

# 画像処理におけるエッジコンピューティングを用いた 垂直分散処理方式の検討

塩田純<sup>†1</sup> 滝澤允<sup>†1</sup> 田中裕之<sup>†1</sup> 高橋紀之<sup>†1</sup> 小林英嗣<sup>†1\*</sup>

## 概要:

広まり始めている IoT サービスの中で、カメラ映像を入力とする画像処理アプリケーションは、様々な分野での応用が期待されている。しかし、集中型のクラウド処理では、トラフィックの増加や通信遅延時間が大きくなると考えられる。本稿では、ユーザ近傍に計算資源(エッジサーバ)を配置するエッジコンピューティング技術を画像処理アプリケーションに適用する方式について検討を行った。検討した方式では、エッジサーバで処理コストの高い画像解析処理を行い、解析結果のみを端末に返却する。評価実験を行った結果、高性能の端末を用いる場合、アプリケーションの応答時間を保ち、トラフィックを抑えることができた。低性能の端末を用いる場合には、転送データをより削減する必要があることが判明した。

**キーワード:** エッジコンピューティング, 分散コンピューティング, 画像処理, Augment Reality, AR,

## A Study of Vertical Distributed Image Processing based on Edge Computing

JUN SHIODA<sup>†1</sup> MAKOTO TAKIZAWA<sup>†1</sup> HIROYUKI TANAKA<sup>†1</sup>  
NORIYUKI TAKAHASHI<sup>†1</sup> EIJI KOBAYASHI<sup>†1\*</sup>

## 1. はじめに

### 1.1 IoT サービスと画像処理

コンピュータやスマートフォンだけでなく、あらゆるモノがインターネットに繋がり相互に通信を行う IoT (Internet of Things) に注目が集まっている。インターネットに繋がるモノからは、その種類に応じて様々なデータを取得することができる。中でも、カメラから取得する動画への期待が高まっている。一般的にモノからは、特定のデータしか取得することができない。例えば、温度計からは気温データ、赤外線センサからは距離データ、などとなる。一方、カメラで撮影した動画を画像処理することにより、顔認証や物体認識、異常物検出、色判定など様々なアウトプットを得ることができる。応用範囲の広い動画を利用した IoT アプリケーションの研究開発が盛んに行われている。

例えば、M. Satyanarayanan らは、Google Glass とクラウドを連携させた AR (Augmented Reality; 拡張現実) のシステムの研究をしている[1]。Google Glass のカメラで取得した動画を、クラウド上の画像認識エンジンで解析する。その解析結果を Google Glass を通してユーザが見ている現映像に照らし合わせて提示することで、ユーザの作業支援を行うことができる。

交通分野では、自動車の車載カメラや信号機に設置した路側カメラで取得した動画から、自動車外部や周辺の状況を解析することで、安全性の向上や自動運転支援の実現を図っている[2][3]。

製造業では、機械や生産ラインに設置したカメラの動画を利用して、作業の自動化を図り、省力化や作業精度の向上を目指している[4]。

### 1.2 IoT サービスにおける画像処理の課題

今後、IoT サービスは、モノから集めたデータに基づいて、ユーザへのフィードバックやモノの制御をリアルタイムに行うかたちへと発展していくと考えられる。現在、多くの IoT アプリケーションは、データセンタを利用した集中型処理基盤でのクラウドサービスとして構想されている。しかし、集中型処理では、1.1 で示したような画像を入力とするアプリケーションの要件を満たすのは困難な場合が生じると考えられる。

課題の一つは、トラフィックの増加である。文献[5]では、インターネットに接続させるモノは、2015年に約250億台、2020年までに500億台になると予測している。また文献[6]では、トラフィック量は、2019年に2016年の2倍程度になり、以降指数関数的に増加し、その8割を動画が占めると予想している。これら文献でのトラフィック量は、動画配信サービスを含むものであるが、1.1のような画像処理アプリケーションのための動画のアップロードの増加も見込まれている。集中型処理のクラウドでは、モノから取得した膨大な画像データを収集・計算すると、通信帯域とク

<sup>†1</sup> 日本電信電話株式会社  
NIPPON TELEGRAPH AND TELEPHONE CORPORATION  
\*現在は東日本電信電話株式会社に所属。

クラウド上のサーバ処理性能が圧迫され、アプリケーションの応答時間（サーバ上の処理時間と通信遅延時間を合算したものの）が増加してしまう。

もう一つの課題は、端末とサーバ間の通信遅延時間（処理時間を含まないものとする。）である。クラウドコンピューティングでは、端末（スマートフォン、タブレット、カメラなど）とクラウド上のサーバが通信を行っている。端末とクラウド間の往復の通信遅延時間は、平均で数十msとなり、画像処理アプリケーションの要件を満たすことができないユースケースもある。

