

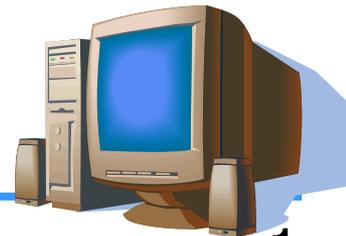
2019年度 第4回 SWIM研究会

フォグコンピューティングにおける センサデータ管理方式の提案

2020年 2月 21日

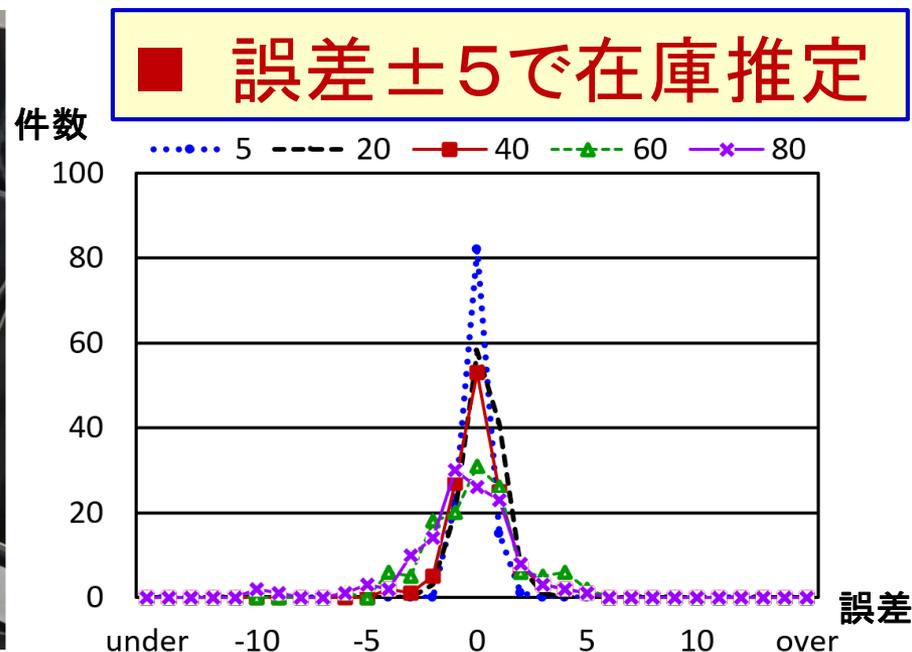
静岡理工科大学
情報学部 情報デザイン学科

工藤 司



背景と本研究の狙い

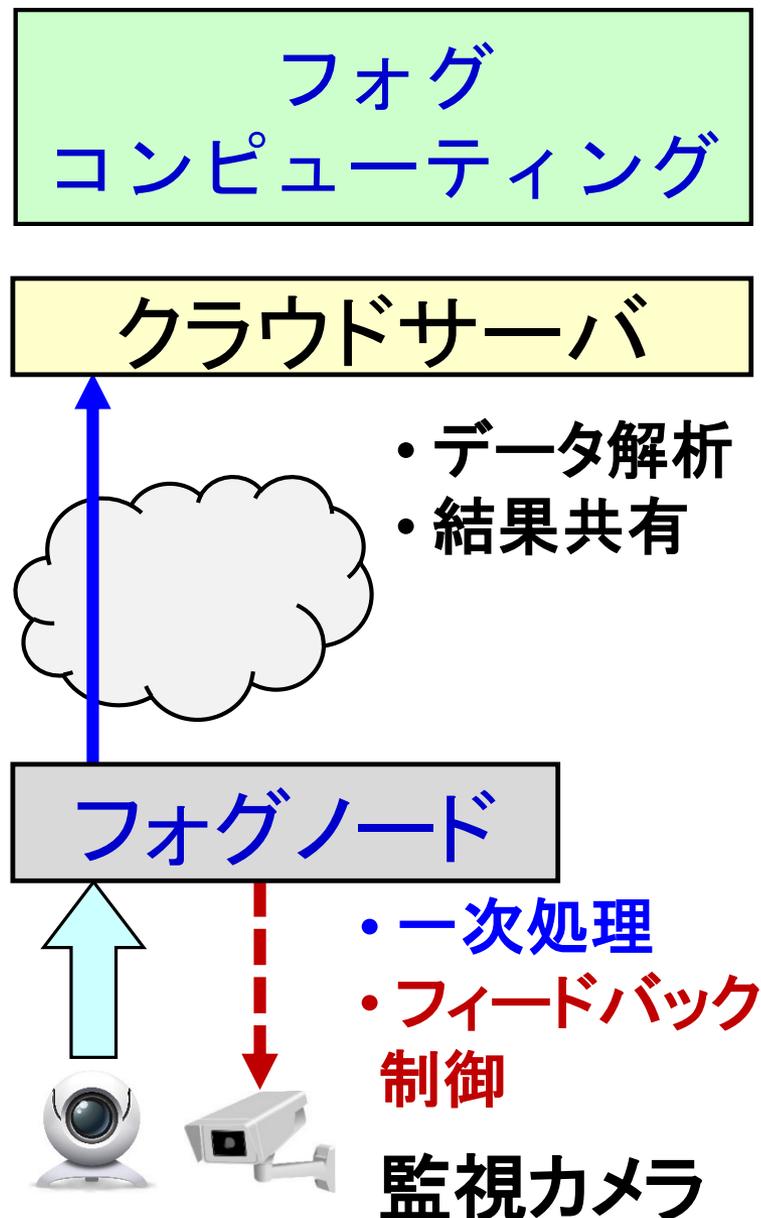
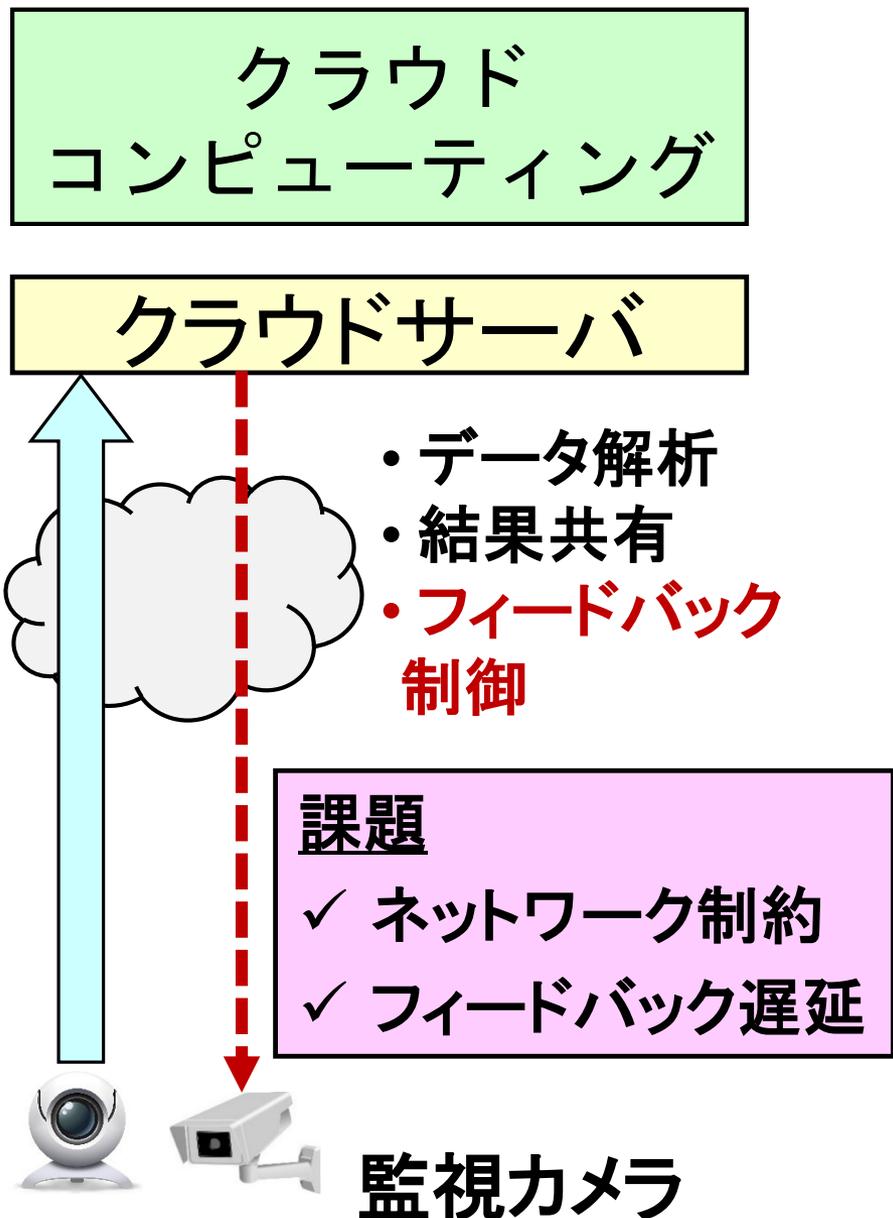
- 背景：目視で把握困難なバルクコンテナの在庫管理
 - ✓ 深層学習を活用，画像から実用的な精度で推定



