

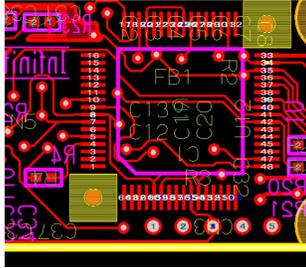
AndroidとUSBカメラでWebRTC

2018/10/9

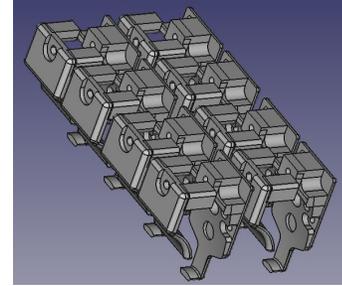
インフィニテグラ株式会社 馬場 鉄平

自己紹介

- 主にプログラミング、たまに回路・基板設計、筐体設計しています



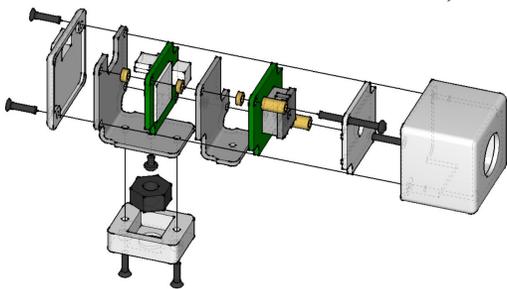
サーマルカメラ



3Dプリンタデータ



モデリングマシン



液体窒素実験



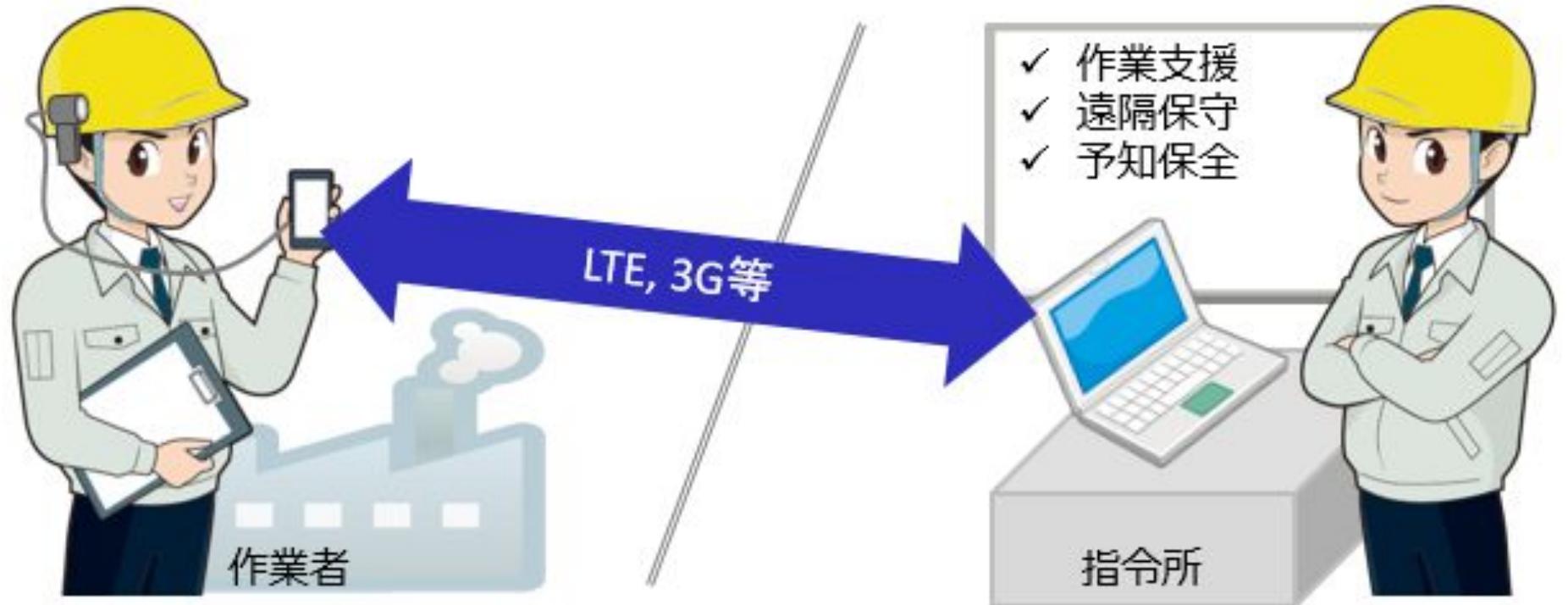
RasPi



LED栽培

