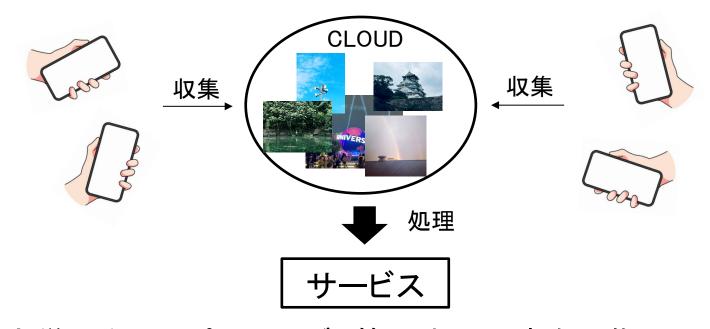
Mobile Crowd Photographing の類似度に基づく キャッシュ置換

立命館大学 情報理工学部 鄧 千宜 上山 憲昭 2023.03.03

Mobile Crowd Photographing

- Mobile Crowd Photographing (MCP)
 - スマホで撮影された写真を様々なサービスに活用



- 大学のキャンパス内でゴミ箱の中の写真を取集し、 リサイクルボックスの設置場所を決定
- スマホの写真を用いて屋内のナビゲーション
- 災害発生時に、スマホで撮影された被災状況に関する写真を 取集し、避難者の誘導や救助活動に活用

MCPの課題

- ネットワーク上に数千万枚以上の写真が存在
 - 画像は冗長性が高い
 - ユーザの要求画像と完全に一致する画像を配信する必要性 は低い
- キャッシュサーバから配信される機会が多い
 - CDNやエッジキャッシュなど
 - キャッシュ容量が有限
 - キャッシュ置換(LRUやFIFOなど)が必要

研究目的

- ユーザに低遅延で配信
- ユーザのニーズをできるだけ満たしたい

- キャッシュ内に存在する画像の中で、要求画像との類似度が 最大の画像を配信(Soft Cache Hit)
- 提案キャッシュ置換法: 他のキャッシュ画像との類似度が高い (冗長性が高い) 画像をキャッシュから優先的に削除
 - 類似度計算に要する時間を抑制
 - 配信画像と要求画像との差異(ソフトキャッシュヒット距離)を向上
 - 画像の冗長性に加え, 人気度も考慮

提案方式の処理の流れ

- 画像の類似性を高速に判定するため、SIFT (scale-invariant feature transform)アルゴリズムで画像の特徴点を抽出
 - ⇒ 画像データを全体的に利用する必要性なし
- 特徴点を、Bloom Filterでベクトルに変換
 - → 特徴点数が異なる多様な画像を統一的に処理可能
- E2LSH (exact euclidean locality sensitive hashing)アルゴリズムで類似度に基づき画像を複数のグループに分類
 - ⇒ 画像の類似度判定に要する時間を短縮
- キャッシュ内の画像を、冗長性と人気度の両方を考慮した総コストによりキャッシュ置換

Bloom Filterでベクトルに変換



- 特徴点は様々な長さのベクトル
 - 特徴点1:[103.1, 92.2, 234.9, ...]
 - 特徴点2:[12.1, 32.4, 183.4, ...]
 - 特徴点3:[198.4, 24.4, 35.8, ...]
 - • •

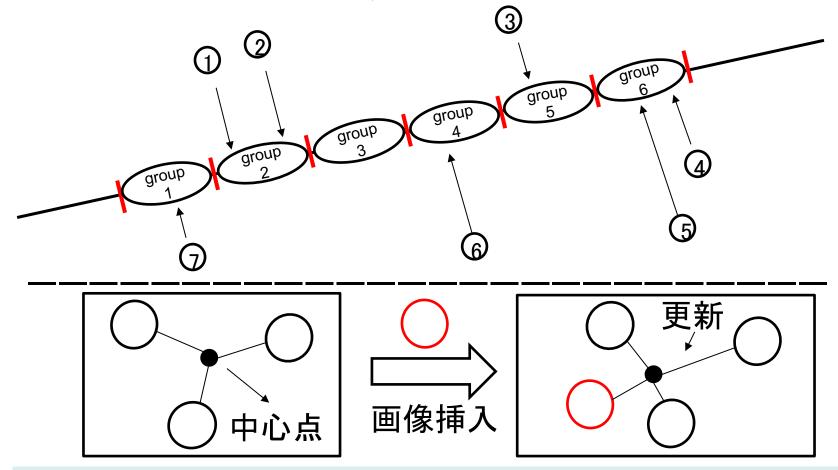
Bloom Filterのビットマップに 変換

[0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, ...]