これらの課題への解決策として、ユーザ近傍に計算資源を配置するエッジコンピューティングが注目を浴びている[7][8][9]。例えば、文献[7]では、クラウドと端末の間に小規模なデータセンタ Cloudlet を設けて、ARアプリケーションの処理を分散配置することで通信遅延時間を低減する方式が検討されている。

本稿では、エッジコンピューティングを活用したARアプリケーションの分散処理方式を検討する。エッジコンピューティングをARアプリケーションに適用することによって、トラフィック及びアプリケーションの応答時間の削減を目指す。

2章ではエッジコンピューティングの概要とシステム構成を紹介する。3章では、ARアプリケーションへの適用方針を述べる。エッジコンピューティングをARアプリケーションに適用する意義を説明し、具体的な要求条件を定めるためにARアプリケーションのユースケースとしてメイクのサポートを取り上げ、その意義とサービスイメージ、システム構成と課題を述べる。4章では、分散処理方式を考案し、考案方式の評価実験の結果と考察をまとめた。5章には、まとめと今後の課題を述べる。

## 2. エッジコンピューティング

### 2.1 概要

エッジコンピューティングとは、ネットワーク上の端末の近傍に計算資源（エッジサーバ）を配置する概念のことを言う。一連のアプリケーション処理を機能ごとのモジュールに分割し、端末、エッジサーバ、クラウドそれぞれに割り当て（垂直分散）て処理する。これによって、端末とエッジサーバの往復通信遅延時間を数ms程度にすることができ、以下の4つのメリットが生じる（図1）。

1つ目は、リアルタイム・アプリケーションの実現である。遠隔にあるクラウドのサーバと比較して近傍で処理を行うため、往復の通信遅延時間が低減され、ユーザの体感する応答性や操作性を向上させることができる。2つ目は、アプリケーション処理の高速化である。これまで端末が行う処理の一部を、エッジサーバにオフロードすることによって、端末の処理負荷を減らし、アプリケーションの応答時間を低減することができる。また、端末の消費電力

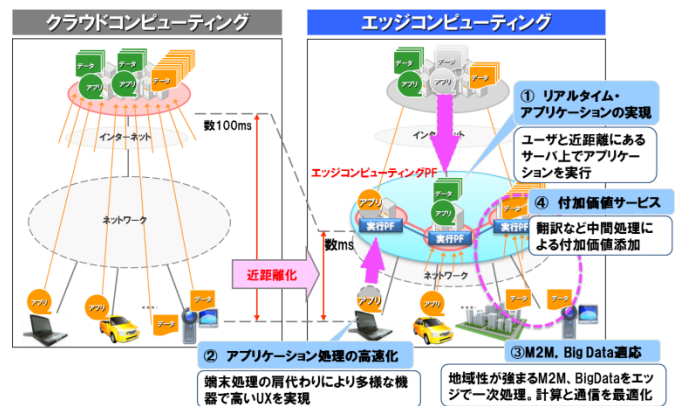


図1 エッジコンピューティング概要

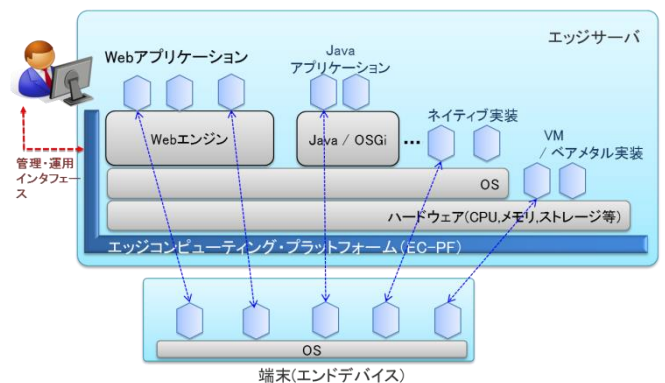


図2 エッジコンピューティングのシステム構成

を抑えることができる。3つ目は、M2M, BigData への適用である。地域性の高いM2MやBigDataのアプリケーションにおいて、大量のセンサから取得したデータを、遠隔のサーバでなくエッジサーバで収集、解析し、必要なデータのみを遠隔のサーバに送る。これによって、トラフィックを削減することができる。4つ目は、付加価値サービスである。エッジサーバの中間処理によって、ソフトウェアのアップデートやユーザのインターフェース拡張など付加価値サービスを提供することができる。