背景と本研究の狙い

- 背景：目視で把握困難なバルクコンテナの在庫管理
 - ✓ 深層学習を活用，画像から実用的な精度で推定
- 本研究の狙い：作業者に負荷をかけずに，多数のコンテナの画像を収集する方式の提案
 - ✓ 方法：ウェアラブルカメラで常時動画を撮影，コンテナを自動判別
 - ✓ 課題：自動判別漏れ・在庫数異常の場合の対応
 - ⇒ 過去に遡った膨大な動画検索が発生
 - ✓ この課題の対策と，ウェアラブルカメラで画像収集が可能であることを示す。

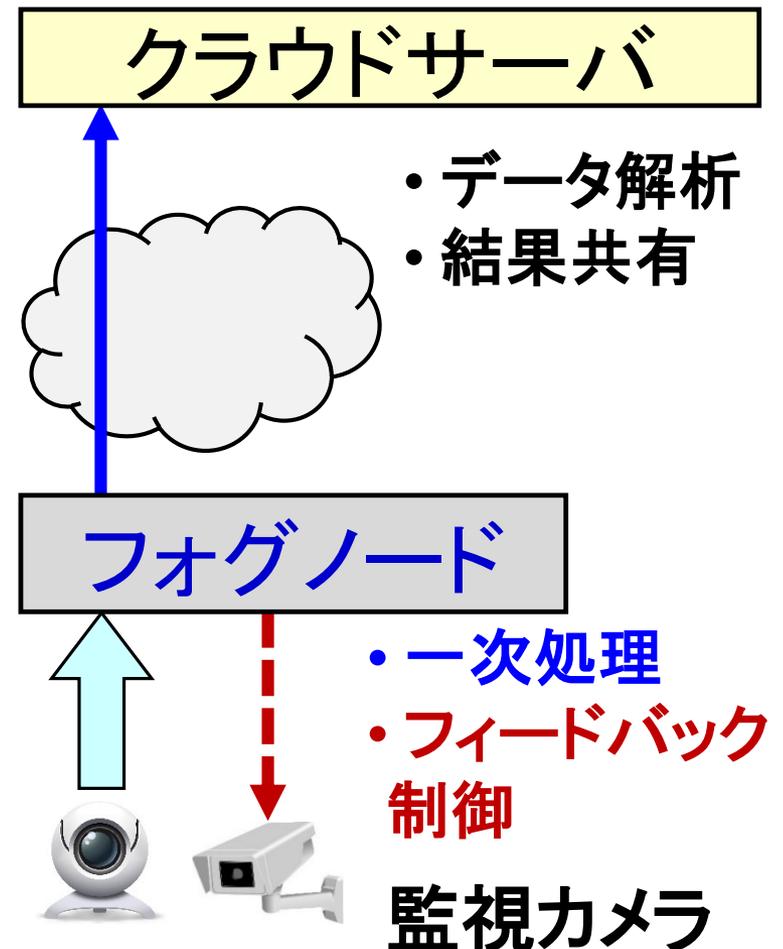
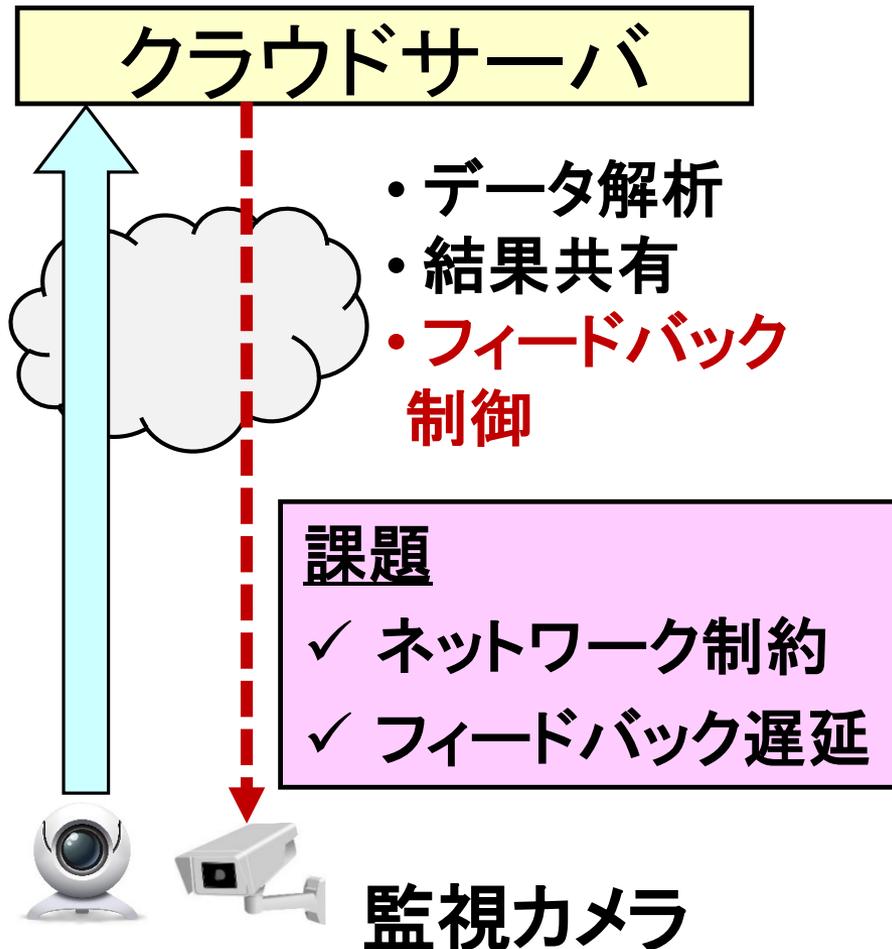
フォグ (エッジ) コンピューティング



