

高品質動画処理のための低 CPU 負荷監視カメラシステムの開発

Development of Surveillance Camera System with Low CPU Load for High-Quality Video Processing

深井裕二*

Yuji Fukai

概要

監視カメラシステムはセキュリティ対策をはじめ安全向上や遠隔管理など広範囲に活用されており、装置やシステムの高性能化、応用分野の拡大および生活への浸透が進んでいる。詳細かつ滑らかなカメラ映像は視認性および記録情報としての利用価値が高いが、監視カメラシステムは導入コストが高く、大量のデジタルデータを扱う高い処理能力が要求される。本研究では動画データのパイプライン処理と GPU (Graphics Processing Unit) の活用により、安価な低スペック CPU 搭載のスモールフォームファクタ PC で実用可能な低負荷監視カメラシステムの設計開発を試みた。

1. はじめに

高精細・高画質のネットワーク監視カメラ⁽¹⁾（高品質 IP カメラ）が普及し、より詳細なカメラ映像により様々な活用の可能性や記録情報としての利用価値が高まった。監視カメラシステムの導入費用は高額であり、中枢となる映像レコーダー機器は価格が数十万円のものも少なくない。それらは数台～十数台のカメラに対する処理性能を持つが、1～2台程度のカメラ構成であれば、PC を用いて安価にシステム構築できる場合がある。一般に PC 利用では

様々なソフトのインストールを考慮し、処理能力に余裕がある高 CPU スペックが好まれるが、特定機能に特化した監視カメラシステムは、必要を満たすスペックでよい。むしろ常時稼働させるため省電力性が重要であり、小型であるほど設置形態の自由度も高い。小型マシンとして、Raspberry Pi⁽²⁾はカードサイズのコンピュータであるが高品質 IP カメラ映像を処理するには能力が低く、上位の性能を有する図 1 のようなスモールフォームファクタ (Small Form Factor, SFF) 型 PC が候補に挙げられる。省電力性の高い SFF 型はデスクトップ PC に比べて低スペックであり、高品質 IP カメラ映像の処理は高負荷となることが予想される。そこで、本研究では低スペックマシンでも、市販の高品質 IP カメラの映像処理が実用可能な監視カメラシステムの設計、開発および性能評価を行った。



図 1 SFF 型低スペック PC

2. 高品質動画処理と低負荷のための設計方針

監視カメラシステムの処理機能である動画の圧縮・展開・保存、画面表示、動体検知などにおいて、高品質化に関与する主な性能項目を次に挙げる。

- 解像度
- フレームレート
- 圧縮方式

* 北海道科学大学工学部情報工学科

これらの項目において、本研究で対象とする高品質 IP カメラの性能レベルを以下に示す。解像度について、映像細部や遠方被写体の確認には高解像度が望ましい。防犯カメラ認定基準 RBSS 0002-8⁽³⁾では、高画素の画像サイズを水平 1280、垂直 720 画素以上としている。高品質 IP カメラ製品⁽⁴⁾ではフルハイビジョン (FHD, 1920×1080pixel, 207 万画素) から 4K (QFHD, 3840×2160pixel, 829 万画素) 程度が主流であり、本研究では FHD 以上を対象とする。次に、被写体の動きを確認する場合、動きが滑らかな高フレームレートが望ましい。RBSS 0002-8 では防犯用デジタルレコーダーの評価に用いる画質レベルを 30fps としており、また同製品群の設定可能上限は 20~30fps 程度であり、本研究では 20fps 以上を対象とする。動画圧縮方式では、一般に多くの製品で採用されている高品質・高圧縮率を有する H.264 圧縮を対象とする。以上のような性能レベルを考慮し、次のように設計方針をたてた。

- (1) 小型省電力の SFF 型 PC での稼働
- (2) 低 CPU 使用率の実現
- (3) ハードウェア画像処理の活用
- (4) リアルタイム動画表示処理の実装
- (5) 有線 LAN 接続による IP カメラ 1 台の使用
- (6) 補助記憶装置への圧縮動画保存処理の実装
- (7) 動体検知による保存容量の効率化