### 2.2 エッジコンピューティングのシステム構成

エッジコンピューティングのシステム構成を図2で示す。エッジサーバでは、アプリケーションの開発や運用を効率的に行うために、計算資源の仮想化技術やソフトウェアのコンポーネント化などのクラウド技術を活用できることが望ましい。また、リアルタイム処理を重視し、オーバーヘッドが少ないアプリケーション実行環境や、端末の処理負荷を抑えて電力消費を低減するアプリケーションの実行環境など、エッジコンピューティングを活かした実行環境を提供する。エッジサーバでは、VM単位やプロセス単位まで様々な粒度での実行環境を提供する必要がある。また、端末の位置、実行アプリケーションの要件に合わせて、適切なエッジサーバと実行形態を割り当てて、管理・運用をするプラットフォーム機能も必要となる。

これまでに我々はエッジコンピューティングを活用したWebアプリ実行エンジンを開発し、実用化をしている

[10][11]が、これらのエッジコンピューティングの特徴は、比較的大きな演算処理を必要とする画像処理アプリケーション、中でも双方向性を持つARアプリケーションにおいて有効と考える。

### 3. ARアプリケーションへの適用方針

#### 3.1 エッジコンピューティングをARアプリケーションに適用する意義

Digi-Capitalの2015年の調査では、ARの市場規模は、2020年に1200億ドルに達するとも言われており[12]、ARの新興企業のMagic Leapは、GoogleやQualcomm等から5億ドル以上の出資を受けている[13]。また、2015年1月に、Microsoftは、マーカレス型のARを用いたメガネ型ホログラムコンピュータHoloLensを、2014年、Googleが3Dスキャンを搭載した開発者向けのスマートフォンProject Tangoを発表した[14][15]。ARアプリケーションはユーザーの行動や作業を支援することに利用され、製造業、教育業など幅広い分野での応用が期待されている。

ARアプリケーションは、様々な情報を適切なタイミングで提示することが重要である。つまり、多種多様なデータを高速に処理することが求められており、1.2で述べたトラフィックの増加と通信遅延時間の大きいことが課題となる。ARアプリケーションに垂直分散方式を適用することによって、2つの課題を解決できると考える。

1つ目のトラフィックの増加の課題については、アプリケーション処理の分割方法と分割した処理の配置の場所によって、転送データ量を削減することができる(図3)。具体的には、エッジサーバの高い処理能力を活かし、エッジサーバ上で処理コストの高い画像解析処理を行うようにする。従来のARアプリケーション(図3左側)では、端末がクラウドからコンテンツをダウンロードして、端末で取得した画像を解析して表示する方式がほとんどであった。エッジコンピューティングを適用することで、エッジサーバがクラウドからコンテンツ取得を行う。端末からエッジサーバへのデータ転送では、動画をそのまま送信するのではなく、前処理したデータ(特徴量、輝度、必要な個所にトリミングした画像データなど)を送信する。エッジサーバでは、端末から受け取った動画を、クラウドから取得したコンテンツを元に解析を行う。エッジサーバから端末へのデータ転送では、解析結果のみを送信する。また、コンテンツの更新などクラウドと連携する場合は、エッジサーバ上で1次処理をして必要なデータのみを送信する。これらにより、転送データ量を削減するとともに、端末の処理コストを抑えて高速なアプリケーション処理を実現することができる。

