

複数カメラを用いた分散型 リアルタイム人物追跡手法の 比較検討

片山 ひかる[†], 矢野 英人[‡], 義久 智樹[‡], 下西 英之[‡]

[†]大阪大学 大学院情報科学研究科

[‡]大阪大学 サイバーメディアセンター



目次

- **背景と関連研究**
- 設計
- 実装
- 評価
- まとめ

背景

- 街中には多数の防犯カメラが設置されている
 - 東京には約4万台存在[1]
- 広範囲な人物追跡に適応可能
 - 混雑回避
 - マーケティング
 - 都市計画

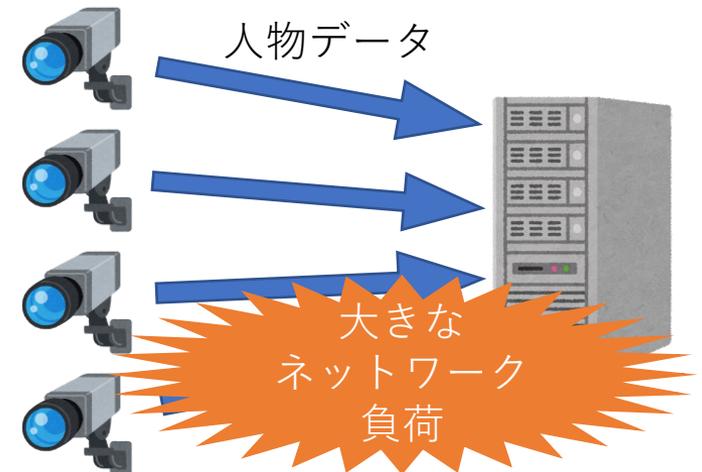
関連研究

- 一般的な人物追跡手法[2][3]
 - カメラで取得した映像をサーバに送信
 - サーバで全ての映像を処理



カメラが増加するとネットワーク負荷が増加

- データ送信速度にも比例
 - パケットロスの発生
 - リアルタイム性がない問題



[2] Ristani, Ergys, and Carlo Tomasi. "Features for multi-target multi-camera tracking and re-identification." the IEEE conference on computer vision and pattern recognition. 2018.

[3] Neehar Peri, et al. "Towards Real-Time Systems for Vehicle Re-Identification, Multi-Camera Tracking, and Anomaly Detection". the IEEE conference on computer vision and pattern recognition. 2018.