フォグ(エッジ)コンピューティング

■ 事例：監視カメラによる入退室管理

- ✓ 入退室のあった時刻，画像，人数を抽出・転送

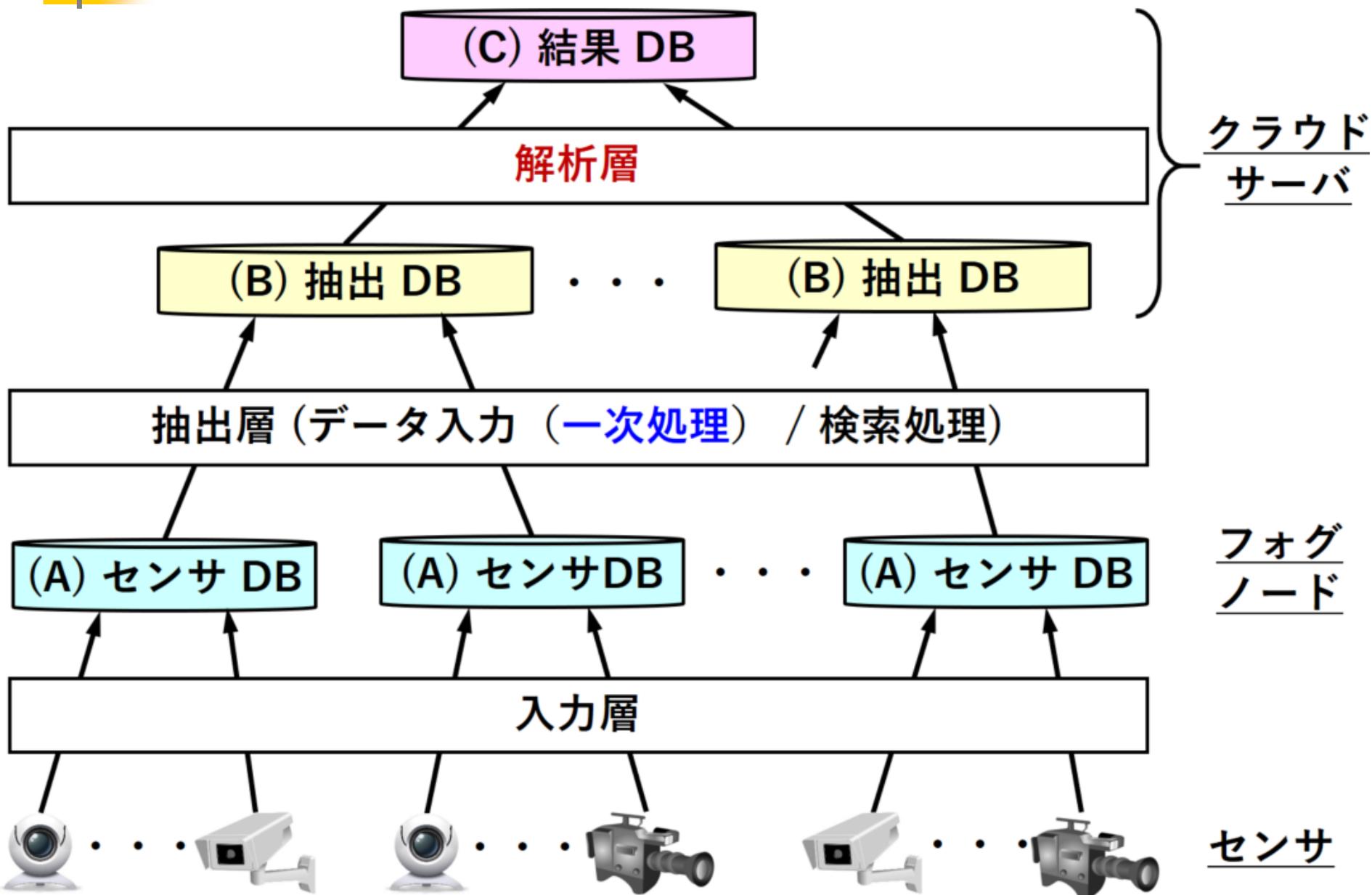


- 一次処理結果のみクラウドサーバへ転送
 - サーバの解析処理でオリジナルデータが必要な場合に欠損が発生

- 複数の監視カメラによる入退室の自動監視
 - 動きのあった部分の画像のみがサーバへ転送
(動き: 設定された閾値以上の変動の検知)

 - 事件・事故発生の場合は, 周辺の画像を解析.
 - ✓ 前後の状況
 - ✓ 関連するカメラの状況
(閾値以下の変動のビデオ解析)

3階層データモデル (関連研究)



3階層データモデル（関連研究）

- 構成: フォグノードに分散データベースを設置
 - オリジナルデータを保存,
 - サーバから必要なデータを検索可能にする

抽出層 (データ入力 (一次処理) / 検索処理)

(A) センサ DB

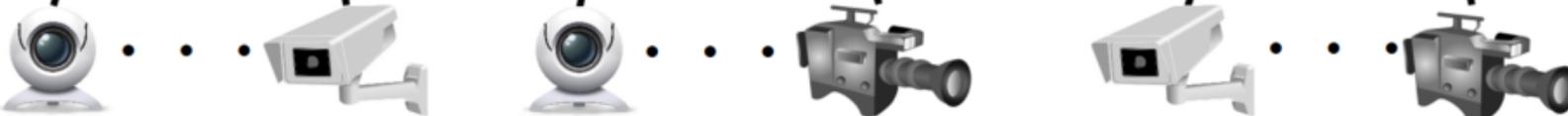
(A) センサ DB

...