ウェアラブルなビデオチャットシステム RazVision WR

- Android端末 + USBカメラ でウェアラブル
- Androidは専用アプリ、PCはChrome専用
- Ruby on Rails
- PeerJS リプレースしたい...。参考:「PeerJSを今後も使い続けるのは危険」@Nakaさん



AndroidでUSBカメラを使用するには

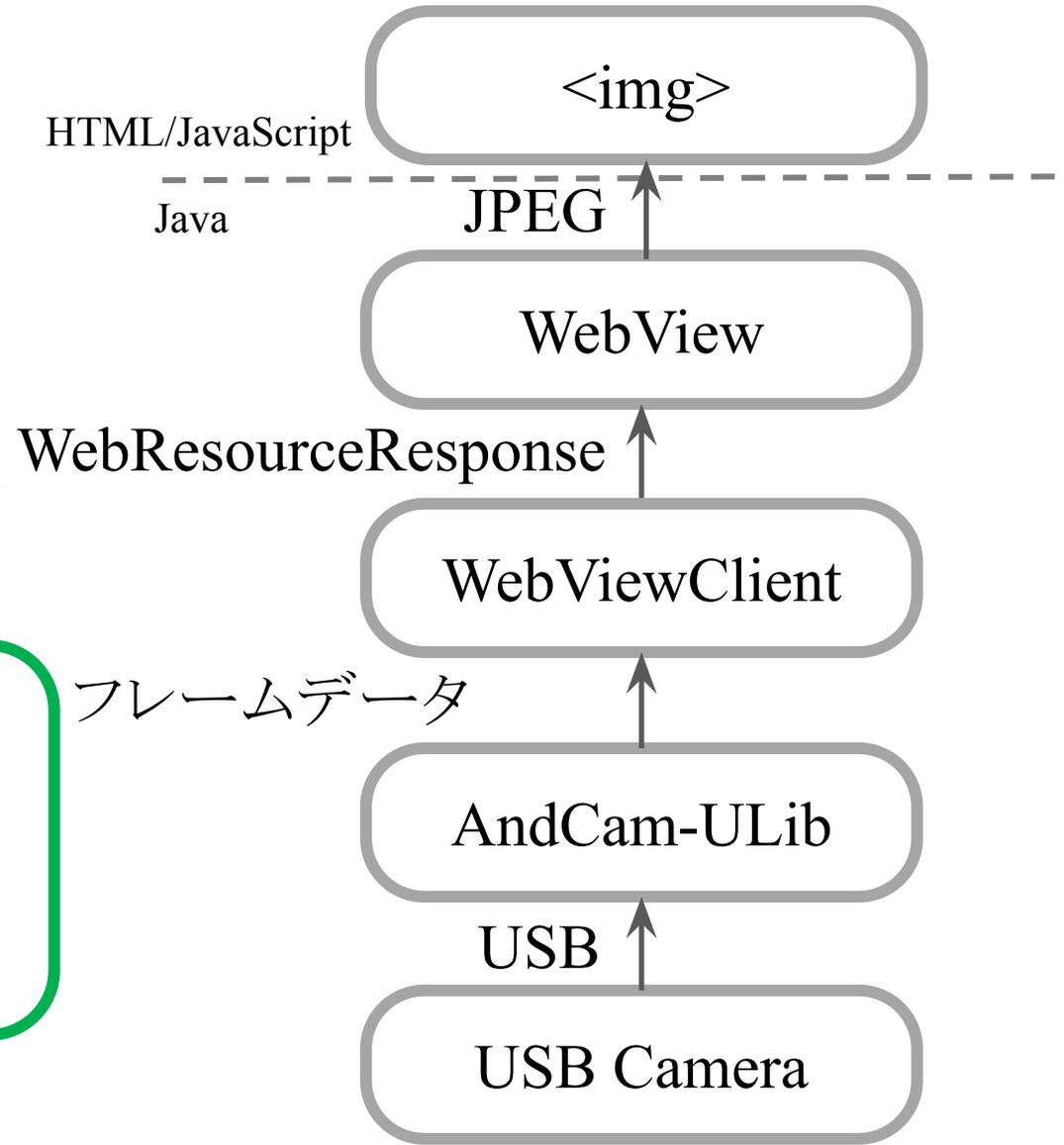
- Android ~8.0
USBカメラは非対応、外部ライブラリで対応
 - 弊社開発のライブラリ「AndCam-ULib」
 - オープンソースのライブラリ
- Android 9.0
ついにUSBカメラ標準対応