本研究の目的

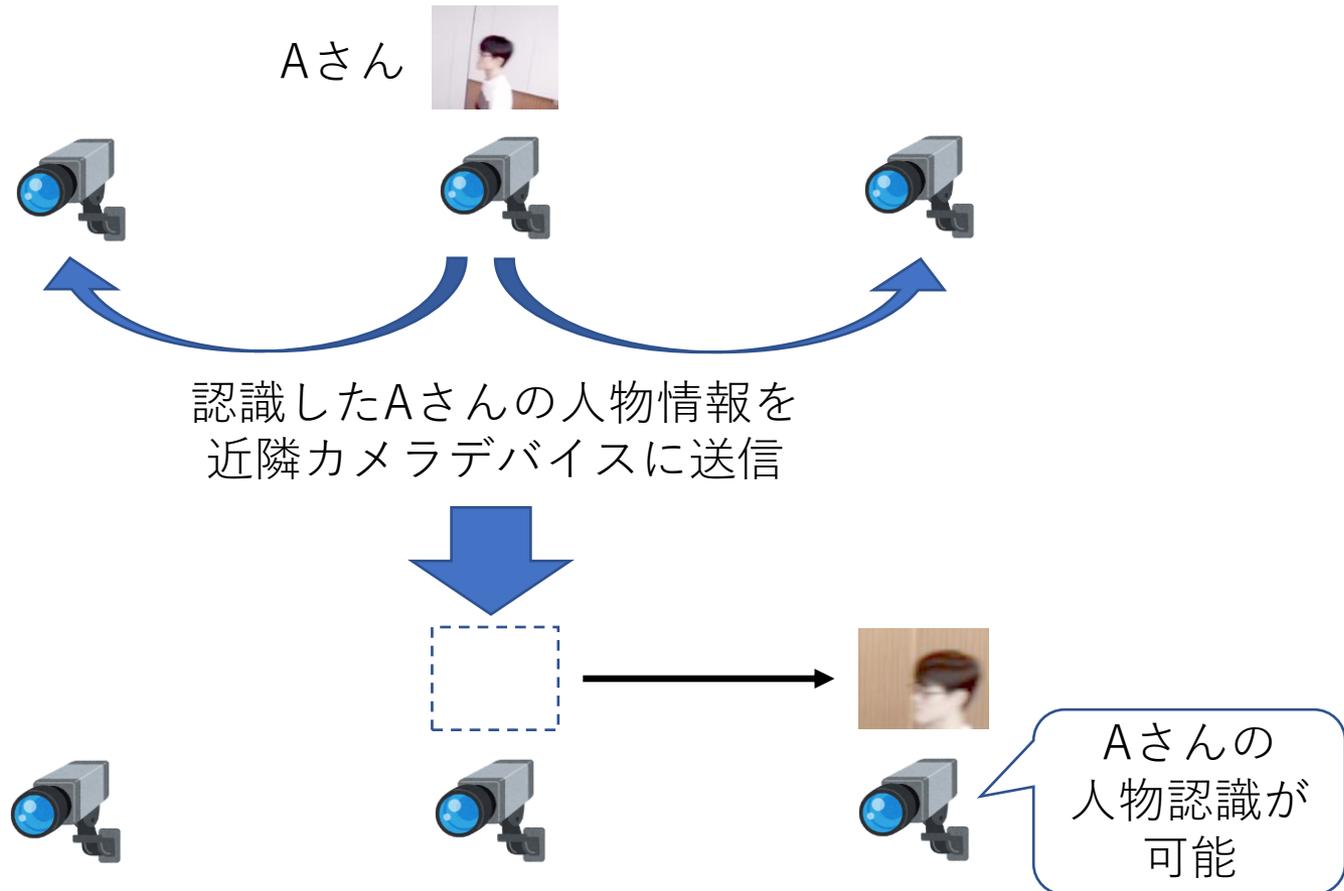
- 人物追跡システムにおけるネットワーク負荷軽減
- アプローチ
 - 人物情報をカメラデバイス間で共有
 - 共有するタイミングが異なる二種類の手法
 - プッシュ型共有手法
 - 人物認識処理後に人物情報を他のカメラデバイスに送信
 - プル型共有手法
 - 人物認識処理前に新規人物情報を他のカメラデバイスから受信