(A) センサ DB

フォグ
ノード

入力層



センサ

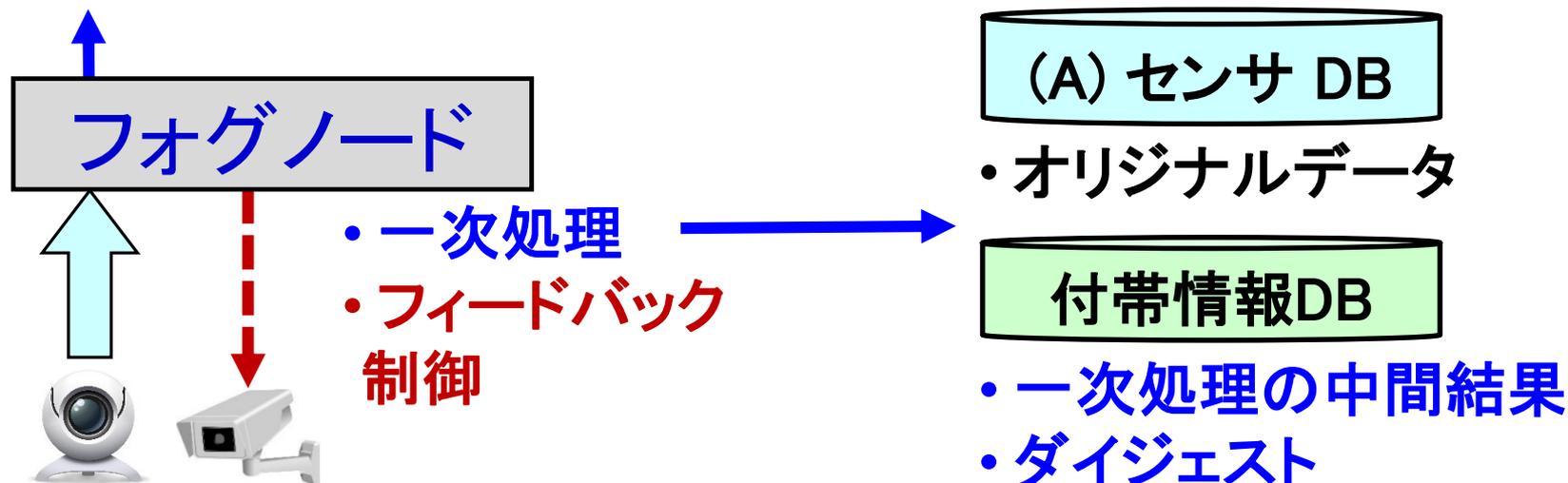
本研究の解決しようとする課題

- 前提: バルクコンテナの画像による在庫推定
 - ✓ ウェアラブルカメラ動画からコンテナ画像を自動判別
 - ✓ 自動判別したコンテナ画像から, 在庫数を自動推定
 - ✓ 自動判別漏れ・在庫数異常の場合は, 動画を再検索
- 課題: 長時間の動画からの再検索
 - ✓ 膨大な動画データの中から対象画像を検索
 - ⇒ 検索効率の低下

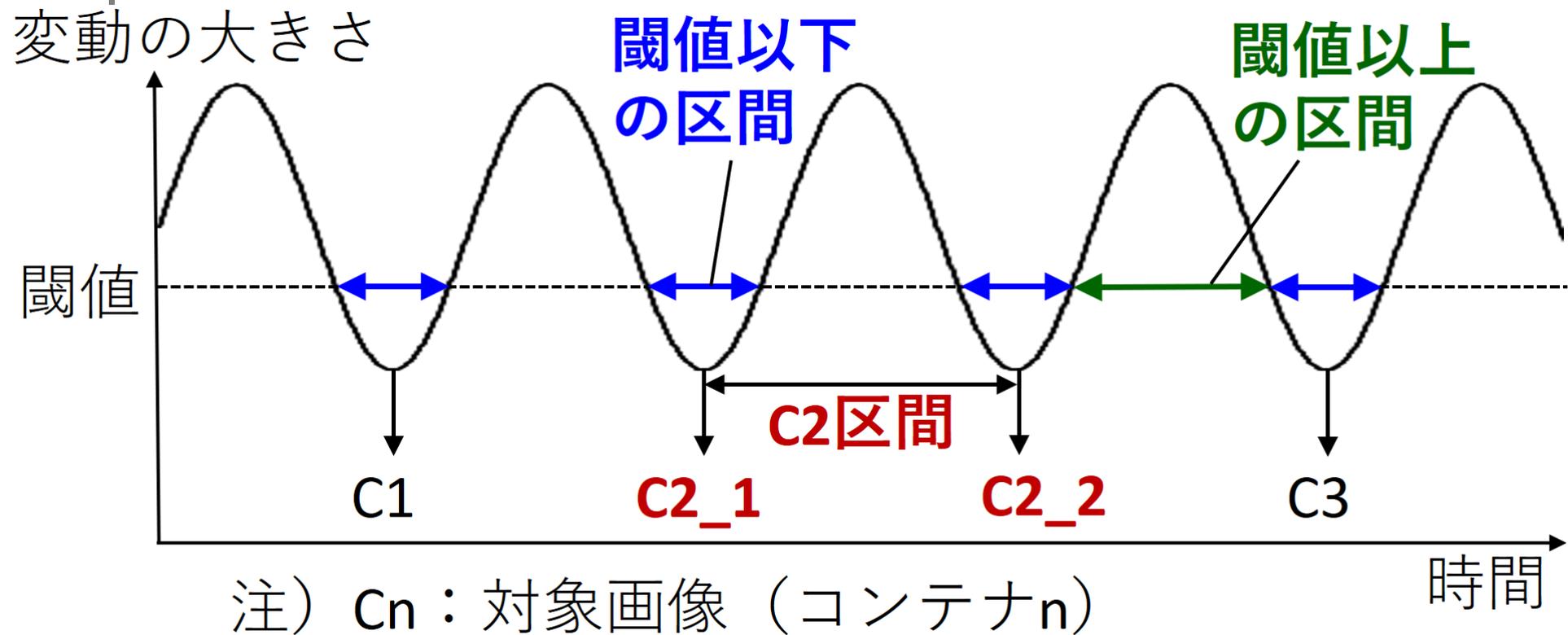
提案

- **一次処理**の過程で、以下のデータを保存、検索対象部分の絞込みを可能にする。
 - オリジナルデータ（従来方式で保存）
 - **一次処理の中間結果**: 変化点, 特徴点など
 - 対象部分を絞り込むための**ダイジェスト**