方針(1)では、SFF マザーボード規格として VIA Nano-ITX(120×120mm)、intel NUC(101.6×101.6mm)などの小型サイズであり、かつ、低発熱 CPU 搭載などの省電力性の高い PC を運用対象とする。(2)と(3)は互いの関係性が高く、GPU のハードウェア機能の活用やアルゴリズムによる処理効率化が CPU 使用率低減に有効となる。(4)の動画表示は高品質動画処理の一環として、高詳細性や高フレームレートで現在記録中のカメラ映像のリアルタイム表示機能である。これにはハードウェアレンダリング機能の活用で CPU 負荷を低減できる。(5)の有線 LAN 接続はノイズ耐性や通信品質・速度の安定性が良好で高品質動画処理に適している。(6)のハードディスクへのファイル保存では、膨大な動画サイズのため高圧縮による動画保存が必須となる。圧縮(エンコード)は高負荷であるが、ハードウェア機能の活用により CPU での圧縮を不要にできる。(7)の動体検知時のみの動画保存により、処理の負荷と記録消費量が効率化できる。なお検知処理による負荷上昇の把握と対策が必要となる。

3. システムの機器構成

表 1 は、本システムに用いた SFF 型 PC (Links INTERNATIONAL 製 LIVA Z, 図 1) の仕様である。CPU は intel 製 Pentium N4200 を搭載している。図 2 は同世代における intel 製 CPU の性能比較である。標準的性能の Core i5-7500 と Pentium N4200 を比較すると、ベンチマークテスト PassMark の CPU 性能指標である CPU Mark⁽⁵⁾は、6089 に対し 2113(約 0.35 倍)であり、また、CPU の最大消費電力に相当する TDP (Thermal Design Power, 熱設計電力)は 65W に対し 6W (0.09 倍)である。このように Pentium N4200 は低消費電力で低スペックではあるが、CPU に内蔵された GPU はハードウェアによる H.264 デコードやレンダリングなどが可能である。近年の CPU は GPU 内蔵型であり、GPU カードを増設できない SFF の小さな筐体でも画像処理能力は高い。これを監視カメラシステムに活用すれば、低スペック PC でボトルネックとなりがちな CPU 負荷を低減できる。

表 1 監視カメラシステムの PC 仕様

CPU	Pentium N4200 1.1GHz (Burst 2.5GHz), 4 コア/4 スレッド, TDP 6W, ファンレス
GPU	intel HD Graphics 505 200MHz (Burst 750MHz), Max 8GB
メモリ	8GB (DDR3 1600MHz)
補助記憶	256GB (M.2 SSD)
画面出力	HDMI 4K (30Hz), mini DP 4K (60Hz)
入出力	有線 LAN (100BASE-TX/1000BASE-T), Wi-Fi 802.11ac, Bluetooth 4.0, USB3.0
OS	Windows 10 Pro 64bit
消費電力	5~15W
サイズ	117×128×33 mm