目次

- 背景と関連研究
- **設計**
- 実装
- 評価
- まとめ

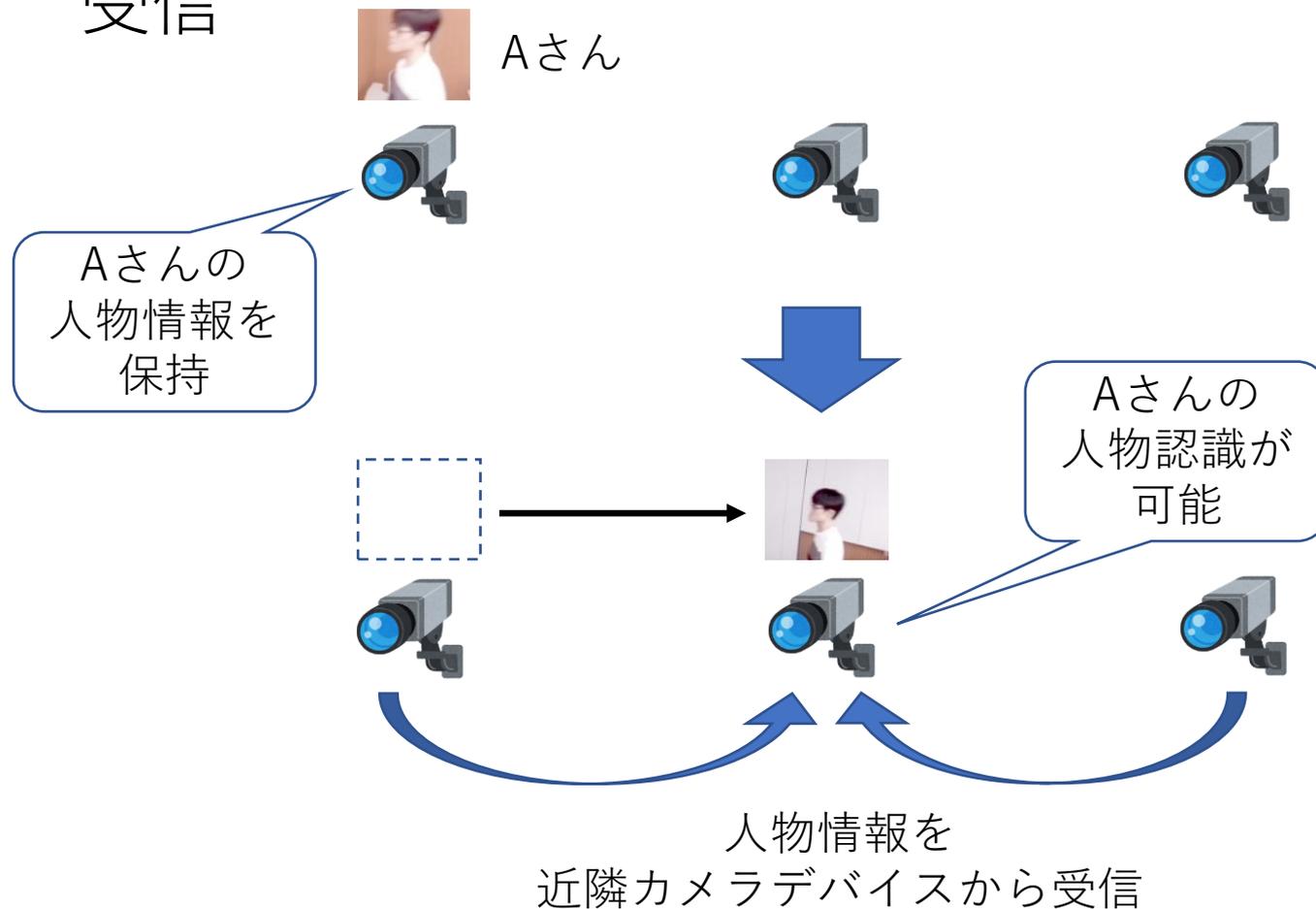
プッシュ型共有手法

- 人物認識後，近隣カメラデバイスに人物情報を送信



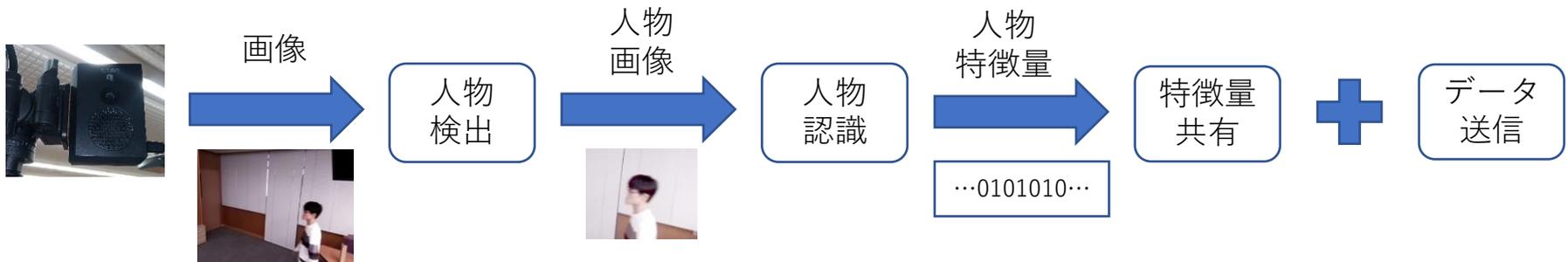
プル型共有手法

- 人物認識の直前に近隣デバイスから人物情報を受信



カメラデバイスの処理内容

- カメラデバイスで人物検出・人物認識
 - 人物の特徴量の抽出
- 人物の特徴量とIDを近隣のカメラデバイスと共有
 - 人物特徴量の比較が可能
- 人物IDとカメラIDをサーバに送信

