

## ライブカメラ映像を対象としたプライバシー指向映像管理方式の提案

松本哲<sup>1</sup> 義久智樹<sup>1</sup> 川上朋也<sup>2,1</sup> 寺西裕一<sup>3,1</sup>

**概要:** 近年、公の機関が管理するネットワークに繋がったライブカメラが多く設置され、高解像度化、高性能化している。また、一部の交差点などにあるライブカメラは WEB 上で一般公開されている。被撮影物体の映像利用方針の高速把握が難しいために、一般的にライブカメラのシステムにおいて、プライバシー保護機構を適切に設計することは難しいと考えられていた。本研究では、「エッジカメラサーバ連携」「ルール型映像利用方針記述」「共用映像加工認識」と呼ぶ革新的な技術を備えた映像管理方式により、一般の人々が視聴できるライブカメラ映像におけるプライバシー指向映像管理方式（プライバシーの保護を考慮した映像管理方式）を可能にする事を目指し提案する。特に、視聴者の立場により、配信される映像の画像フレーム毎にぼかし処理を自律的に施すなどの情報隠匿を行う機能について、効率化を図る。それにより、配信時に遅延やフレーム落ち等が起こらないようにする保護処理方式の提案をおこなう。

## Proposal of privacy-oriented video management method for live camera video

Satoru Matsumoto<sup>1</sup>, Tomoki Yoshihisa<sup>1</sup>, Tomoya Kawakami<sup>2,1</sup>, Yuuichi Teranishi<sup>3,1</sup>

### 1. はじめに

近年、公の機関が管理するネットワークに繋がったライブカメラが多く設置され、高解像度化、高性能化している。また、一部の交差点などにあるライブカメラは WEB 上で一般公開されている。被撮影物体の映像利用方針の高速把握が難しいために、一般的にライブカメラのシステムにおいて、プライバシー保護機構を適切に設計することは難しいと考えられていた。本研究では、「エッジカメラサーバ連携」「ルール型映像利用方針記述」「共用映像加工認識」と呼ぶ革新的な技術を備えた映像管理方式により、一般の人々が視聴できるライブカメラ映像におけるプライバシー指向映像管理方式（プライバシーの保護を考慮した映像管理方式）を可能にする事を目指し提案する。特に、視聴者の立場により、配信される映像の画像フレーム毎にぼかし処理を自律的に施すなどの情報隠匿を行う機能について、効率化を図る。それにより、配信時に遅延やフレーム落ち等が起こらないようにする保護処理方式の提案をおこなう。

これら対案を実現する際、より高速な人間の検出、柔軟なポリシー記述、および、より高速な画像処理を行える「NGPCS (Next Generation Public Camera System)」と呼ぶプライバシー指向のビデオ配信プラットフォームを提案している。2 章では関連研究について述べ、3 章では提案手法による効率化検討について述べ、4 章ではその評価について述べ、5 章で纏めを述べる。

### 2. 関連研究

我々の研究グループでは、一般の人々が視聴できるライ

ブカメラ映像におけるプライバシー指向映像管理方式（プライバシーの保護を考慮した映像管理方式）を研究開発している[1, 2]。[1]では、ECA ルールによる自律映像処理を伴うトラスト指向インターネットライブ配信についての効率化を図り、[2]では、高性能なクラウドサーバ内でのグラフ処理を用いたぼかし計算の効率化を図った。いずれも、本研究の3章で述べるエッジサーバを主に用いる効率化とは違うアプローチであった。[3]では、放送波とインターネット放送を利用した映像配信についての安定化技術について述べ、インターネット以外を利用した映像配信についてのアプローチも検討している。

[1]で評価されたプラットフォームでは、ハイパーバイザータイプの仮想環境でクラウド上のリソースを使用して評価したが、本研究にて著者らは、拡張性、リソース効率の高いコンテナタイプの仮想環境に着目した。ハイパーバイザータイプでは、ホスト OS を必要とせず、VM を稼働するために OS レベルのコンポーネントハードウェアリソースを用いて、ハイパーバイザーの OS とコンポーネントハードウェアリソースを分離しながら、リソースの割り当て、スケジューリング管理を行う。CPU、メモリ、ストレージなどのリソースを、容易に再配置可能なプールとして扱い、処理は仮想ではなく、物理ハードウェアで行う。ハイパーバイザーによりスケジュール管理をさせながら、VM の CPU 命令も遂行できる。また、複数の異なるオペレーティングシステムを同時に稼働させる事ができ、同じ仮想化されたハードウェアリソースをハイパーバイザーと共有できる特徴がある[4]。ハイパーバイザータイプでは、ゲスト OS を稼働させておかなければならない一方、コンテナタイプ