通信遅延時間が大きいという課題に対しては、エッジサーバの配置する場所を、端末近傍に配備することで低減することができる。アプリケーションの要件に応じて、エッ

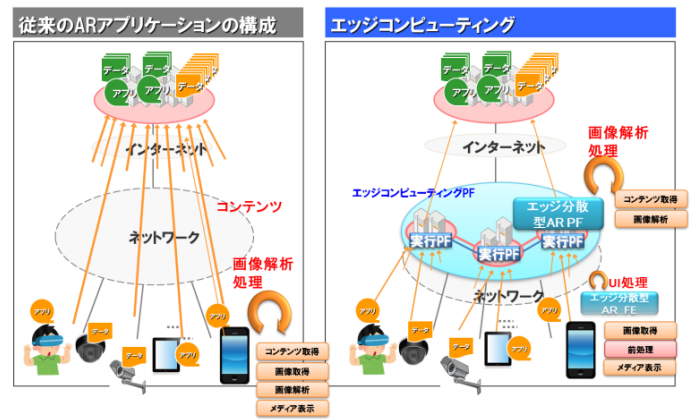


図3 エッジコンピューティングによるトラフィック削減

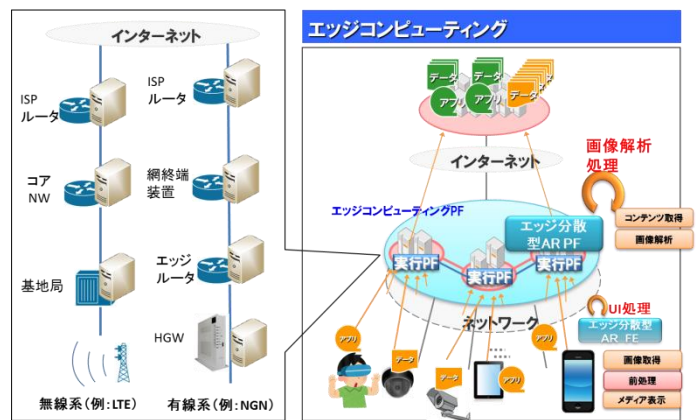


図4 エッジコンピューティングによる通信遅延時間の削減

ジサーバの配置場所の候補が考えられる(図4)。例えば、LTEなどの無線系の場合は、基地局やコアネットワークルータ、ISPルータ、NGNがあり、有線系の場合は、ホームゲートウェイ、エッジルータ、網終端装置、ISPルータがある。ARアプリケーションの要求条件やターゲット、ユーザー数、設備や運用のコストに応じて、エッジサーバの配置場所を決定する。

#### 3.2 ARアプリケーションのユースケース:メイクサポート

ARアプリケーションのユースケースとして、メイクサポート(以下、ARメイクアプリケーションとする。)に着目する。メイクは数mm単位の位置や色の差で大きく印象が変わる。応答時間が大きくなると、メイクする領域をユーザーの撮影映像に重畳する場合、本来あるべき表示位置とずれて、ユーザーは数mmずれた箇所にメイクをしてしまい、理想とするメイクを実現できなくなってしまう。また、ユーザーが実施したメイクをリアルタイムに評価してアドバイスを提示する場合、ユーザーが誤った位置や色でメイクしていても、そのミスに気付くのが遅くなり、メイクの修正範囲が大きくなってしまう。

このように高い応答性が求められるARメイクアプリケーションを取り上げ、具体的な要件を定めて、垂直分散方式を検討する。

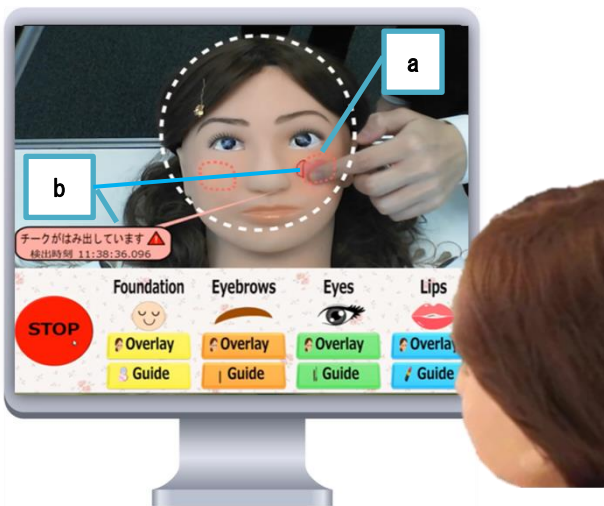


図 5 ARメイクサポートのイメージ図  
 (a:ガイド表示, b: 位置に関するフィードバック)

### 3.3 アプリケーションイメージ

本稿で想定するアプリケーションのイメージは以下の通りである。

- ①鏡を模擬した端末の前に立つと、センサからは肌状態などのデータ、カメラの映像からは顔の特徴などのデータを取得する。
- ②センサやカメラから取得したデータと、クラウド上にある事前に登録した年齢やスケジュール、トレンド情報などを用いて、ユーザに合うメイク案を提示する。
- ③ユーザが提示された案から1つを選択する。
- ④選択したメイクが鏡に映っているユーザの顔の上に重畳表示される。
- ⑤ユーザがそれを気に入れば実際にメイクをどのようにしていくか指示が表示される。

ARメイクアプリケーションは、隣にメイクアップアーティストがいるかのように、ユーザのメイク操作に対してリアルタイムかつ事細かにアドバイスなどのフィードバックを提示できることが理想である。①から⑤の中でも高い応答性が求められる⑤に着目し、図5には頬のメイク方法を提示するときのサービスイメージを示す(実際に実装したものを動かしているときの写真が図5である。)

頬のメイクを例に、ARメイクアプリケーションの特徴を以下に示す。

- a) ガイド表示: ユーザが選択した頬を実現するためのメイクすべき領域を表示する。また、頬以外はメイク完成後の状態を表示することで全体のバランスを意識させる。
- b) 位置に関するフィードバック: ユーザの塗っているものがガイドに沿っているかリアルタイムに評価し、評価結果や修正方法などを提示する。
- c) 形と濃さに関するフィードバック (図5には表記されていない): ユーザが塗ったある範囲における眉毛の形や濃