クラウドサーバ

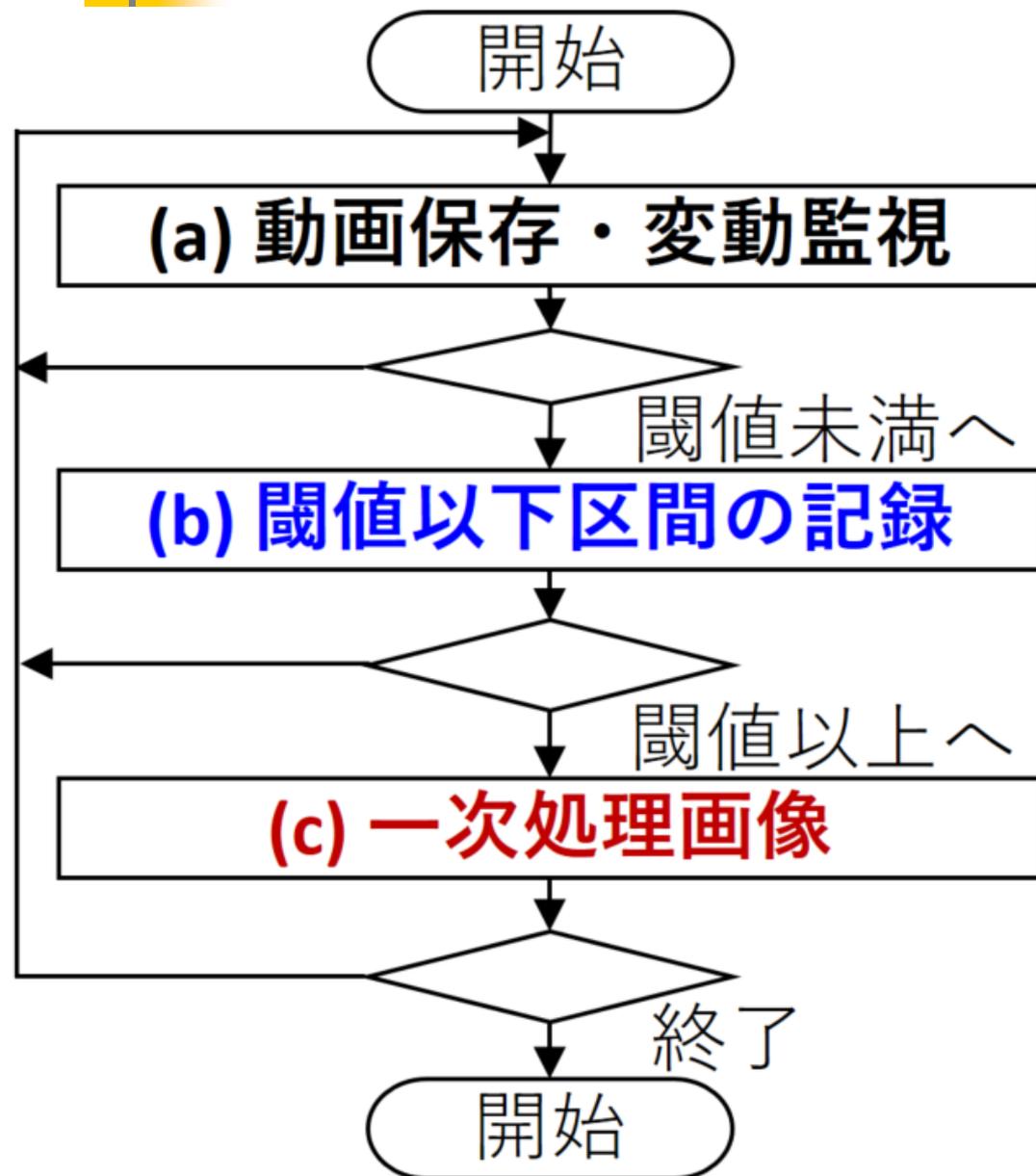


ウェアラブルカメラの動画変動の事例



- コンテナの自動判別に失敗しても、以下で絞込み可能
 - 一次処理の中間結果: 閾値以下の区間, C2_1, C2_2
 - ダイジェスト: 変動の推移

一次処理の流れ



■ オリジナルデータ

- 動画

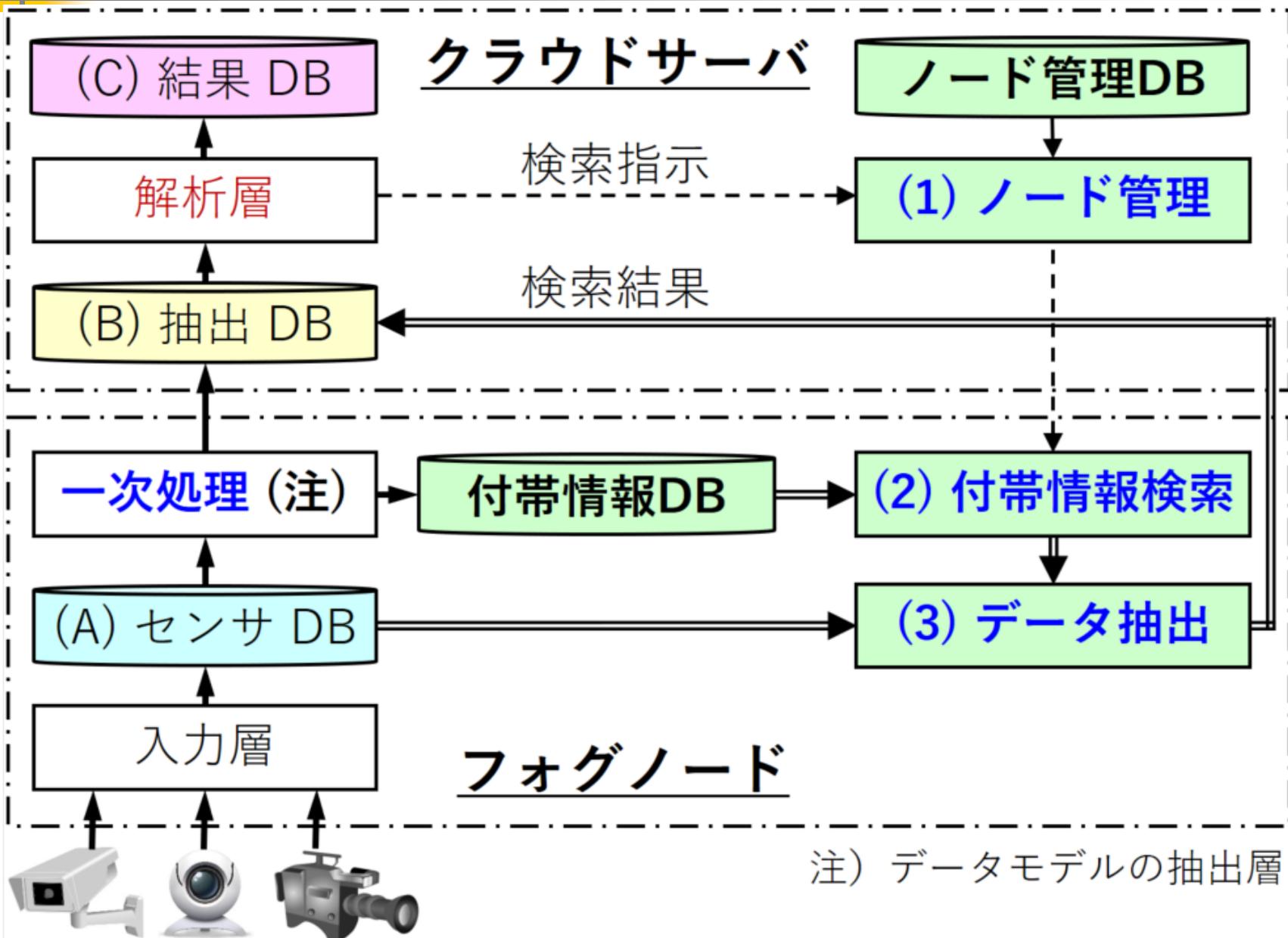
■ 一次処理結果

- 画像 (C2_1, C2_2)
- 判別結果

■ 付帯情報

- 変動の推移
- 閾値以下区間情報
- C2_1, C2_2
- C2区間情報

提案方式の構成



(1) 動画の変動

1. 動画のフレームをグレースケールに変換
2. 変換フレームの絶対差分のヒストグラムを作成