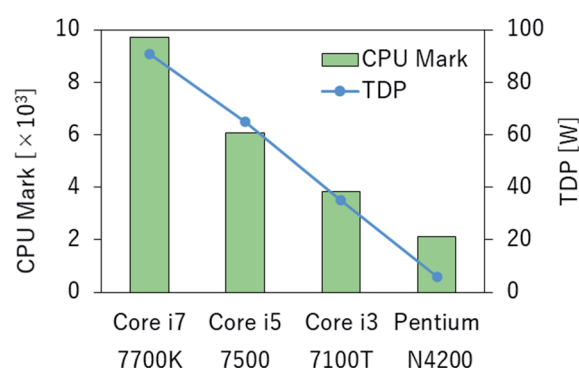


図 2 CPU 性能の比較

本システムでは PoE (IEEE802.3af) 対応の高品質 IP カメラ (SINEI 製 SecuSTATION SC-PE35) を住宅外壁へ設置して使用した。稼働条件として、解像度を WQHD (2560×1440pixel, 368 万画素カラー)、フレームレートを 25fps、圧縮方式を H.264 として RTSP ストリーミングで常時映像配信する。これを PoE 対応スイッチングハブ、CAT6 ケーブルを用いて 100Mbps で接続した。動画の表示には一般的な PC 用 4K モニターを用い HDMI 接続で使用する。動画の保存は、圧縮動画を補助記憶装置 (SSD) へファイル保存する。なお、保存先は外付け HDD や NAS などによる大容量記憶装置へ容易に変更できる。

4. 監視カメラシステムの開発

本システムの開発では、処理方式による性能の違いを確認するために、次の 2 つの処理方式を用いて開発および性能テストを行った。これらの方式では、画像処理ライブラリの OpenCV⁽⁶⁾、マルチメディアフレームワークの GStreamer⁽⁷⁾ を使用し、開発言語・環境に Microsoft VisualC++ 2019 を用いた。

- **OCV 方式**

OpenCV を使用した CPU のみによる処理方式

- **GST 方式**

OpenCV に加え GStreamer を使用し GPU に負荷分散させた処理方式

図 3 は OCV 方式の処理ブロック図である。動画関連における OpenCV の一般的手法として、入力処理に VideoCapture::read メソッド、表示処理に imshow メソッド、保存処理に VideoWriter::write メソッドを用いた。read メソッドは RTSP パケットを H.264 デコードして無圧縮画像を取り出すものである。この画像が得られる度に imshow メソッドでウィンドウの画像表示を更新する。また write メソッドは連続する画像を H.264 圧縮によってファイル保存する。これらの OpenCV ライブラリは高水準で簡易に使える反面、処理途中のデータを外部から参照できない。このため、処理の自由度や柔軟性が低く効率的な処理が記述できない場合がある。サイズ縮小、モノクロ化は動体検知の前処理であり、これらも OpenCV ライブラリを活用した。以上の処理はソフトウェアで行われ、デコード、レンダリング、エンコードなど、CPU に高負荷を与えるものである。

図 4 は GST 方式であり、GStreamer では図 5 のようなパイプラインを構築して並列的かつ効率的な処理を作ることができる。パイプラインの構成要素

(element) は、入力、デコード、変換、保存などの動画処理の機能要素であり、GPU の処理機能を有するものを含む。element は入力(sink)や出力(src)のインタフェース (pad) を持つ。pad 間を接続し element を組み合わせることで、様々な逐次処理、分岐処理、並列処理が構成でき、OpenCV に比べ、より柔軟な処理が行える。GST 方式では、カメラ映像の入力では GPU デコードとファイル保存にパイプラインを分岐させた。これによりデコード前の圧縮動画をそのまま保存できるため、再圧縮のためのエンコードを省略できる。デコード後の無圧縮画像はサイズ縮小と画面表示にパイプラインが分岐し、縮小画像を CPU 側の処理に引き渡し、同時に GPU の 3D レンダリング機能を利用して高速画面描画する。このように GPU のハードウェア処理により、CPU 負荷を下げることができる。