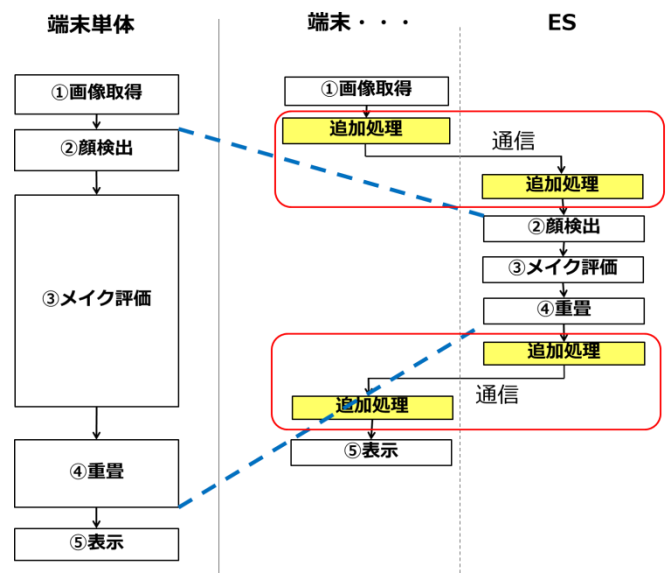


図 6 エッジコンピューティングを  
 ARメイクアプリケーションに適用する場合の構成

さについて、ある程度塗るごとに段階的に評価し、評価結果や修正方法などを表示する。

d) メイク道具の使い方に関するフィードバック (図5には表記されていない): ユーザが頬を塗る道具(チーク, ブラシ, パフ)の使い方や持ち方(角度等)についてアドバイスを表示する。

### 3.4 システム構成

垂直分散による最大限の効果を得るためには、端末とエッジサーバ間における通信のオーバーヘッドを抑える必要がある。エッジコンピューティングをARメイクアプリケーション(3.3の(b))に適用する場合の構成を図6に示す。

図6の①~⑥の処理概要を以下に示す。

- ① 画像取得: 端末において、カメラから1フレーム分の画像データを読み込む。
- ② 顔検出: 1フレームの画像データにおいて、顔の輪郭やパーツの特徴点座標を抽出し、顔パーツごとにグルーピングする。
- ③ メイク評価: パーツごとに、取得した画像データと前フレームの画像データにおいて、1ピクセルごとと比較し色差判定を行う。色差データの差がある値以上であり、かつ、一定個数以上隣接する座標がある場合、その座標群とそれに関する色差を差分とする。画像データにおいて、差分の領域を示す輪郭を抽出する。抽出した輪郭の座標を赤色にした画像データを生成する。
- ④ 重畳: 元の画像データに対して、②顔検出で求めた特徴点座標をもとに輪郭画像データを重畳させる。
- ⑤ 表示: 重畳させた画像データを端末の画面上に表示する。

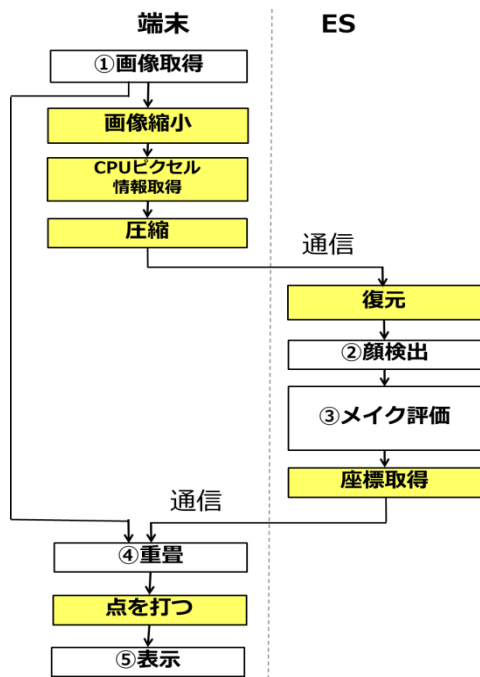


図 7 考案方式

図 6 に端末単体の場合 (左側), 端末とエッジサーバの垂直分散に適用させた一例 (右側) を示す. 垂直分散を適用させた場合, エッジサーバの高い処理能力を活用することで, 図 6 中の処理②から④を高速化することができる (図 6 の太点線). 一方, 端末とエッジサーバ間でデータ転送が必要となり, 追加処理の時間や通信遅延時間といったオーバーヘッドが生じる (図 6 の 2 つの枠部分). ここで, 追加処理とは, データ転送するための転送データ形式の変更処理や圧縮処理である. 通信遅延時間は, エッジサーバと端末の物理的な距離に依存し, 適切な位置にエッジサーバを配置することで低減することができる. 従って, 今回は, 追加処理によるオーバーヘッドを最小限に抑えることに注力した.

## 4. 実装と評価

### 4.1 考案方式

考案方式は, 端末では多くのライブラリで提供されている一般的な画像処理を, エッジサーバでは処理コストの高い画像解析を行うことで, 追加処理による処理時間の増加を低減した方式である. 図 7 を用いて説明する. 追加処理は塗りつぶしで示す.

