00
0x00000020  6d 65 74 61 00 00 00 00                        meta
＜略＞
```

HEIFコンテナ構造 (nokiatech/heif)

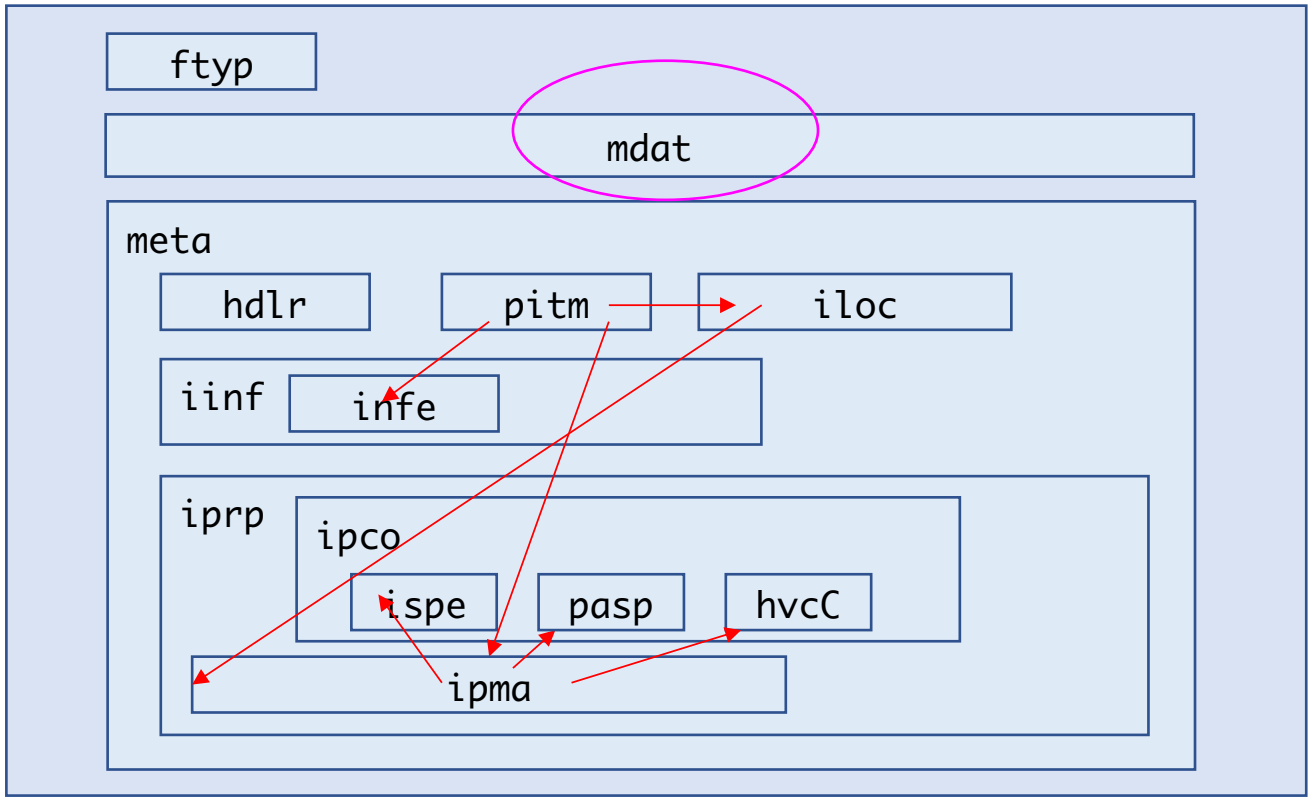
- autumn_1440x960.heic





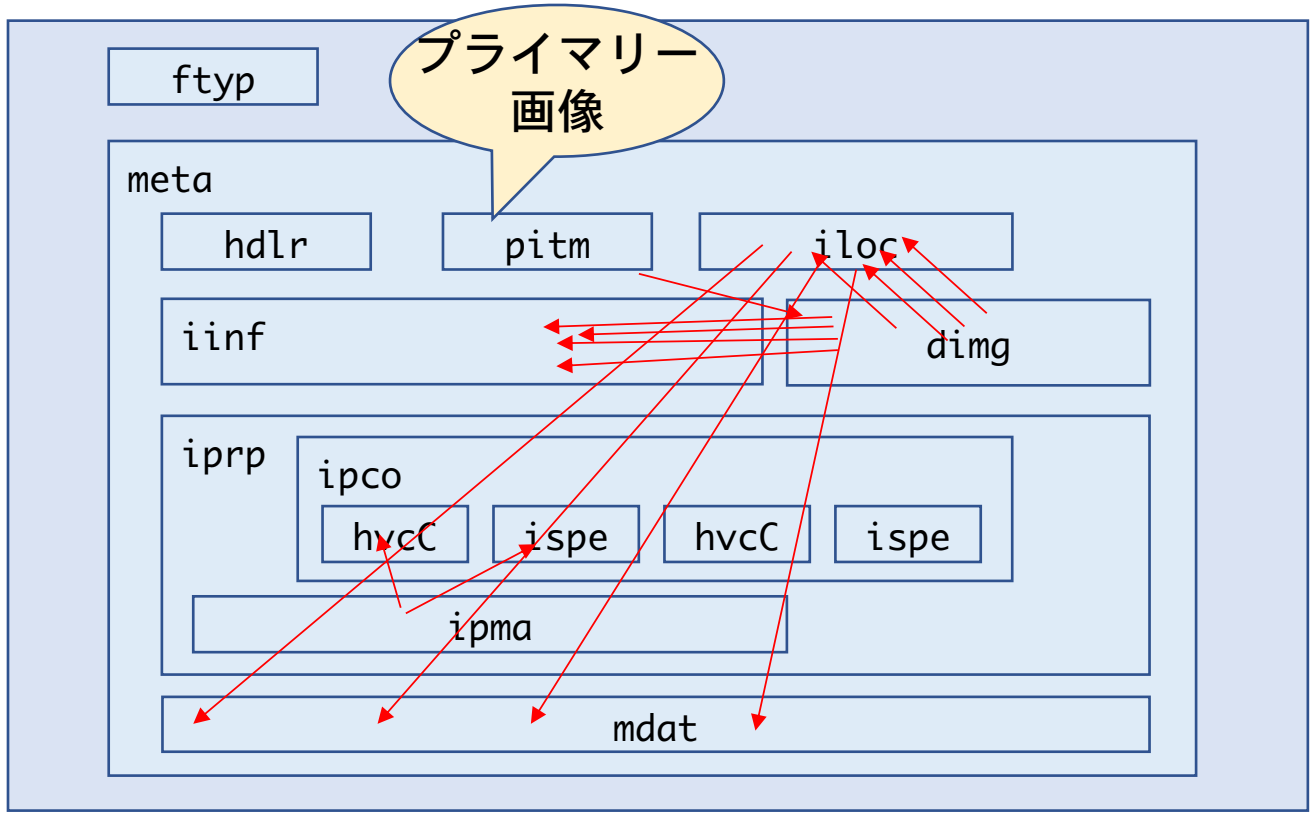
HEIFコンテナ構造 (GPAC MP4Box)

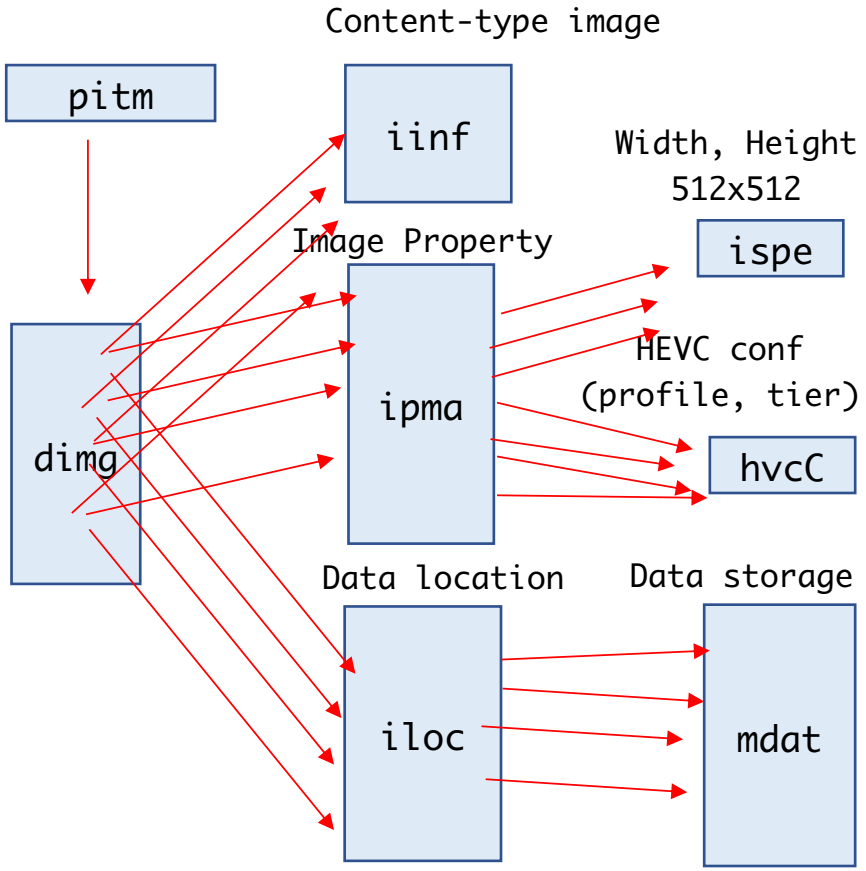