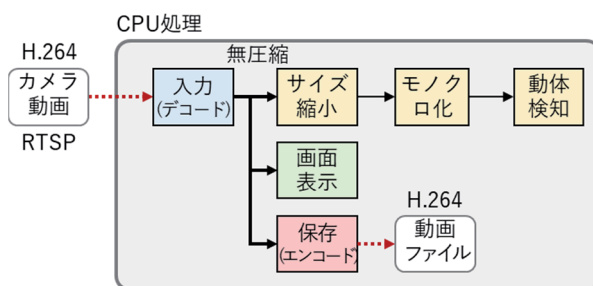


図 3 OCV 方式

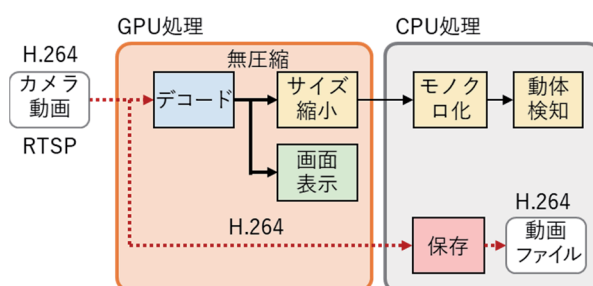


図 4 GST 方式

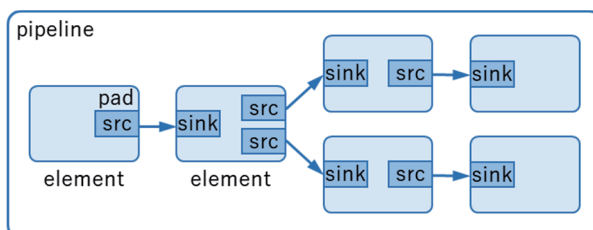


図 5 パイプライン構成の概念

5. 各処理方式の性能評価

監視カメラ映像を常時ファイル保存するシステム方式による実験を行った。本実験バージョンにはカメラ映像のデコード、画面表示、圧縮動画ファイル保存の各機能を実装し、動体検知は実装しない。

図6はシステム実行開始10秒後から1分間における本システムのCPUおよびGPUの平均使用率である。使用率の測定には、OSのパフォーマンスカウンタ機能を用いた測定プログラムを作成して使用した。なお、CPU使用率は全CPUコアの平均値であり、GPU使用率は単位時間当たりの最も負荷の高いGPUエンジン機能の使用率を用い、その平均値を求めたものである。これらの使用率算出法は、Windowsタスクマネージャのパフォーマンス表示⁽⁸⁾における算出法に準拠している。使用率を見るとOCV方式ではCPUが91.7%、GPUが0%である。CPU使用率は100%まで若干余裕があるように見えるが、このときのPC全体のCPU平均使用率は96.7%と飽和状態に近かった。このような高負荷状態ではCPUの処理待ちが多発するため、OCV方式のCPU使用率は本来より低下した値である可能性が高い。一方、GST方式ではCPUが3.5%、GPUが14.3%と処理の多くがGPUに負荷分散され、全体として低負荷に抑えられている。

図7はこのときの画面表示、動画ファイル保存における処理フレームの損失率である。配信動画フレ

ームレートに対しリアルタイムに処理が追いつかないフレームは破棄され損失となる。なお、これらの処理およびカメラ映像のデコード処理は、それぞれ異なるスレッドで実行させ、処理で生ずる遅延の伝搬を抑えている。OCV方式のフレーム損失率を見るとファイル保存で58.5%と大きく、保存動画はコマ落ちが顕著であった。これは、PCのCPU使用率がほぼ飽和状態かつ圧縮動画保存処理が高負荷であるためと考えられる。なお、動画保存を無圧縮のAVI形式に変更した場合、CPU使用率は89.1%に、また、ファイル保存処理のフレーム損失率は40.5%に減少したが、性能劣化は依然大きく、ディスク使用量の大幅な増加も深刻であった。一方、GST方式ではフレーム損失は見られず安定動作している。

図8は本実験時におけるPCのおおよその消費電力である。測定にはPC電源とACコンセント間に挿入するワットチェッカー（測定誤差±30mW）を用いた。グラフ中の破線は、本プログラム実行前のアイドル時におけるPCの消費電力の測定値であり、約4.4Wであった。結果を比較するとOCV方式に対しGST方式は約5.1W低く62.4%に省電力化されている。これはCPU使用率の大幅な減少および低いGPU使用率によるものと思われる。監視映像レコーダー製品⁽⁹⁾の消費電力は約50W～100Wであり、対して本実験結果では、かなり低い消費電力となった。これにより電力ランニングコストに削減効果が期待できる。