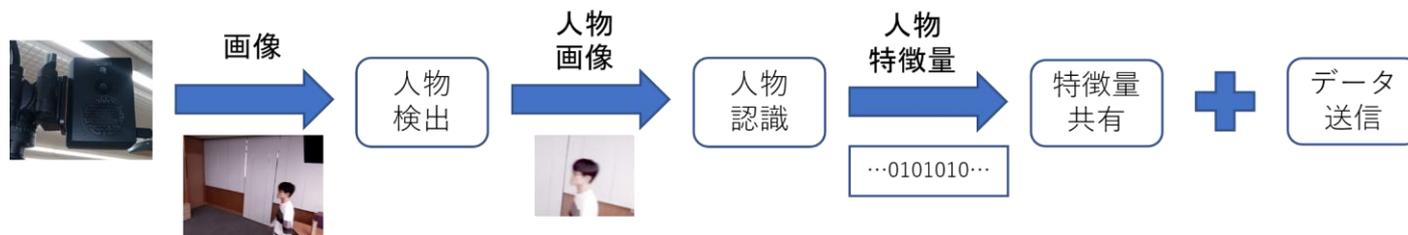


目次

- 背景と関連研究
- 設計
- **実装**
- 評価
- まとめ

カメラデバイス

- Raspberry Pi 4を使用
 - 比較的安価
 - 性能は低め
 - GPUなし
 - CPU性能低め
 - 少量のメモリ



人物検出・人物認識

- カメラデバイスの処理性能の低さ

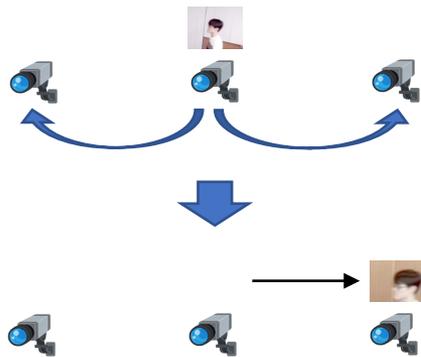
- 軽量なモデルを使用

- 人物検出：Haar-like cascade分類器
 - 人物の上半身を検出
 - 検出部分を切り抜き
- 人物認識：AKAZE特徴量マッチング
 - AKAZE特徴量を抽出
 - 特徴量同士の類似度から人物を認識
 - 類似している人物がない場合，新規IDを付与

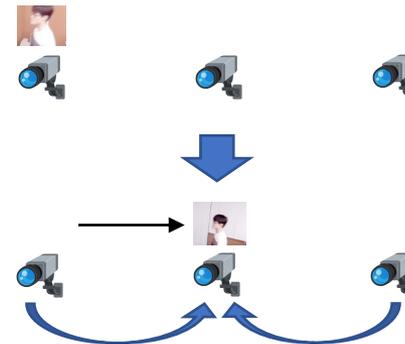


人物特徴量の共有

- 人物情報として人物特徴量と人物IDを近隣カメラデバイスと共有
- 人物特徴量をpickleファイルとして保存
- データの通信にはrsyncを利用
- 二種類の共有手法を実装