3. ヒストグラムを輝度で重み付けし, 画素数で平均

$$D_t = \sum n_{it}i / N$$

i: 輝度, n: 輝度iの画素数, N: 全画素数, t: 時刻

(2) コンテナ判別: QRコードを使用 (D_tが極小のフレーム)

(3) プログラム: Python; データベース: MongoDB

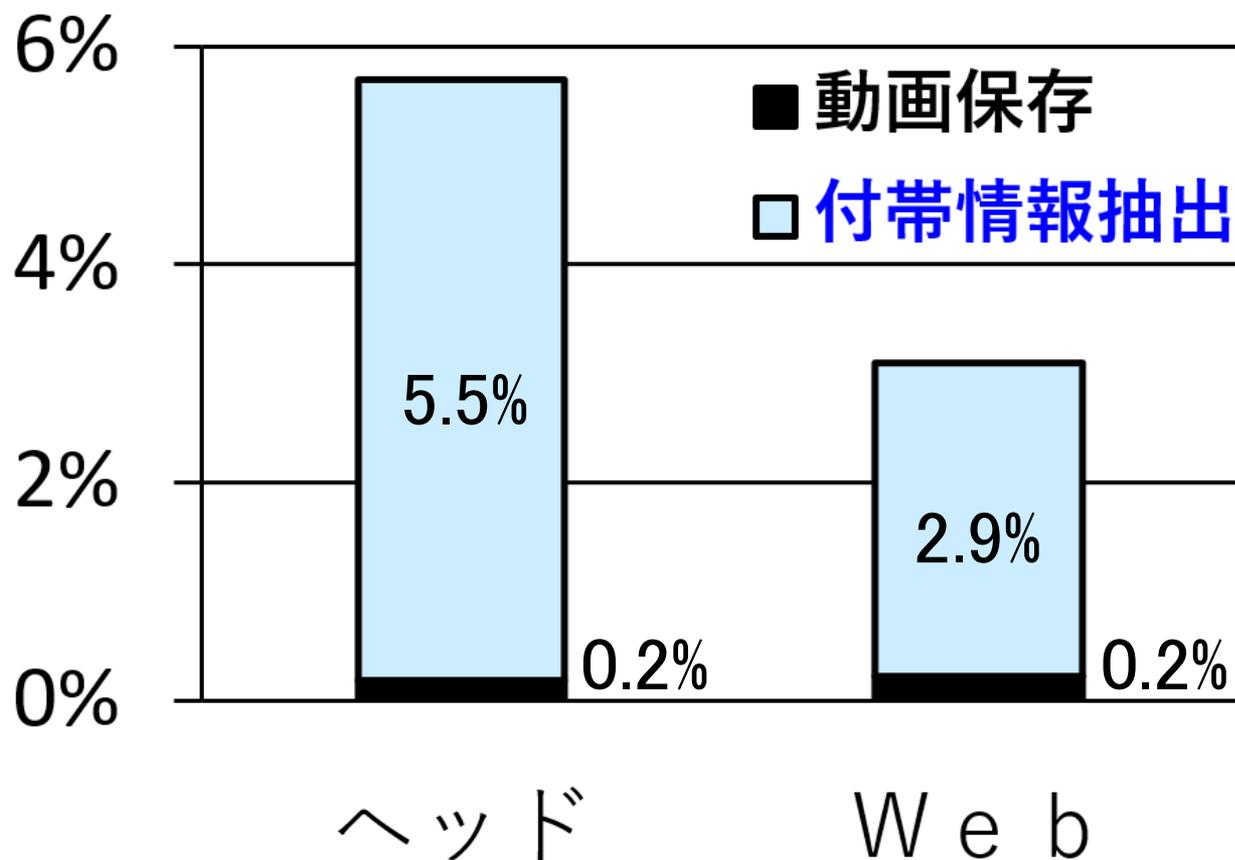
■ 研究室の本棚を使用



- (1) 入口, 本棚にQRコード (6.7cm × 6.7cm) を貼付
- (2) ウェアラブルカメラ (640 × 480dpi) で順次確認
- (3) 本棚の前 2m に**固定Webカメラ** (精度: 同上) 設置
- (4) 入口から本棚まで約5m