MediaStreamを生成するまで(1) - 画像の取得

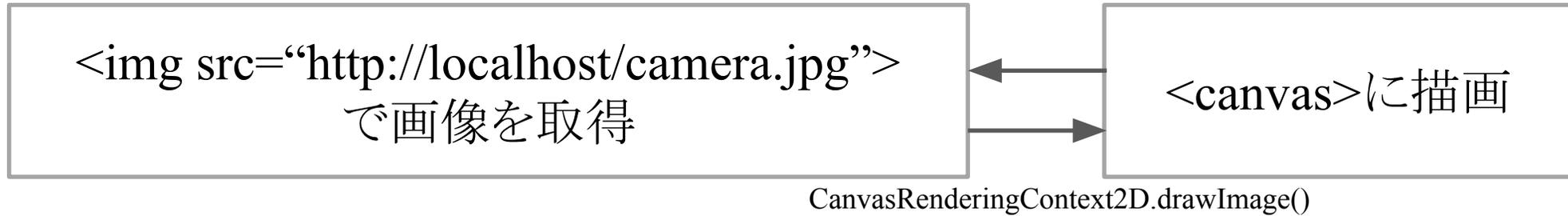
- アプリはWebView 使用 (実体はChrome)

特定URLをカメラ画像に割り当て
WebViewClient.shouldInterceptRequest()
(例) `http://localhost/camera.jpg`