プッシュ型共有手法



プル型共有手法

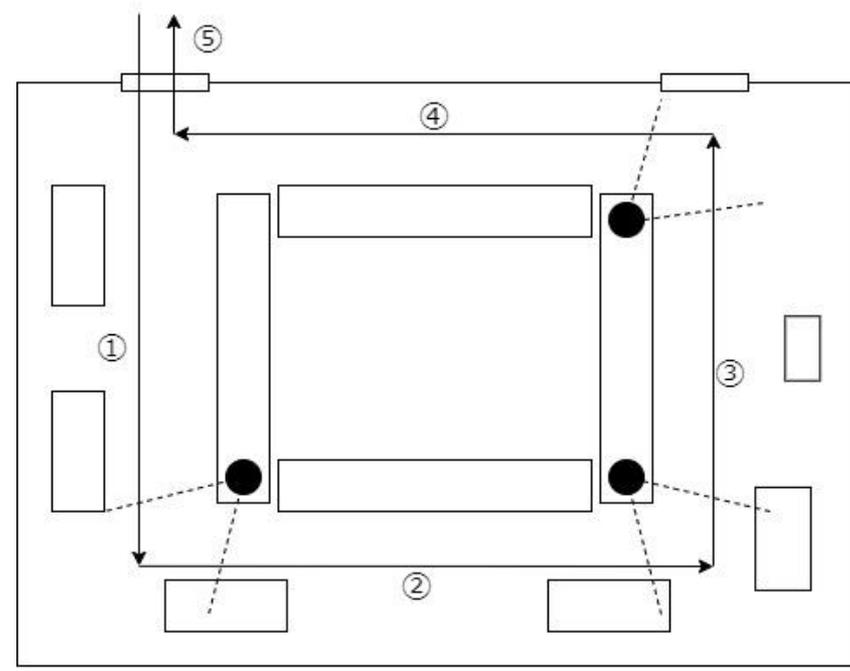
目次

- 背景と関連研究
- 設計
- 実装
- **評価**
- まとめ

実験

- 実験内容
 - 部屋にカメラデバイスを設置し，人が移動
 - 人物追跡処理を実行
- 計測内容
 - 条件を変化させた場合の通信データ量と処理時間
 - 移動人数: 1~3人
 - カメラデバイス数: 2,3台
 - 手法: 3種類
 - 提案手法の2種類
 - 比較対象のサーバ集約手法
 - カメラデバイス数と移動人数を増加させた際の通信データ量を推測

実験環境



実験環境の概要図
●: カメラデバイス, 矢印: 人の移動,
点線: おおよそのカメラの画角



左下のカメラから撮影される画像



右上のカメラから撮影される画像
(カメラ台数3台の時のみ使用)



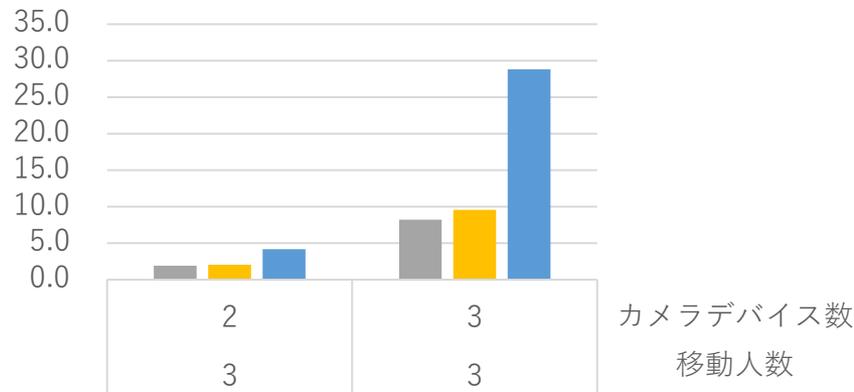
右下のカメラから撮影される画像

実測結果

- プル型共有手法の通信データ量が多い
 - データ共有方法の違い
 - プッシュ型共有手法：1人分の認識処理に1回
 - プル型共有手法：認識処理前に1回のみ

移動人数3人の時の通信データ量の実測値

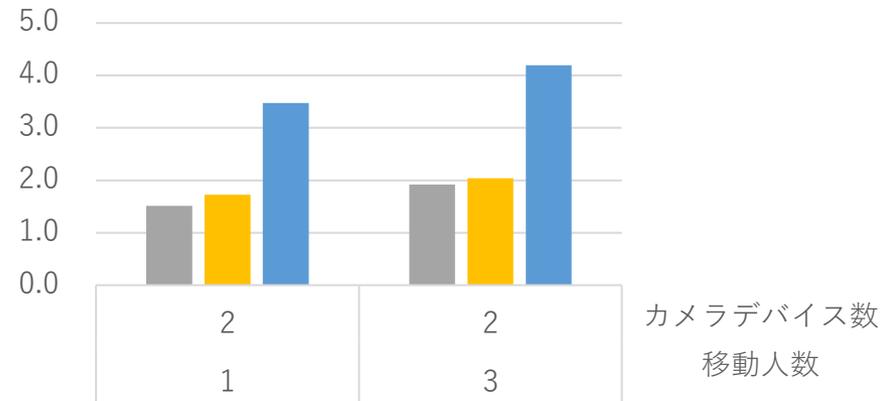
通信データ量[Kbps]