1 大阪大学

2 福井大学

3 情報通信研究機構

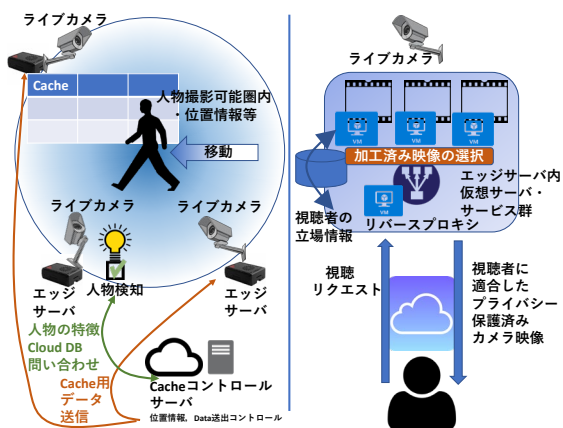


図 1. プライバシー指向映像管理方式の実験環境

の仮想環境では、アプリケーションプログラム、ライブラリや設定ファイルをひとまとめのコンテナとして扱い、コンテナエンジン上で動作させることにより、ゲスト OS すべてを稼働待機させる必要が無い為、必要最小限の CPU やメモリしか使用されず、リソース効率をあげられ、負荷が小さく高速な動作を実現できる。

近年、コンテナタイプの環境では、Docker [5]が一般的に広く使用されている。コンテナの機能を分解して小さなサービスとして提供し、1つのアプリケーションを構成するマイクロサービスアーキテクチャが用いられる事が多くなってきた。近年、マイクロサービスについて、配備/設定/管理の自動化を各サービスのリクエストを中央集権型でコントロールすること（オーケストレーション）で、コンテナ型の仮想環境を安定させ、クラウド環境に適した IP アドレスでサービスを管理できる Kubernetes [6]というプラットフォームが注目されている[7]。Kubernetes は、前述の Docker のようなコンテナタイプの仮想環境で機能し、コンテナタイプとの親和性が高い。著者らは、将来の可用性を考慮して、コンテナタイプの仮想環境の導入を検討した。モノのインターネットに関連する調査がされている[8]では、Kubernetes テクノロジーを適用すると、提案されたアプローチでは、標準のスケジューリングメカニズムと比較してネットワーク遅延が 80%削減されること調査され、示されている。

[9]では、プライバシー決定モデルの設計のために、人間とコンピューターの相互作用に基づく裏付けとなる証拠と、ソーシャルネットワーク上で画像を共有するためのアクセス制御メカニズムを示している。SNS の顔画像タグ付けデータ一覧へのインターフェースがライブカメラの普及とともに充実した場合、ライブカメラの一般利用の実現が加速すると著者らは着目する。例えば、迷子になった子供や犬の顔の写真を公開するためのオプトインとオプトアウトの決定権については、保護者や飼い主が決定権を通常は持つ。しかしながら、友人が SNS で公開している、視聴者自身が

写っている顔の写真のオプトインとオプトアウトについて、要求することはできるが、その制御権はないため、社会的セキュリティは保たれない。しかし、視聴者自身が新たに、自身が SNS で公開している顔写真にタグを付けた場合、その友人がすでに顔写真にタグを付けている場合でも、視聴者自身のオプトイン指示を優先し、ライブカメラの映像より視聴者を見つけられない映像処理を実装することも、肖像権を適切に解決すれば可能と考えられる。著者らは SNS 等のソーシャルログイン情報やタグ情報をもとにライブカメラの検出ルールを自動生成できないか着目している。