- 各画像の特徴量 (Bloom Filterのベクトル) の長さを統一
- バイナリ演算が可能
 - 計算速度高速 & 計算量減少
- 同じビットが多い画像 ⇒ 類似

E2LSHによる画像のグループ化

- 画像の類似度判定に要する時間を短縮したい
- キャッシュ中の画像を類似度に基づきグループ化



新たな画像をキャッシュに挿入する際,類似度の近いグループの中心点との距離だけを考慮

提案キャッシュ置換法

■ キャッシュ内の画像の中から、要求画像と特徴量の距離が

最小の画像をユーザに配信

- 距離に関するコスト(C_d)
 - 類似性が高い画像を削除
 - ullet m個の画像中、k番目に冗長な画像データ: $\mathbf{C}_d = rac{m-k}{m-1}$
- 人気度に関するコスト(C_p)
 - m個の画像中, j番目に人気の画像データ: $C_p = \frac{j-1}{m-1}$

3 (2)	3 B 2
(C) 2	1 A 1

	A	В	С
$C_{\rm p}$	0	0.5	1
C_d	1	0	0.5
ω	0.5	0.5	0.5
C_{total}	0.5	0.25	0.75

- キャッシュ内の画像の総コストによってキャッシュ置換
 - $C_{total} = C_p * \omega + C_d * (1 \omega)$

 $0 \le \omega \le 1$

※ωは人気度の重み

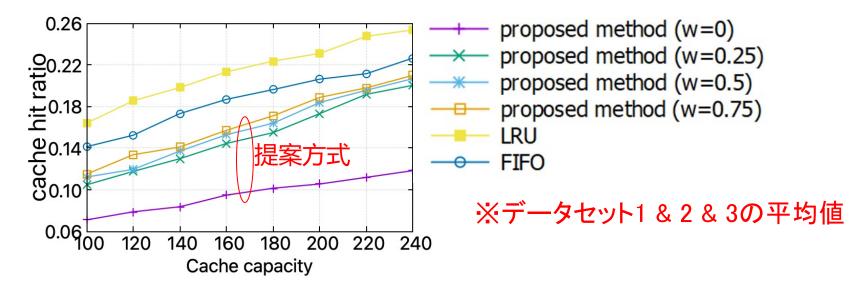
■ C_{total}が最大の画像を削除

性能評価条件

- ユーザから1万回の要求
- Zipf分布(パラメータ:0.8)でランダムに要求画像を選択
- 四つの画像データセットを各々,用いた結果で評価
 - データセット1 & 2 & 3 ⇒ 異なる種類の画像から構成
 - データセット4 ⇒ 似た種類の画像から構成
- LRUとFIFOと性能を比較
- 評価尺度
 - キャッシュヒット率 (Cache hit ratio)
 - キャッシュ平均距離 (Cache average distance)
 - 平均キャッシュヒット距離 (Average cache hit distance)
 - 平均ソフトキャッシュヒット距離 (Average soft hit distance)

評価結果:キャッシュヒット率

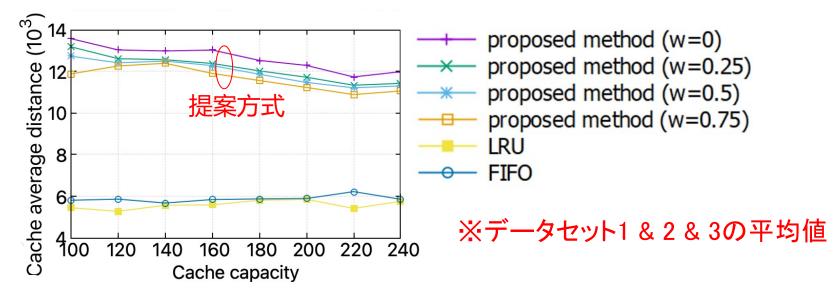
■ キャッシュヒット率 (Cache hit ratio)



- 提案方式はキャッシュ内の多様性を向上する反面⇒ キャッシュヒット率はLRUとFIFOより低下
- 人気度の重み ω の増加
 - 高人気の画像がキャッシュに残る可能性が増加
 - キャッシュヒット率が増加