- mdat (画像データ)が頭の方にくる



HEIFコンテナ構造 (iPhone X から吸出し)

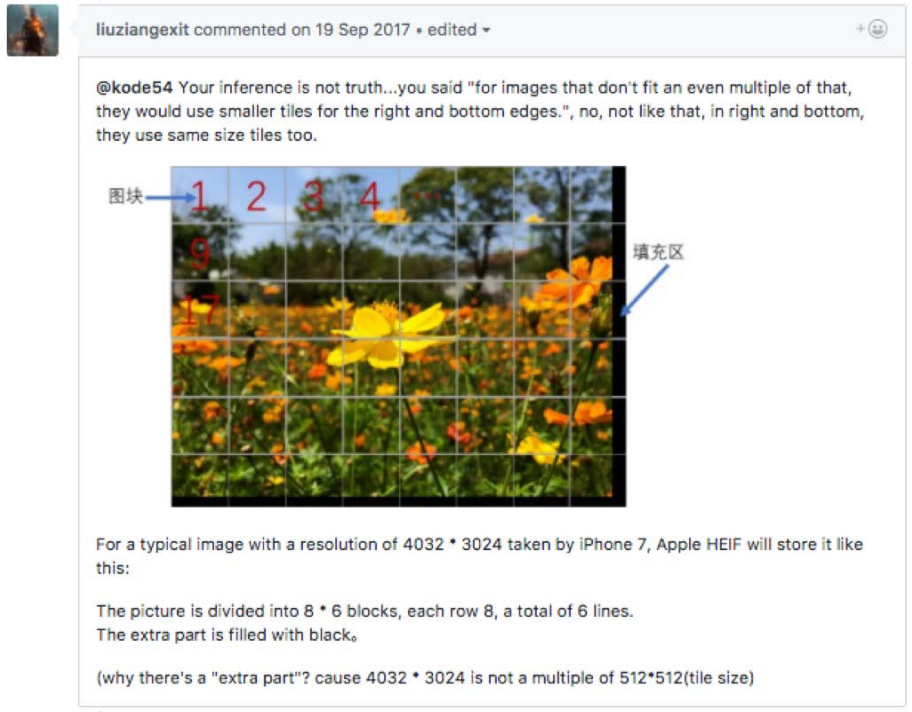
- 512x512 の画像が沢山ある？





HEIFコンテナ構造 (iPhoneX)

- タイリング仕様
  - https://github.com/ImageMagick/ImageMagick/issues/507#issuecomment-330382041
- 512x512 単位で分割



liuziangexit commented on 19 Sep 2017 • edited

@kode54 Your inference is not truth...you said "for images that don't fit an even multiple of that, they would use smaller tiles for the right and bottom edges.", no, not like that, in right and bottom, they use same size tiles too.