端末からエッジサーバへのアップロードデータを削減するために, 処理①で取得した画像データを縮小し, 圧縮処理を施した. 画像縮小のための画像処理には, CPU 内蔵の GPU を用いることで高速化を図っている. 縮小した画像データの CPU のピクセル情報を取得して, 画像処理ライブラリを用いて JPEG 方式に圧縮処理を施した. 端末からエッジサーバに圧縮処理を施した画像データを転送する. このとき, 端末とエッジサーバ間の通信プロトコルには,

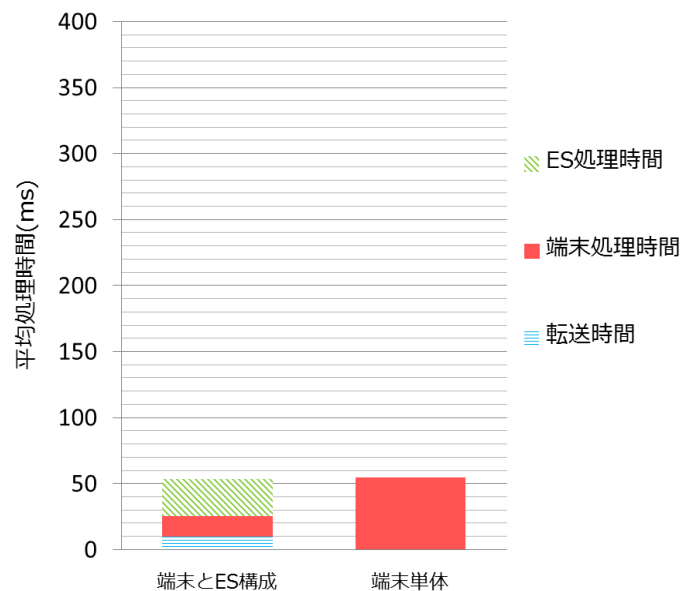


図 8 測定結果(端末性能が高い場合)

WebSocket を用いた.

エッジサーバでは, 圧縮した画像データを受け取り, 処理②ができる画像形式に復元し, 処理②, ③を行う. 処理③の解析結果の輪郭画像データにおいて, 輪郭の座標を抽出する. 端末は, 抽出した座標のみをエッジサーバに送信する. このとき, アップロードと同様に通信プロトコルには WebSocket を用いた.

端末は, ①で取得した元の画像データに, エッジサーバから受けとった座標の点を打ち, 表示する.

### 4.2 評価実験

4.1 で示した考案方式において, 2 つの環境で処理時間を測定した. その結果について考察する.

#### 4.2.1 測定環境

端末が高性能の場合と低性能の場合で測定を行った. 高性能端末には, カメラが搭載されたノート PC1 台を用いた. 端末の CPU は Intel (R) Core (TM) i7-6700K, メモリは 16GB, GPU (CPU 内蔵) は Intel (R) HD Graphics 530 である. エッジサーバには, CPU が Intel (R) Core (TM) i7-2600K, メモリが 24GB, また GPU の NVIDIA Quadro M3000M が搭載させたデスクトップ PC を利用した. 低性能端末には, スティック型 PC を用いて, USB カメラを接続した. スティック型 PC の CPU は, Intel (R) Atom (TM) プロセッサ Z3735F (Bay Trail), メモリは 2GB, GPU (CPU 内蔵) は intel (R) HD Graphics である. エッジサーバは, 端末が高性能のときと同様のデスクトップ PC を用いた.

また, 端末とエッジサーバを同一 LAN 内で接続したため, 通信遅延はほぼない.

#### 4.2.2 測定結果と考察

考案方式と分散によるオーバーヘッドがない端末単体の場合を測定した. その結果を図 8 に示す.

ノート PC とエッジサーバ構成の合計処理時間は 54.0ms,

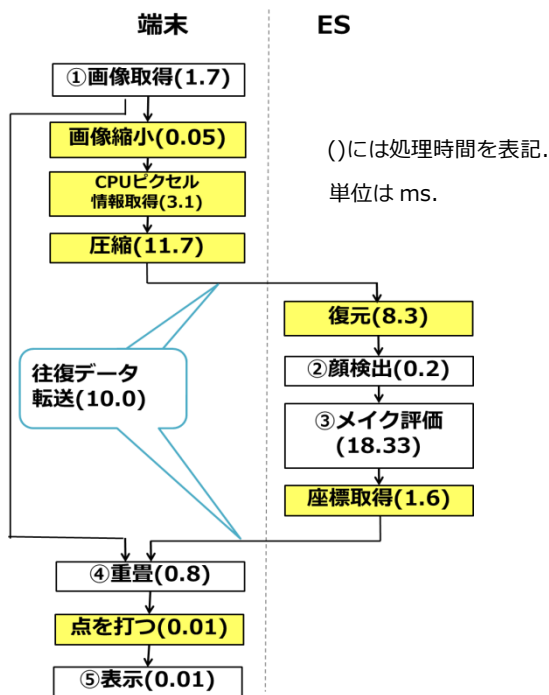


図 9 各処理時間の測定結果(端末性能が高い場合)