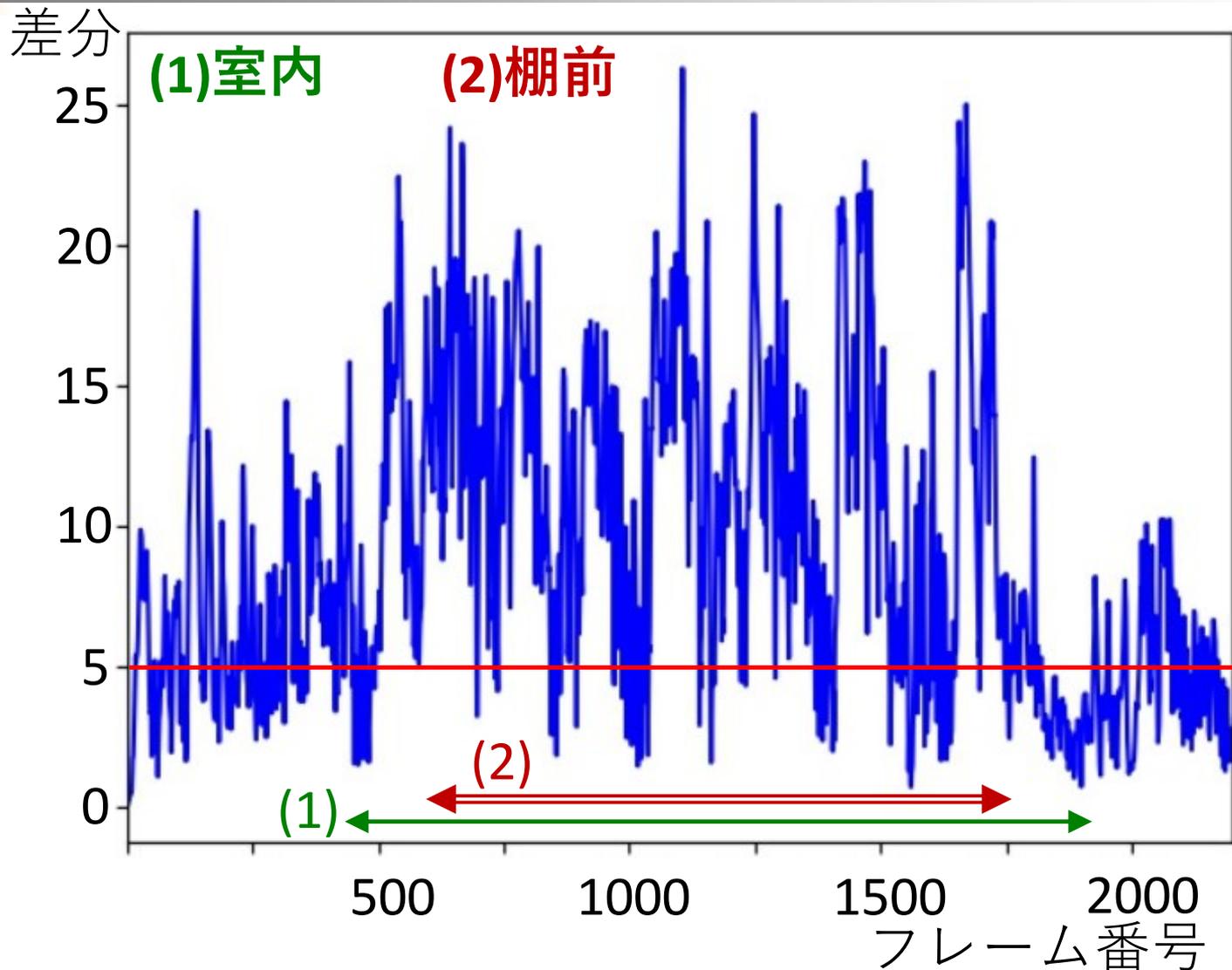
データ入力効率の評価

時間割合



■ 付帯情報抽出の効率はカメラで異なった。

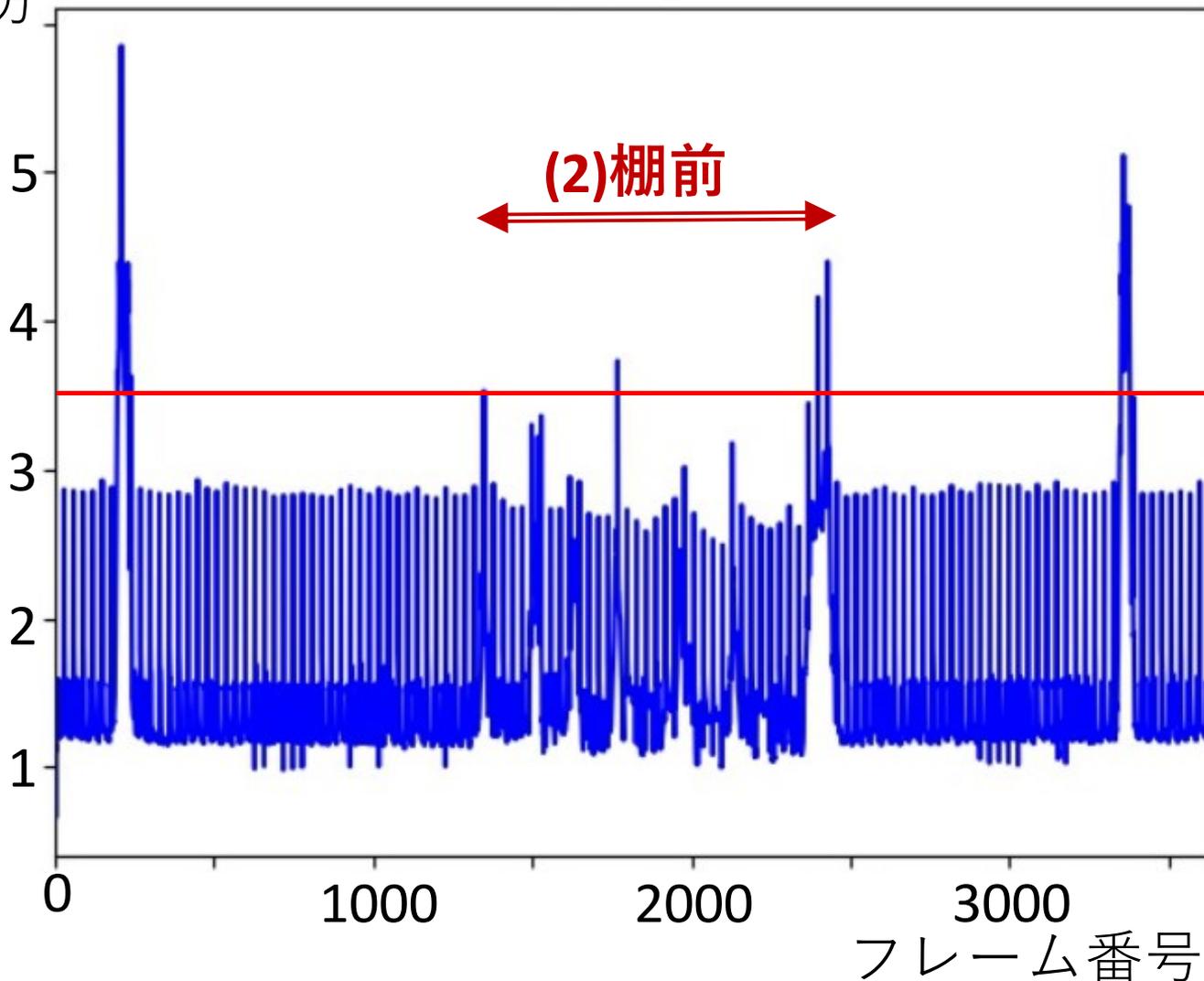
効率の差異分析(ウェアラブルカメラ)



■ 変動頻度大: 閾値以下区間=48 (棚前)

効率の差異分析(固定Webカメラ)

差分



■ 変動少: 閾値以下区間=3 (棚前)

QRコードの判別

■ 以下の方法で全てのQRコード判別

- 作業の前後で1秒程度QRコードを確認
- 変動が極小になるフレームでQRコードを判別

■ C2区間情報 (QRコード判別区間)