本システムの基本的性能比較の実験結果から、GST方式の性能が優れていることがわかる。その主な理由として挙げられることは、高負荷な画像処理のGPUへの分散、パイプライン処理の利点としてファイル保存時の動画再圧縮が不要なことである。処理負荷では、CPU使用率がほぼ頭打ちとなったOCV方式に対しGST方式は余裕があり、解像度やフレームレートのさらなる高品質化や、高品質IPカメラの複数構成などが期待できる。

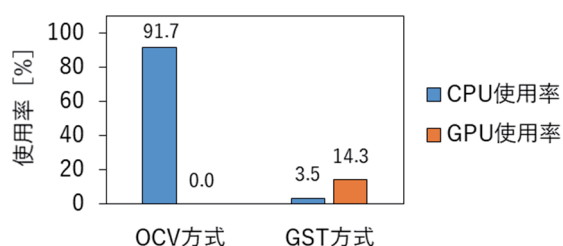


図6 プロセッサ使用率（常時ファイル保存）

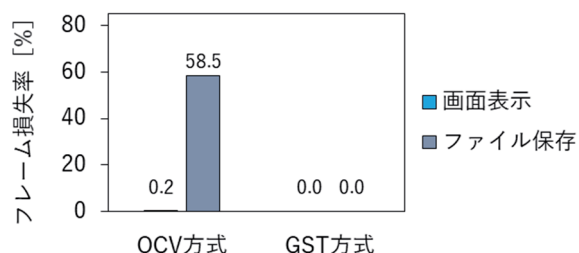


図7 フレーム損失率（常時ファイル保存）

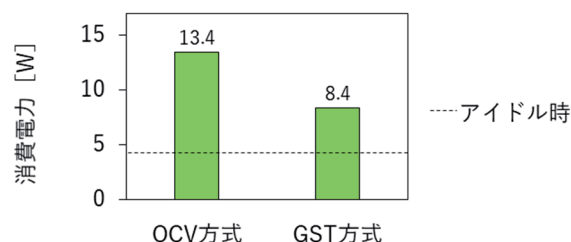


図8 PCの消費電力（常時ファイル保存）

次に、システムに動体検知処理を実装した。本実験バージョンでは、システム用途を防犯や幼児・ペットの監視などとし、動く対象物を記録対象とすることを想定している。よって、動体検知時のみの動画ファイルの保存により記憶容量の消費を抑えられる。また、検知するごとに日時情報をファイル名に含む新たなファイルが作成され、記録動画の利用性が高まる。動体検知に関して次のような処理をOCVおよびGST方式に実装した。

- (1) サイズ縮小
- (2) モノクロ化, ヒストグラム平坦化
- (3) 背景差分抽出
- (4) 2値化, クロージング
- (5) 輪郭検出, 動体判定

これら一連の画像処理は手順が多いため、画素数に応じて処理負荷が大きく増加する。その対策として処理(1)により画像サイズを2560×1440pixelから320×180pixel(縦横1/8, 画素数1/64)に縮小して負荷を軽減させた。(2)では処理の簡素化のためにモノクロ化し、さらにヒストグラム平坦化によりコントラストを明瞭化した。(3)では、動体と背景を区別する背景差分抽出として、混合正規分布(Mixture of Gaussian, MoG)をもとにした前景・背景の領域分割アルゴリズム⁽¹⁰⁾(MoG背景差分)のOpenCV実装であるBackgroundSubtractorMOGを使