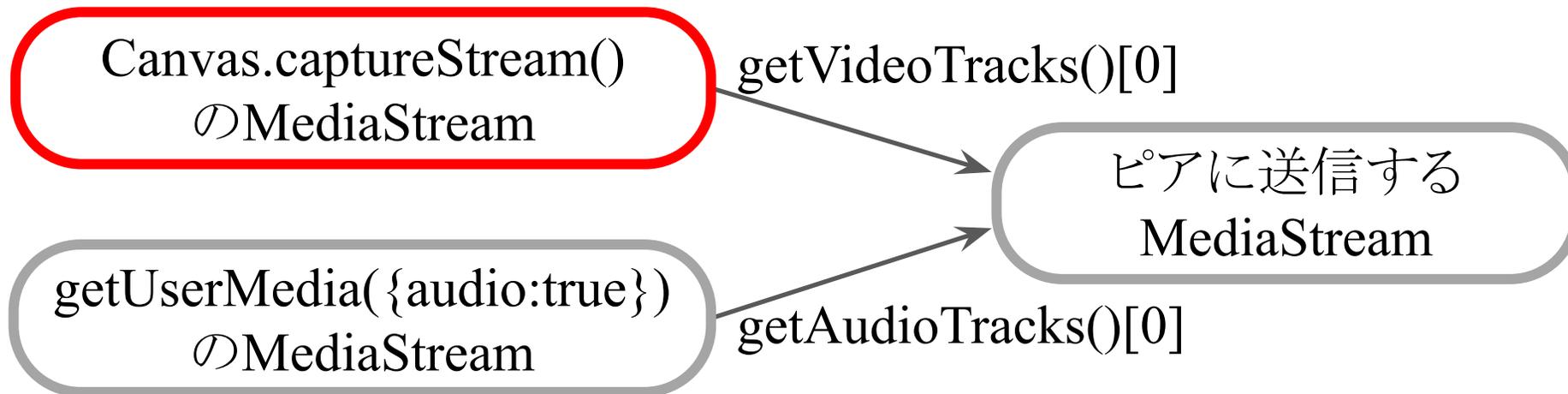


MediaStreamを生成するまで(2) - Canvasの利用

①画像の更新



②MediaStreamの生成

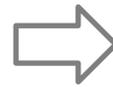


応用

- 画像のソースがUSBカメラでなくても応用可
- 画像を生成して送信する手法

例)

カメラ画像



背景除去



送信

問題発生:時間が経過すると解像度が落ちる

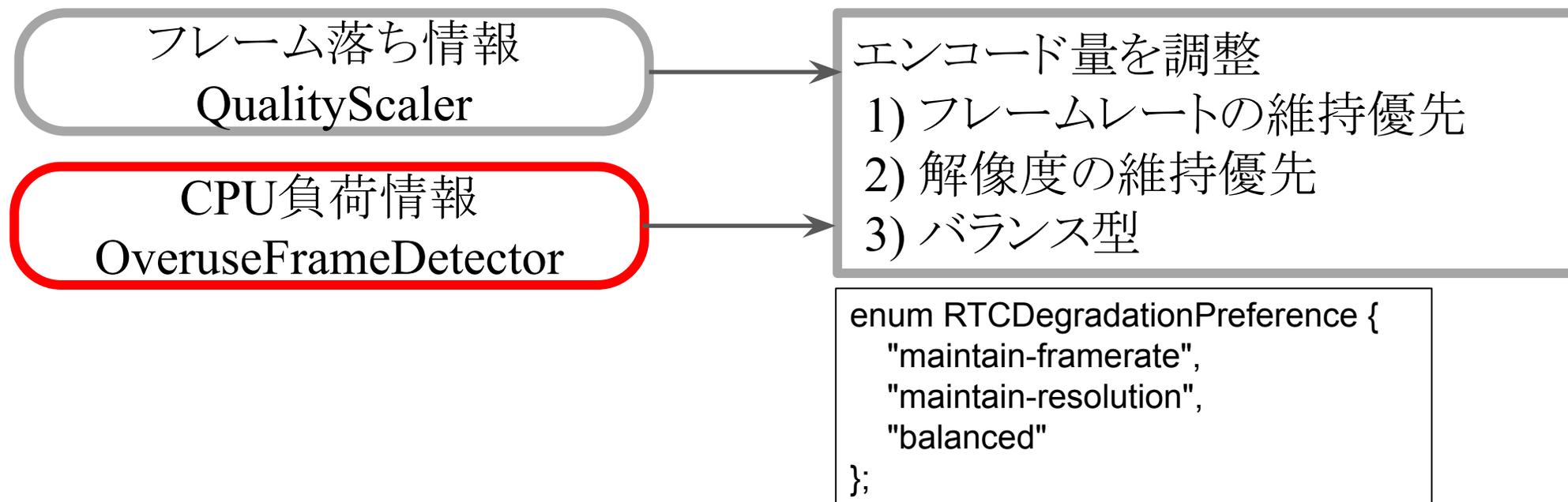
- 通信開始後、数分で画素数が1/16以下に低下
- 機種ごとに低下速度が異なる
- SDPのVP8独自オプション
x-google-start-bitrate,
x-google-min-bitrate,
x-google-max-quantization
なども試したが解決せず

chrome://webrtc-internals
受信された画像の解像度の推移



原因: CPU負荷が高いとき解像度を下げる処理がある

- 弊社アプリではUSBカメラの画像を取得する分、(内蔵カメラに比べて)CPU使用率が高い
- WebRTCのモジュールにおいて
CPU負荷が高いとき画像エンコードの量を調整する処理を発見
video_stream_encoder.cc, overuse_frame_detector.cc