### 3. 提案手法による効率化検討

著者らは「エッジカメラサーバ連携」「ルール型映像利用方針記述」「共用映像加工認識」の技術を備えた試行用の統合環境を設計した。一部シミュレータを用いて、プライバシーを確保する機能を有した効率的な映像配信について評価をおこなう。上記 3つの技術については、以下 (1) ～ (3) の通りである。

(1)「エッジカメラサーバ連携」：被撮影物体の識別や映像加工の処理を次世代防犯カメラサーバで行うと、次世代防犯カメラサーバに処理負荷が集中し、処理時間が長くなって映像視聴までの遅延が長くなる問題が起きる。そこで、短時間でエッジサーバ間の連携を行う、送受信方式や選定方式についての効率化を図る技術。

(2)「ルール型映像利用方針記述」：映像利用方針ファイルには、視聴者の属性に応じた映像加工方法を記述する。例えば、被写体個人の立場別にプライバシー配慮を行い、被写体の立場にも添って、個人番号識別番号表示や顔を隠すといった加工を行う。映像利用方針に関する複雑な処理をプログラム等で記述すると、映像利用方針の記述が複雑で困難になる問題がある。それを解決すべく、柔軟に記述を行える仕組みを提供する技術。

(3)「共通映像加工認識」：視聴者毎に視聴映像を生成すると映像生成に伴うエッジカメラサーバの処理負荷が大きくなって映像生成に時間がかかる問題がある。視聴者の属性が異なっても共通する映像加工を短時間で認識する技術。

システムの設計に関して創意工夫を行った点は、移動する被写体となる人物の画像認識用特徴データをシステムで初見された周辺のエッジカメラサーバへキャッシュデータとして予め送信しておき、認証のための問い合わせデータの送受信の効率化を図る点と、想定されるプライバシー保護の画像処理をエッジカメラサーバにて予め行い、映像要求時に視聴者の立場に合わせた加工済み画像フレームを選択し、エッジカメラサーバ環境内のリバースプロキシを介して配信する方式とする。実験環境の全体構成を図 1 に示す。

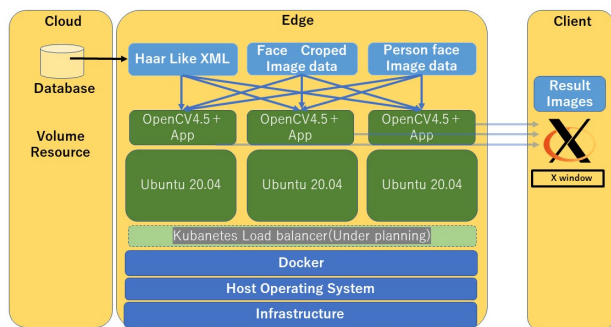


図 2. 評価実験環境のシステム構成

表 1. 評価環境

| 構成     | 構成要素     | 性能                                                |
|--------|----------|---------------------------------------------------|
| VM 環境  | コンテナエンジン | Docker Engine v19.03.13                           |
|        | Guest OS | Ubuntu 20.04 SLT                                  |
| エッジサーバ | OS       | Windows10 Pro                                     |
|        | CPU      | Intel(R) Core(TM) i7-1065G7 CPU @1.30GHz, 1.50GHz |
|        | 主メモリ     | RAM 32.00GB                                       |

## 4. 評価

### 4.1 実験環境

評価に使用した計算処理を分散するのに用いるサーバ及び PC の、OS 及び CPU、メインメモリの性能を表 1 に示す。今回利用したコンピューティング環境は、Docker for Desk top サービス上の Virtual Machine (VM) である。VM の詳細については表 1 に示す。視聴端末と主な計算処理を行う Local PC には、ノート PC を利用する。エッジサーバを、このコンテナタイプの VM とした。この実験環境における顔認識には Open CV 4.5 の Haar-like 特徴分類器を利用した。クラウドサーバ、エッジサーバへの送信は TCP を使用した Socket ライブラリにて通信を行う仮定としてシミュレートした。