用した。MoG背景差分はカメラ映像の緩慢な変動に適応し、人などの速い動体のみ抽出でき、また低負荷な手法でもある。欠点として人がカメラ視野に入った後、じっとしているような場合は抽出できなくなるため、動体の初期検知にのみMoG背景差分を用い、継続検知には初期検知直前のフレームを基準画像とする背景差分を用いた。(4)でグレースケール画像を2値化し、クロージングで動体領域の点状ノイズを除去した。(5)で動体領域の輪郭検出とその面積から動体を判定した。OCV方式ではすべてCPU上で処理するOpenCV機能を使用している。一方、GST方式では(1)のみGPU上で処理し、残りをOpenCV機能で処理している。一般に、GPUとCPU間のデータ転送はボトルネックになりやすいが、縮小した画像データの転送により転送処理時間が短縮できる。

以上のような動体検知処理を実装したバージョンを用いて実験を行った。図9および図10は、動体の未検知および検知時におけるプロセッサ使用率である。OCV方式のCPU使用率は、未検知時89.9%、検知時93.2%と高い。これらは共に動体検知による負荷を含み、検知時はさらにファイル保存による負荷を含む。このときのPC全体のCPU平均使用率は、未検知時96.8%、検知時97.3%と飽和状態に近く、CPUが過負荷である可能性が高い。よってOCV方式は十分にパフォーマンスを発揮できていないもの

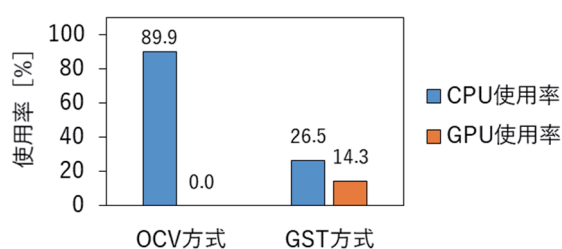


図9 プロセッサ使用率(動体検知 - 未検知時)

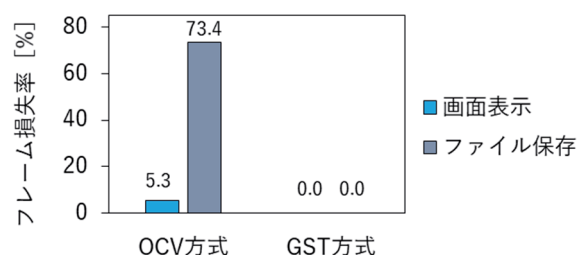


図11 フレーム損失率(動体検知 - 検知時)

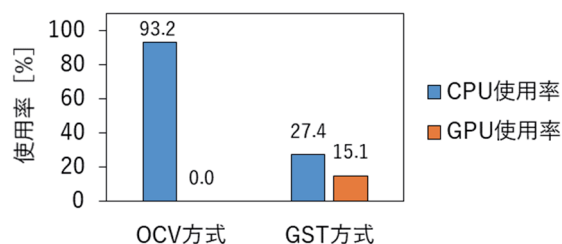


図10 プロセッサ使用率(動体検知 - 検知時)

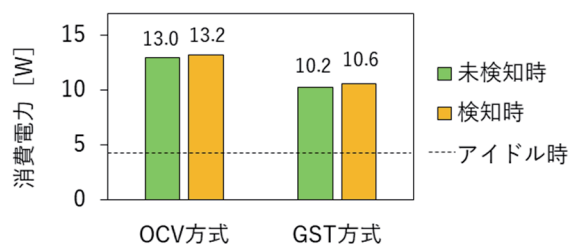


図12 PCの消費電力(動体検知)

と思われる。一方、GST 方式の CPU 使用率は、未検知時 26.5%、検知時 27.4%と CPU リソースの 1/3 に満たない消費状況であり、GPU 使用率も 14~15%程度と低い。本実験の画像解像度 WQHD は 4K に対して画素数が約 44%であり、現状の処理余裕から 4K 画像でも十分処理できる可能性がある。

図 11 は動体検知時のフレーム損失率である。OCV 方式では、図 7 に比べてファイル保存の損失が 73.4%に悪化し、画面表示の損失も 5.3%に増加した。これらの増加は、動体検知による CPU 負荷増加によるものと考えられる。一方、GST 方式は各プロセッサの処理余裕があるため、フレーム損失は見られず安定動作している。