解決: RTCPeerConnection に googCpuOveruseDetection:false を渡す

Chrome/ChromiumでCPU負荷検出の処理を無効にする

```
new RTCPeerConnection(...,  
  { optional: [{googCpuOveruseDetection:false}] })
```

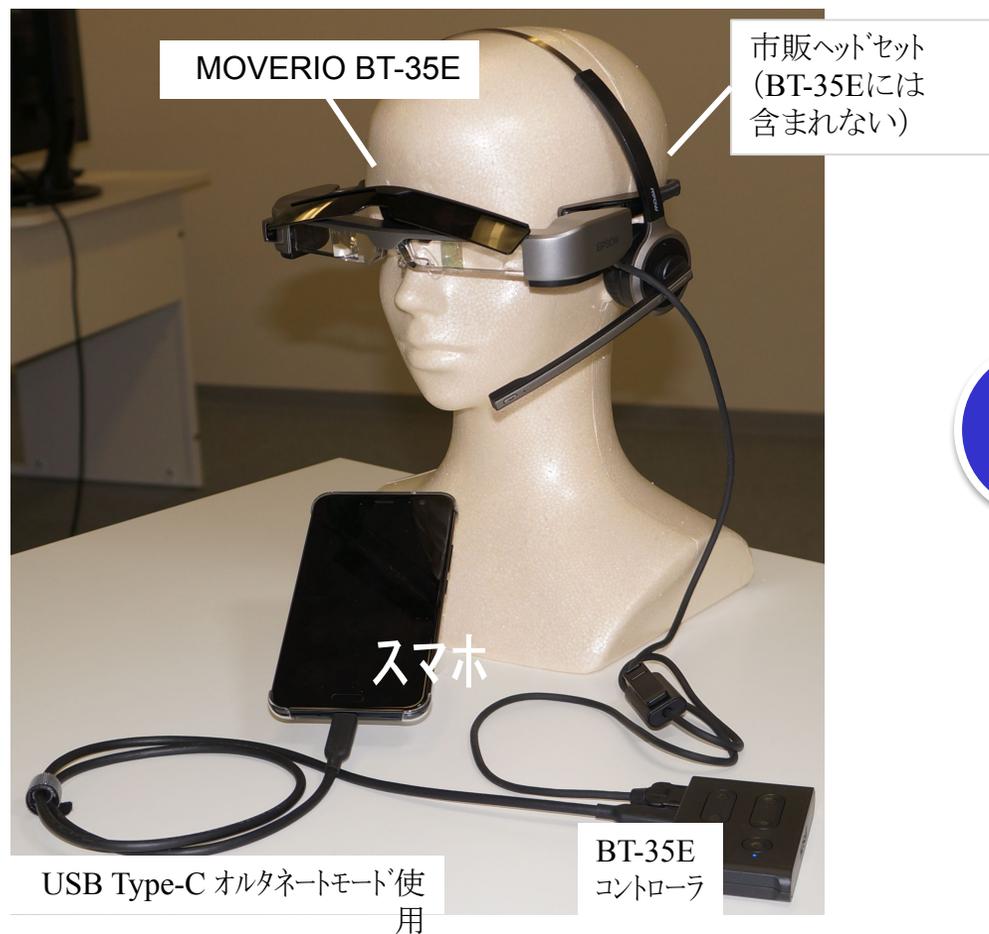
【参考】

- 他の非公式オプションについては mediaconstraintsinterface.cc を参照。

```
// TODO(ronghuawu): Remove once cpu overuse detection is stable.  
const char MediaConstraintsInterface::kCpuOveruseDetection[] =  
"googCpuOveruseDetection";
```

MOVERIO BT-35E対応

- RazVision WRが動くようにしてみました！
- 休憩時間に試してください



一緒にモノづくりしませんか？