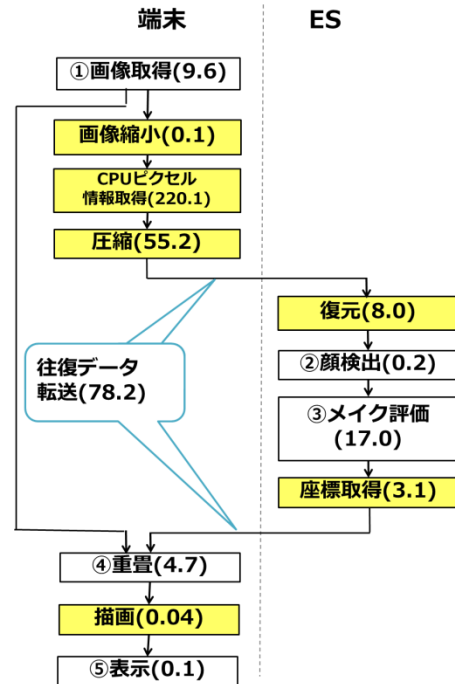


図 11 各処理時間の測定結果(端末性能が低い場合)

ノート PC 単体合計処理時間は 55.1ms, となり, ほぼ同等の値になった. 通常, AR アプリケーションの応答時間が 60ms 以内であれば, 人は違和感を覚えないと言われている [16]ため, 許容範囲であると考えられる. また, エッジサーバへのオフロードにより端末での処理量 (処理時間) は 28.4%に削減されている. この理由は, 端末での処理を極力減らし, エッジサーバに処理コストの高い画像解析処理を行うように分散配置したためと考えられる.

また, ノート PC とエッジサーバ構成の各処理時間の測定結果を図 9 に示す. エッジサーバに処理を移すために生じる追加処理を塗りつぶしで示す. 画像データの圧縮・復元にかかる処理時間が新たに生じ, 合わせて 20.0ms となった. 1Gbps の回線を用いる場合, 非圧縮 (約 2.76MB) の画像デー

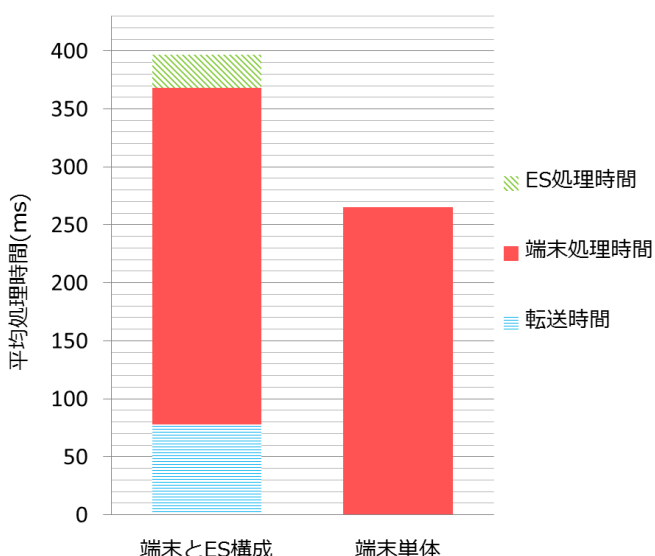


図 10 測定結果(端末性能が低い場合)

タを転送すると 20ms かかると算出できる. しかし, 無線アクセスのようなさらに狭い帯域を用いる場合, 考案方式のように圧縮することでトータルの処理時間を削減することが可能である.

エッジサーバの計算結果であるオーバーレイ画像データでは, 座標のみを送信することによって, 転送データ量と端末処理量の軽減を図っている.

次に, 端末が低性能である場合の測定結果を図 10 に示す. スティック型 PC とエッジサーバ構成の合計処理時間は, 396.8ms となり, アプリケーションの応答時間の許容範囲を超えてしまった. スティック型 PC 単体の合計処理時間は, 265.6ms となり, スティック型 PC の場合に処理時間が 1.5 倍程度大きいことがわかる. 今回の実験で使用したスティック型 PC の性能は, ノート PC の性能の 0.78 倍程度であり [17], 端末性能を下げることによって大幅に処理時間が大きくなることがわかった.