### 4.2 評価方法

評価方法は、視聴端末と主な計算処理を行う Local PC により、Harr-Like 特徴分類器を利用して認識された顔画像部分を切り出し、ぼかし処理をかけて、原画像に合成して表示し、1 フレームの画像処理時間を計測した。この結果と、Local PC での特徴分類認識処理の後、切り出した部分のみをクラウドサーバ、エッジサーバに送信し、分散処理した結果を受信して、原画像に合成して表示する処理を計測し、比較した。今後、クラウドやエッジサーバによる個人ごとの識別や物体の詳細検証等の複雑な処理を担わせる狙いがある。図 2 に、実験環境の概要図を示す。送受信部分で OpenCV4.5 利用上でのコーディング部分の都合上と Docker for Desktop が GUI 環境を持たないため、X 端末に結果をリ

ダイレクトして画像をリアルタイムに送出する環境となった。それに伴い、評価環境では、通信による送受信に掛るターンアラウンドタイムをシミュレートし、計測を行った。ターンアラウンドタイムは、送受信した画像のサイズと通信プロトコルに掛る部分の一定数時間を送受信時間に必要な時間として、38 名が正面を向いて演奏しているオーケストラの映像を用いて 1 フレーム毎の処理時間の測定をした。通信プロトコルに掛る処理時間として、著者らが以前行った（詳細は[10]）、Cloud ネットワーク環境を用いた計算負荷分散処理の実験時に要した、通信プロトコルに掛る処理時間の平均値（1.28msec=ターンアラウンドタイム平均値 16.28msec - 処理時間 15.00msec）をシミュレートに用いた。この実験では、顔を切り出した部分にぼかしをかける画像処理に掛った時間を計測した。

### 4.3 評価結果

4.2 の評価方法でリバースプロキシを通していない場合と通した場合のシミュレートを行った。リバースプロキシは NjgnX をエッジサーバ内の閉じた Docker 上にて配置する方針で、シミュレートした。オーケストラが演奏している映像を用いて実験を行い、楽器パートごとに人物のグループを分けて、リバースプロキシにより、グループ毎に共通の処理を行い、選択的に映像加工と配信処理を並列化する条件下でのターンアラウンドタイムのシミュレートを行った。評価環境に対して、リバースプロキシは図 2 のエッジサーバとクライアント（X 端末）の間に入り、エッジサーバ内の Docker 上の VM として起動させ、グループ毎に加工した映像に対してポートを変えてサービスとして視聴者に映像を提供する設計である。オーケストラの様に、映像に映る人々の増減が無く、人々が一定位置から余り動かない状況の場合、処理の高速化が行えた。この度の実験ではキャッシュの効果のシミュレート実験は行えていない。リバースプロキシを通していない場合のターンアラウンドタイムを図 3 に示し、リバースプロキシを通したシミュレートを行った場合のターンアラウンドタイムのヒストグラムを図 4 に示す。いずれも 2000 フレーム中のターンアラウンドタイムのヒストグラムであり、図 3 では、各フレーム 0.8 ～ 1.6msec の処理時間となった。それに対して、図 4 では、グループ化され、共通の処理として処理が間引かれ、処理時間が 0msec となる場合があり、2000 フレーム全体を実行する処理時間が短くなったことが示されている。図 3 に用いたデータのターンアラウンドタイムの総計は、2667.8msec であり、図 4 に用いたデータのターンアラウンドタイムの総計は、1874.1msec となっている。特筆しなければならない結果として、表 1 に示した性能の機器では、OpenCV の特性で、38 名の顔画像認識を一度に行なえない事がわかった。少人数のグループが 1 フレーム中に時折出

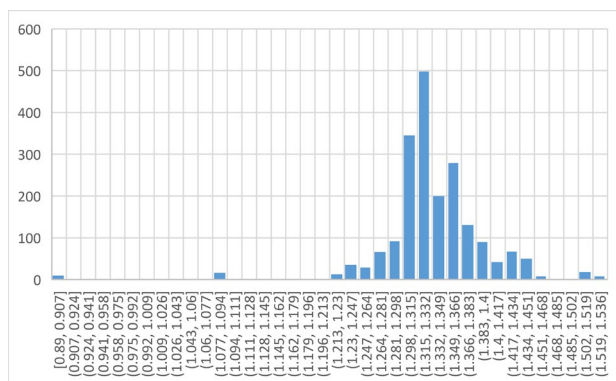


図 3. リバースプロキシを利用しない場合のターンアラウンドタイムのヒストグラム

来上がり，結果のグラフは，図 3. を縮小したような期待通りのグラフとはならなかった．今回の実験では，往來を人々が移動する実際のライブカメラの映像に近い状態ではシミュレートできていないが，図 3 と図 4，およびターンアラウンドタイムのシミュレート結果の比較により，時間短縮の効果は期待できる．