評価結果:キャッシュ平均距離

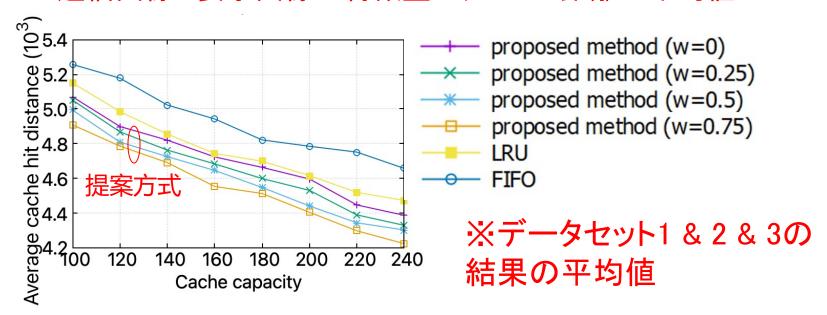
- キャッシュ平均距離 (cache average distance)
 - キャッシュ内の全て画像から中心点までの距離の平均値



- 提案方式は類似度が高い画像を優先的に削除
 - キャッシュ内の多様性が向上 ⇒ キャッシュ平均距離が高い
- 人気度の重み ω の増加 ⇒ 距離の重み減少
 - 冗長な画像が残る可能性が増加 ⇒ キャッシュ平均距離は減少

評価結果: 平均キャッシュヒット距離(1)

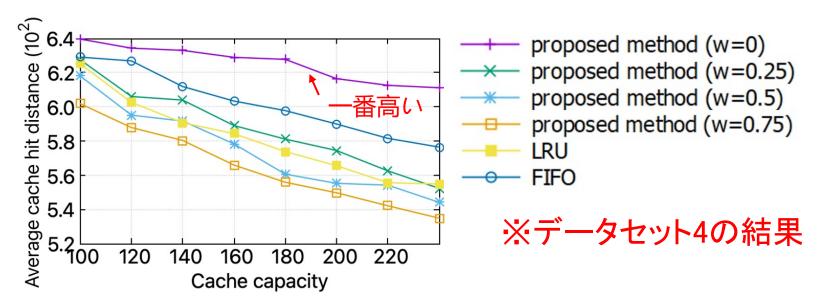
- 平均キャッシュヒット距離 (Average cache hit distance)
 - 送信画像と要求画像の特徴量ベクトルの距離の平均値



- 提案方式はキャッシュ内の多様性が向上し、平均キャッシュヒット距離は減少 ⇒ 提案方式の有効性を確認
- 人気度の重み ω の増加 ⇒ ヒットの機会増加
 - 平均キャッシュヒット距離は減少

評価結果: 平均キャッシュヒット距離(2)

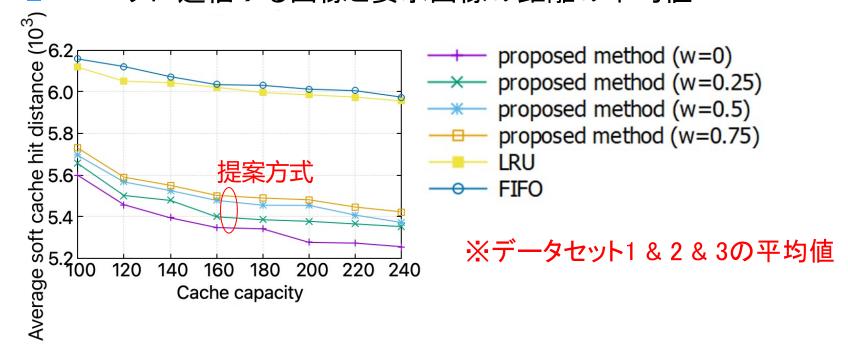
- 平均キャッシュヒット距離 (Average cache hit distance)
 - ユーザに送信する画像と要求画像の距離の平均値



似たタイプの画像から構成される場合,提案方式のキャッシュ内データの冗長性低減効果は減少

評価結果: 平均ソフトキャッシュヒット距離

- 平均ソフトキャッシュヒット距離 (Average soft cache hit distance)
 - ソフトキャッシュヒット(キャッシュミスの場合)
 - ユーザに送信する画像と要求画像の距離の平均値



- 提案方式: キャッシュミスした配信の平均距離は大幅に減少
- 人気度の重み ω の減少 ⇒ キャッシュ内の多様性増加
 - 平均ソフトキャッシュヒット距離減少

まとめ

- MCPでは膨大な数の画像がネットワーク上に存在⇒ユーザの要求画像に完全に一致しなくても、類似度の高い画像を配信すれば十分
- MCPのキャッシュ内の画像の冗長性を低減することで、配信画像と要求画像の差異を低減するキャッシュ置換法を提案
- 画像を複数のグループに分類することで、画像の類似度の 判別の計算処理量を抑制
- 今後の取り組み
 - 機械学習など、他の方法でグル―プ化する方法を検討
 - 画像の特徴点の抽出と判別の効果を評価

ご清聴ありがとうございました