また, スティック型 PC とエッジサーバ構成の場合の各処理時間の測定結果を図 11 に示す. スティック型 PC の各処理時間は, ノート PC の各処理時間と比較して, 2~70 倍大きくなることがわかる. 特に, CPU ピクセル情報の取得, 圧縮処理時間の増加が大きい, これらは使用した画像処理ライブラリに依存するため, 改善の見込みがある. また, 往復のデータ転送時間は, ノート PC のときと比較して 8 倍程度大きくなっている. さらに転送データ量の軽量化を図ることが必要であることがわかった.

## 5. まとめと今後の課題

画像処理アプリケーションに垂直分散方式を適用する

ことで、トラフィックの削減及び応答時間の高速化を図る方式を検討した。画像処理アプリケーションの一例として、ユーザのメイクをリアルタイムに評価するメイクサポートを取り上げた。アプリケーション処理の分割箇所やデータ転送方式によって、垂直分散における端末やエッジサーバの処理時間(処理量)の制御を実現した。有効であった手法は、エッジサーバで解析した結果を画像データとしてではなく座標に変換した方式であった。ただし、端末性能が低い場合、端末での処理時間及びデータ転送時間が大きくなるため、改善する必要がある。

今後の課題は、端末性能が低い場合の処理の高速化方式の検討である。複数の端末性能において、評価実験を行い、端末性能や端末およびエッジサーバ間の通信遅延と、エッジサーバの効果の度合いとの関連を定量的に示すことを目指す。

## 文 献

- [1] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence (invited paper)," Proceedings of MobiCASE 2014: Sixth International Conference on Mobile Computing, Applications and Services, Austin, TX, November 2014.
- [2] Chenyi Chen, Ari Sef, Alain Kornhauser, Jianxiong Xiao, "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving," The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2722-2730.
- [3] Alex M. Kaneko, Kenjiro Yamamoto, "Landmark recognition based on image characterization by segmentation points for autonomous driving," Control Systems (ISCS), 2016 SICE International Symposium.
- [4] Sang Kwon, Jong-hak Kim, Zhongyun Yuan Sim, "Robust-rotation recognition based on contour matching using CUDA in automation system," Consumer Electronics (ISCE), 2015 IEEE International Symposium.
- [5] "Gartner プレスリリース, Gartner Says 4.9 Billion Connected "Things" Will Be in Use in 2015," <http://www.gartner.com/newsroom/id/2905717>(参照 2014-11-11).
- [6] "2015 Cisco VNI Complete Forecast Update: Key Trends Include Mobility, M2M and Multimedia Content" ,<http://blogs.cisco.com/sp/2015-cisco-vni-complete-forecast-update-key-trends-include-mobility-m2m-and-multimedia-content>(参照 2016-8-15).
- [7] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," IEEE Pervasive Computing, Vol. 8, No. 4, pp. 14-23, Oct.-Dec. 2009.
- [8] "Mobile-edge computing - introductory technical white paper," Mobile-Edge Computing (MEC) ISG, European Telecommunications Standards Institute (ETSI), Sept. 2014. (available online) .
- [9] Noriyuki Takahashi, Hiroyuki Tanaka, Ryutaro Kawamura, "Analysis of process assignment in multi-tier mobile cloud computing and application to Edge Accelerated Web Browsing," IEEE Mobile Cloud 2015, San Francisco, U.S.A., March-April 2015.
- [10] "高レスポンスやビッグデータ処理が要求される新たなアプリケーションの開拓を推進する「エッジコンピューティング構想」を策定," <http://www.ntt.co.jp/news2014/1401/140123a.html>(参照 2016-10-20).
- [11] "「ひかり TV」のインターネットブラウザがクラウドに対応~テレビ画面上のインターネット表示スピードが約 2 倍に~NTT が推進するエッジコンピューティング構想の新技术が商用サービスで初採用~," <http://www.ntt.co.jp/news2015/1508/150827a.html>(参照 2016-10-20).
- [12] "Digi-capital Blog," [http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.VX4wd\\_ntlBc](http://www.digi-capital.com/news/2015/04/augmentedvirtual-reality-to-hit-150-billion-disrupting-mobile-by-2020/#.VX4wd_ntlBc)(参照 2015-6-1).
- [13] "Magic Leap PRESS 2014/10/21," <http://www.magicleap.com/#/press>(参照 2015-6-1).
- [14] "Microsoft HoloLens," <https://www.microsoft.com/microsoft-hololens/en-us>(参照 2015-6-1).
- [15] "Project Tango," <https://google.com/atap/project-tango/>(参照 2015-6-1).
- [16] ADELSTEIN, B.D., LEE, T.G., ELLIS, S.R. 2003. Head tracking latency in Virtual Environments: Psychophysics and a model. In Proc. of the 47th Annual Meeting Human Factors and Ergonomics Society, 2083-2087.
- [17] "CPU 性能表 Windows エクスプレィエンスインデックス一覧," <http://www.sokupochi.com/cpu-wei/>(参照 2016-5-20).