## 5. まとめ

本研究では，「エッジカメラサーバ連携」「ルール型映像利用方針記述」「共用映像加工認識」と呼ぶ革新的な技術を備えたプライバシー指向映像管理方式を提案する．提案に基づくライブカメラ配信のシステム設計・構築を用い，視聴用のクライアント端末を用いて，評価を行い，プライバシーを保てた状態で，処理速度を向上できたか確認した．キャッシュの効果は計れず，往來を人々が移動する映像ではないが，この度は「共用映像加工認識」技術の観点から，リバースプロキシによる選択的に共用映像加工認識をおこなった後の配信処理を行った結果，ターンアラウンドタイムの減少が行える事が期待できた．今後の課題として，「エッジカメラサーバ連携」の観点から，キャッシュデータの効率化により更なるターンアラウンドタイムの短縮と，移動する人々の実データによる計測を行う予定である．

**謝辞** 本研究の一部は，G-7 奨学財団研究開発助成事業および科学研究費補助金（18K11316 および 20K11829）の研究助成による．謹んで感謝の意を表する．

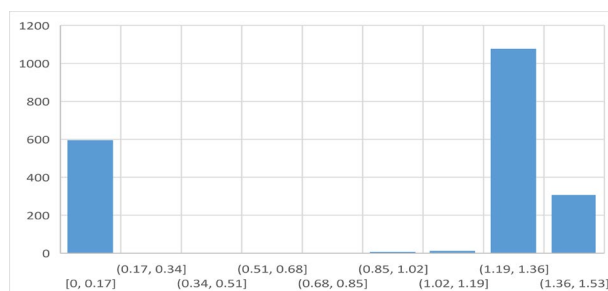


図 4. リバースプロキシを利用した場合のターンアラウンドタイムのヒストグラム

## 参考文献

- [1] 松本哲，義久智樹，川上朋也，寺西裕一，自律映像処理を伴うトラスト指向インターネットライブ配信システムの検討，(DICOMO2019) シンポジウム論文集，pp. 1298-1300, July 4th, 2019.
- [2] 松本 哲，義久智樹，川上朋也，寺西裕一，グラフ表現型画像処理プログラミングを用いたクラウド分散型インターネットライブ配信システム，マルチメディア，分散，協調とモバイル (DICOMO2020) シンポジウム論文集，pp. 1351-1355, June 26th, 2020.
- [3] Tomoki Yoshihisa, Satoru Matsumoto, Tomoya Kawakami, and Yuuichi Teranishi, "A Frame Rates Stabilization Scheme for Cloud Distributed Live Video Processing Systems," in Proc. of the 9th IEEE Global Conference on Consumer Electronics (GCCE 2020), <http://www.ieee-gcce.org/2020/>, (2021 年 5 月 9 日参照)
- [4] RedHat, “仮想化ハイパーバイザー”，<https://www.redhat.com/>, (2021 年 5 月 9 日参照)
- [5] Docker, “Docker.”, <https://www.docker.com/>. (2021 年 5 月 9 日参照)
- [6] Google. Inc, “Kuberntes,” <https://kubernetes.io>, (2021 年 5 月 9 日参照)
- [7] RedHat, “コンテナ・オーケストレーション”，<https://www.redhat.com/>, (2021 年 5 月 9 日参照)
- [8] J. Santos et al., “Towards Network-Aware Resource Provisioning in Kubernetes for Fog Computing applications,” IEEE Conf. Net. Softwarization (2019).
- [9] Donghui H, Chen F, Wu X, Zhao Z “A framework of privacy decision recommendation for image sharing in online social networks.” In: 2016 IEEE First international conference on data science in cyberspace. Changsha, pp 13–16 (2016).
- [10] 松本 哲，石 芳正，義久智樹，川上朋也，寺西 裕一，"全地球カメラを用いたクラウド分散型インターネットライブ放送システムの評価," (DICOMO 2018) シンポジウム論文集，pp. 523-529, July 4th, 2018.