For a typical image with a resolution of 4032 * 3024 taken by iPhone 7, Apple HEIF will store it like this:

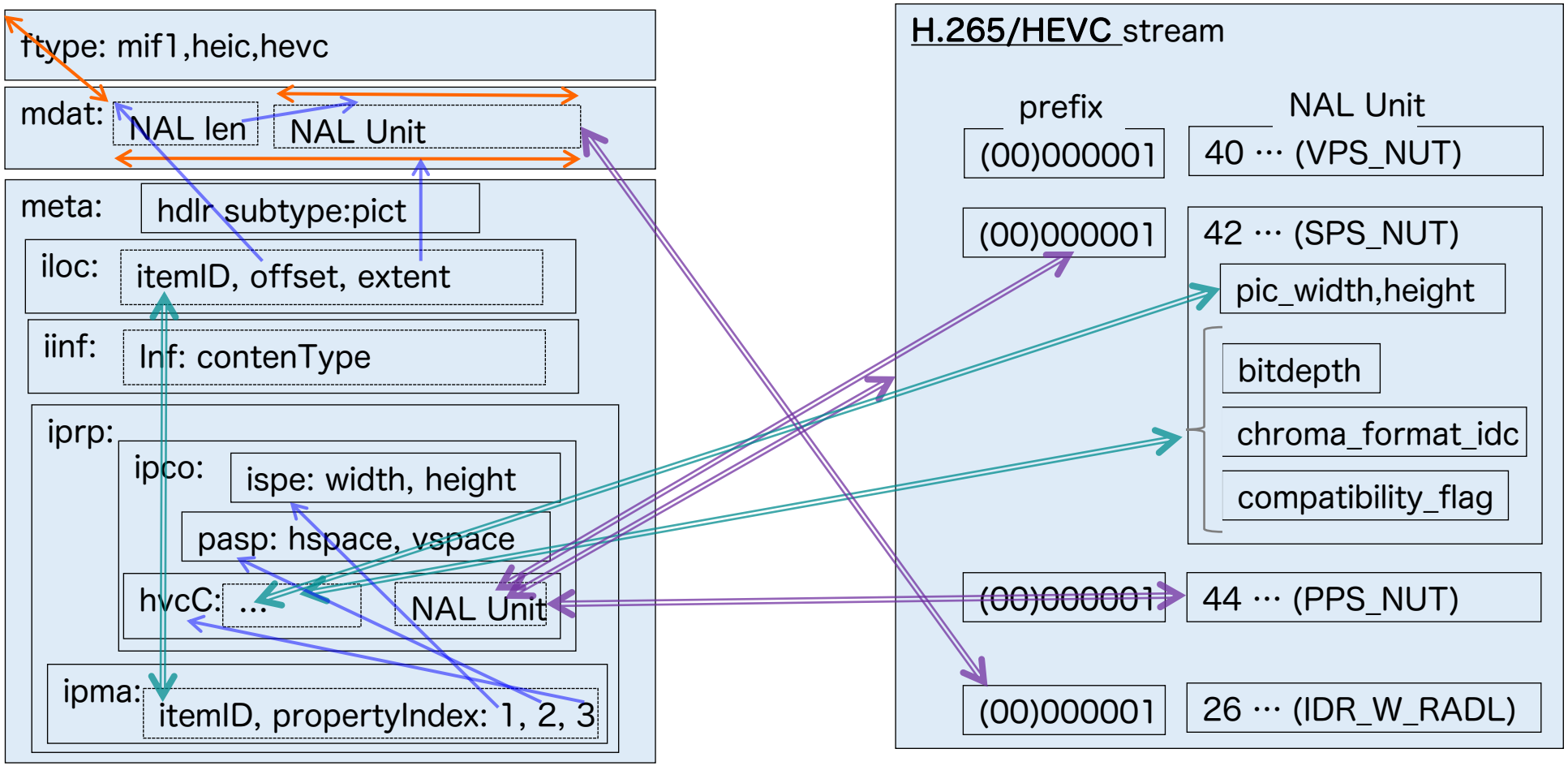
The picture is divided into 8 * 6 blocks, each row 8, a total of 6 lines.