- サーバ集約手法におけるサーバへの通信データ量
- プッシュ型共有手法におけるAPへの通信データ量
- プル型共有手法におけるAPへの通信データ量

カメラデバイス2台の時の通信データ量の実測値

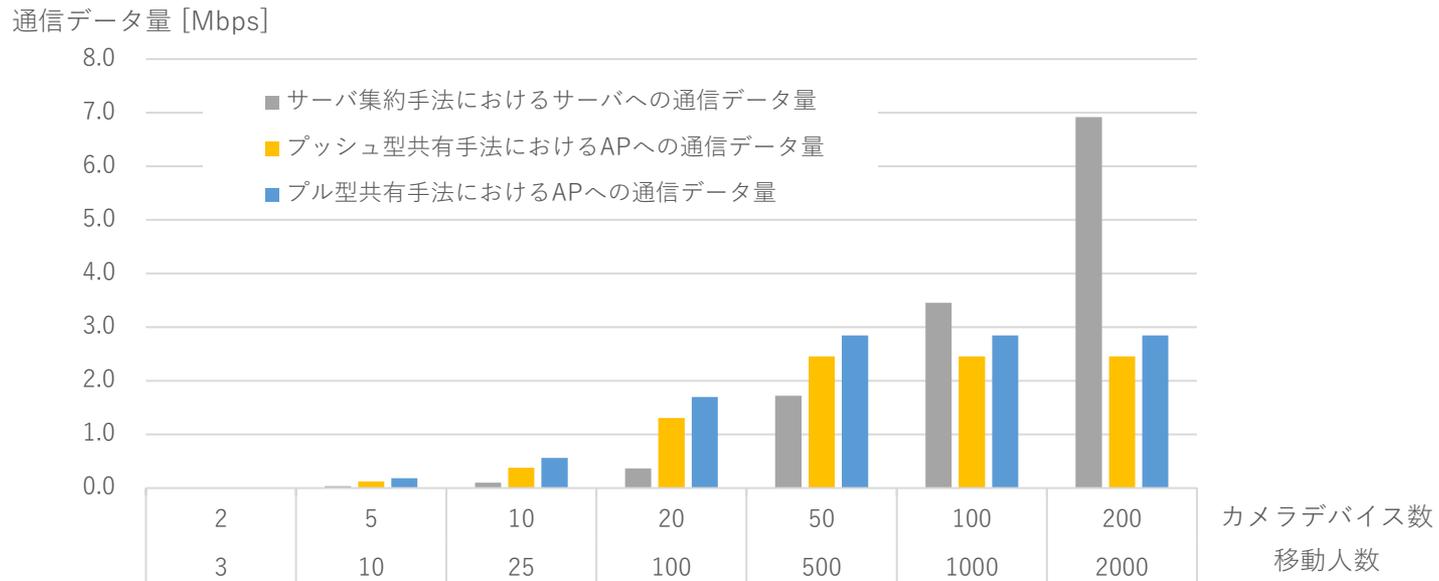
通信データ量[Kbps]



- サーバ集約手法におけるサーバへの通信データ量
- プッシュ型共有手法におけるAPへの通信データ量
- プル型共有手法におけるAPへの通信データ量

規模を拡大させた結果

- カメラデバイス数と移動人数を増加させた際の通信データ量を推測



- ネットワーク負荷の上昇を抑えることを確認

処理速度の計測結果

- 処理速度は1 [fps]とやや遅め

カメラ デバイス数	共有手法	1つの特徴量を取得する際 の処理速度 [fps]	2つの特徴量を取得する際 の処理速度 [fps]
2	プッシュ型	1.250	1.027
	プル型	1.168	1.167
3	プッシュ型	0.8586	0.5756
	プル型	0.7600	0.7270

- プル型共有手法は取得特徴量の数が増加しても処理速度に大きな変化なし
 - 原因はデータ共有方法の違い

人物検出・人物認識の精度

- 人物検出の精度：0.35
 - 全ての人物画像数のうち人物を検出できた割合
- 人物認識の精度：0.55
 - 誤りのないID割り当ての割合
- 精度低下の要因
 - 背景画像を人物として誤検出
 - 人物認識処理において時間を未考慮

まとめ

- 分散型人物トラッキング手法を実装
 - Haar-like cascade分類器とAKAZEを使用
 - 抽出した特徴量とIDを近隣カメラデバイスと共有
- 通信データ量の比較実験を行った結果、実装した手法がネットワーク負荷を抑えることを確認
- 処理速度が1 [fps]とやや遅め
- 今後の予定
 - 処理速度向上と人物検出・認識精度の向上を図る
 - 差分画像の利用
 - 人物特徴量に利用期限を付与