図 12 は、動体の未検知時および検知時における PC の消費電力である。未検知時に対して検知時はプロセッサ使用率の増加とディスクアクセスに応じて電力増加が見られる。また GST 方式は OCV 方式に対し約 80%の電力消費となった。GST 方式を用いた場合、PC の消費電力は常時約 10W であり、概算で 1 か月約 180 円の電力量料金に相当する。

以上の実験結果を総合し、各方式の稼働状況および今回使用した低スペック PC での運用実用性は次のよう評価できる。

- OCV 方式は CPU 使用率が高くファイル保存のフレーム損失が大きい。よって CPU が過負荷状態であり、本マシンスペックでの運用は難しい。完全動作のためには、さらに高い CPU 性能のマシンスペックが必要である。
- GST 方式は CPU および GPU 使用率が低くフレーム損失は見られない。PC 性能に対し処理余裕があり、本マシンスペックで安定稼働できる。
- OCV 方式に対し GST 方式は処理負荷が低いため PC の消費電力が低く省電力性能に優れている。

結論として GST 方式は低 CPU スペックマシンでの高品質監視カメラシステムの運用に適していると思われる。大きな理由として GPU の画像処理能力の利用効果が高く、その実現にあたり GStreamer によるプログラミングが有用であった。GPU を利用する場合、その特有な命令による記述を要する 경우가多いが、本プログラムのソースコードでは GPU 関連命令の記述は無い。本プログラムは、比較的簡易なパイプライン構築と OpenCV による画像処理の記述で構成されている。このように低プログラミングコストで容易に開発できることは、システムの今後の発展においても有利であろう。

6. まとめ

本研究では、低スペックマシンで高品質 IP カメラに対応した監視カメラシステムを開発した。SFF 型 PC を用いることで省電力化を図り、高品質動画の処理に GPU の処理能力を活用した。結果として実用可能な処理負荷レベルが得られ、本研究をもとに低コストである高品質監視カメラシステムが実現できるものと思われる。今後、高品質に加えて動体検知の高精度化や被写体の認識・分類機能の付加など、システムの機能的発展が考えられる。

参考文献

- (1) 公益社団法人日本防犯設備協会: 4K カメラシステムの調査研究報告書, 2017.
- (2) Raspberry Pi Foundation, BCM2711 - Raspberry Pi Documentation, <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/README.md>, 参照日: 2021-2-10.
- (3) 公益社団法人日本防犯設備協会: 優良防犯機器認定基準 RBSS 0002-8 (2018 年版) 防犯カメラ認定基準, 2020.
- (4) パナソニック株式会社: EXTREME シリーズ監視・防犯システム, https://sol.panasonic.biz/security/camera/ipro_extreme/, 参照日: 2021-1-21.
- (5) PassMark Software: CPU Benchmarks, http://www.cpubenchmark.net/cpu_list.php, 参照日: 2021-1-24.
- (6) OpenCV team: OpenCV, <https://opencv.org/>, 参照日: 2021-1-25.
- (7) gstreamer.freedesktop.org: GStreamer: open source multimedia framework, <https://gstreamer.freedesktop.org/>, 参照日: 2021-1-25.
- (8) Pronovost, S.: GPUs in the task manager, <https://devblogs.microsoft.com/directx/gpus-in-the-task-manager/>, 参照日: 2021-1-25.
- (9) パナソニック株式会社: 監視映像レコーダー製品一覧 - 監視・防犯システム, https://biz.panasonic.com/jp-ja/products-services/security_recorder/lineup, 参照日: 2021-1-30.
- (10) KaewTraKulPong, P., Bowden, R.: An improved adaptive background mixture model for real-time tracking with shadow detection, In Video-based surveillance systems, pp.135-144, 2002.