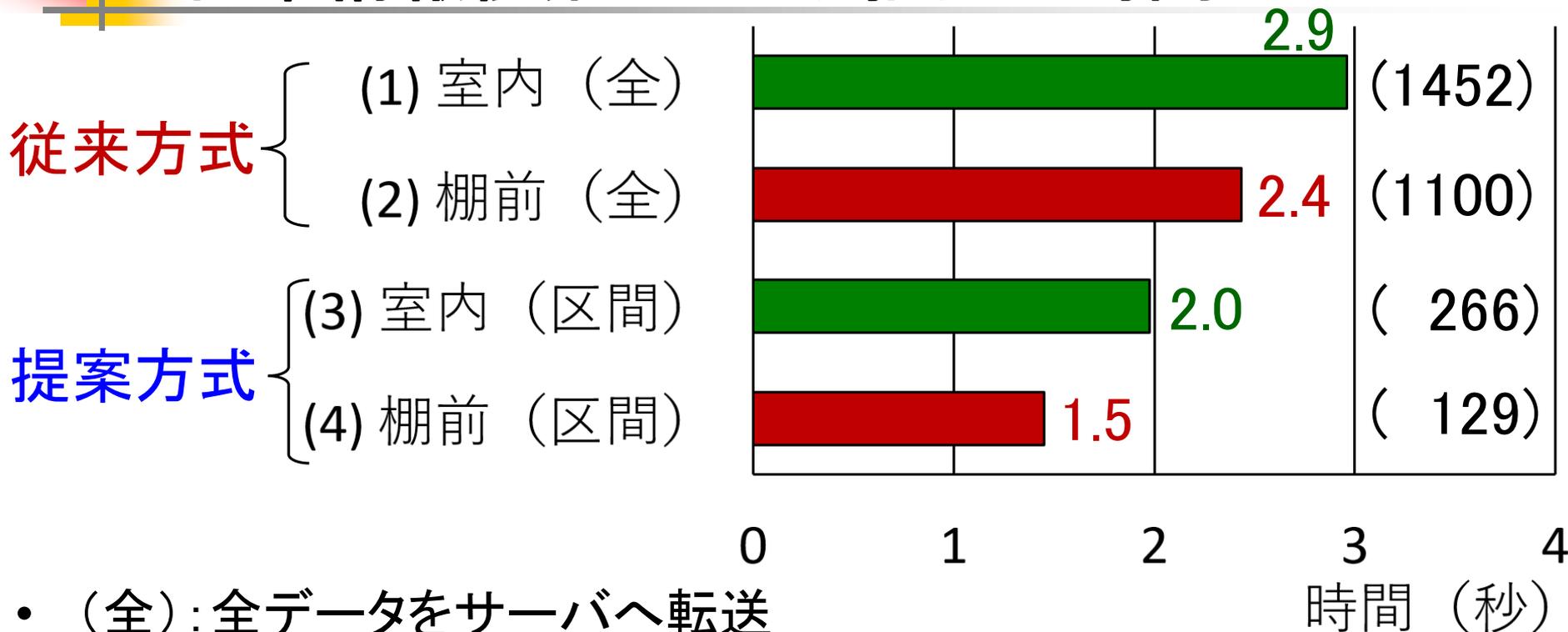
QRコード	0	7	6	5	4	3	2	1
開始 F	448	732	845	970	1140	1371	1520	1900
終了 F	493	738	895	1017	1232	1405	1647	1900

注) F : フレーム番号 (30フレーム/秒)

●QRコード

0: 入口; 1: 出口; 7~2: 本棚に貼付

付帯情報検索とデータ抽出の時間



- (全): 全データをサーバへ転送
- (区間): 閾値以下の区間のみサーバへ転送(付帯情報含む)

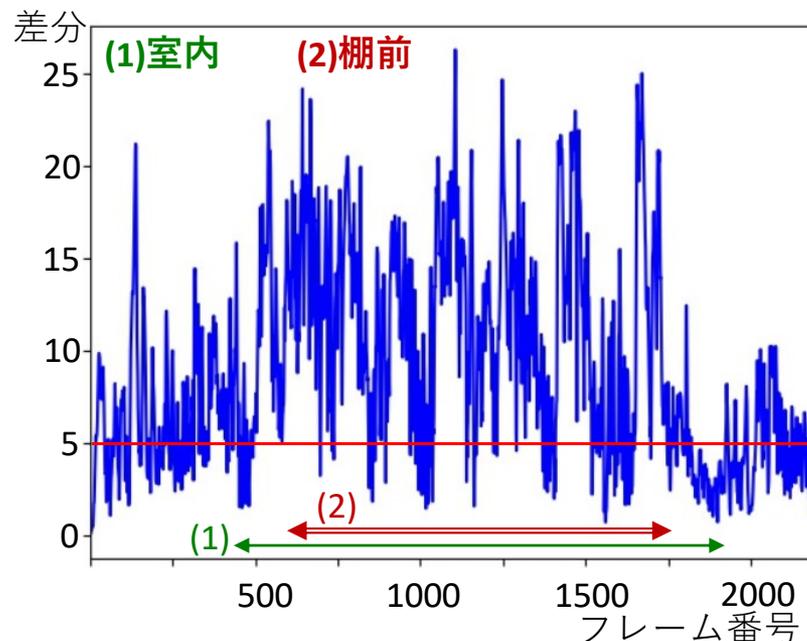
- 閾値以下の区間のみを転送する方が効率的
 - データ量(18%, 12%)に比較し改善は少ない
- 固定Webカメラにより, 区間の絞込みが可能

1. 閾値以下の区間のデータ転送効率の改善が少ない

✓ 原因:

- 1回の転送フレーム数は平均4.2および2.7枚, 効率は転送回数に依存 (0.03秒/回)
- 本棚前は対象との距離が小さく変動頻度が増大

✓ 対策: 間隔の短い区間は一括して転送



1. 閾値以下の区間のデータ転送効率の改善が少ない
 - ✓ 原因:
 - 1回の転送フレーム数は平均4.2および2.7枚, 効率は転送回数に依存 (0.03秒/回)
 - 本棚前は対象との距離が小さく変動頻度が増大
 - ✓ 対策: 間隔の短い区間は一括して転送
2. 本方方式は, 常時監視など, **監視時間に比較して対象時間の割合が小さい場合に有効と考えられる.**
 - ✓ 例: 監視カメラの動画から変動部分のみを抽出

■ 解決しようとする課題

- ✓ フォグコンピューティングにおけるデータ検索で、膨大なデータが検索対象になる場合は効率が低下

■ 提案: 以下の方式で検索対象を絞り込み

1. 付帯情報(一次処理中間結果, ダイジェスト)
2. 複数のセンサの付帯情報を活用(棚前の時間帯検知)

■ 結論

- ✓ 提案方式はデータの絞り込みに有効
- ✓ ただし, 対象データの過度な細分化では効率が低下

フォグコンピューティングにおける センサデータ管理方式の提案

ご静聴ありがとうございました。

静岡理科大学

総合情報学部 情報デザイン学科

工藤 司