The extra part is filled with black。

(why there's a "extra part"? cause 4032 * 3024 is not a multiple of 512*512(tile size)

HEIF 生成

- ここまで理解すれば、H.265/HEVC から HEIF に自分で変換できる

自分で H.265/HEVC => HEIC 変換



PHP で実験実装

- 手作業で分解するの面倒なのでツール作成
  - https://github.com/yoya/IO_HEIF
    - https://github.com/yoya/IO_HEIF/blob/master/sample/heiffromhevc.php

```
% composer require yoya/io_heif
% convert wizard: wizard.jpg
% ffmpeg -i wizard.jpg ¥
      -pix_fmt yuv420p -vcodec libx265 ¥
      -f hevc -preset slower wizard.hevc
% php vendor/yoya/io_heif/sample/heiffromhevc.php ¥
      -f wizard.hevc > wizard.heic
```

wizard.heic

問題点

- MPEG なのでライセンス怖くない？

ライセンス問題

- H.265/HEVC はパテントプールが分かれてちゃってる
  - パレントプールとは。。。

    - MPEG LA
    - HEVC Advance
    - Velos Media
    - Technicolor

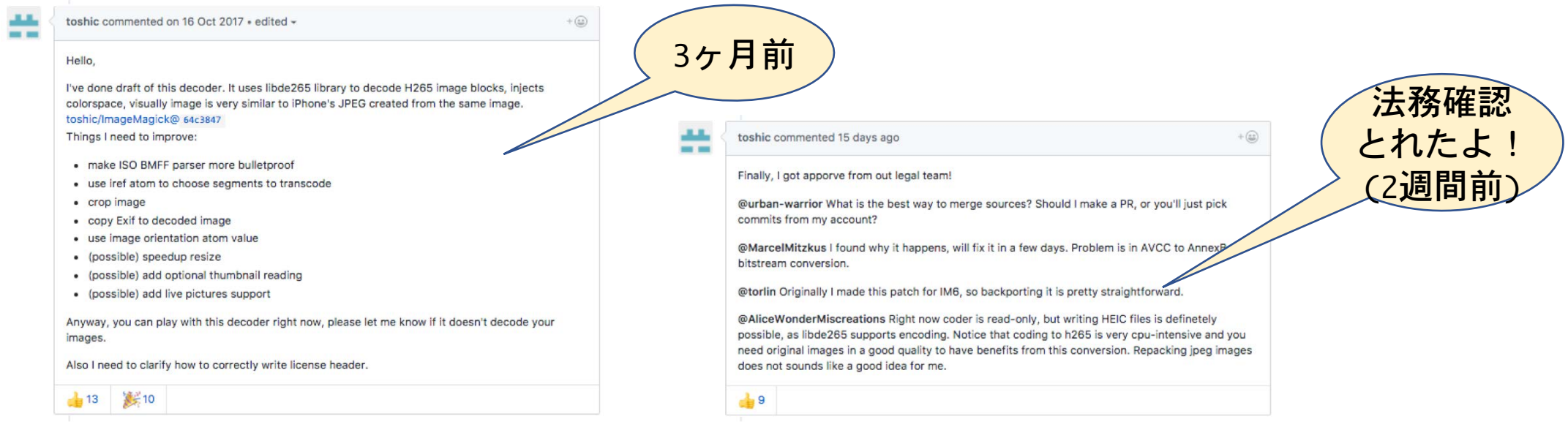
- 参考: https://qiita.com/yohhoy/items/c2579097a507b1fbdddb

HEIF 最近のトピック(1/2)

- HEVC の代わりに AV1 を入れる提案が OpenMedia から
  - **AV1 Still Image File Format (AVIF)**
- https://aomediacodec.github.io/av1-avif/
  - https://github.com/AOMediaCodec/av1-avif
  - 提案者は Netflix 。Google や Mozilla もこの陣営
  - H.265/HEVC のようなパテント縛りがない！

HEIF 最近のトピック (2/2)

- ImageMagick が対応しそう
  - https://github.com/ImageMagick/ImageMagick/issues/507



  - https://github.com/ImageMagick/ImageMagick/blob/master/coders/heic.c

最後に (課題)

- 調べきれずに話に混ぜられなかったもの
  - 画質評価
    - いろいろ評価方法がある
  - ノンブロッキング
  - Derived Image (派生画像)
    - 回転とかオーバーライドとか

以上です

- ご質問があれば是非。もしくは間違いの指摘なども