一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
IN2022-104(2023-03)

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

鄧　　千宜† 　上山　憲昭†

† 立命館大学 情報理工学部 情報理工学科
〒525-0058 滋賀県草津市野路東 1-1-1
E-mail: †is0652sr@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

あらまし　近年，スマートフォンの発展と Twitter や Facebook などの SNS (Social Network Service) の普及に伴い，スマホからアップされた写真を様々なサービスに活用する MCP (Mobile Crowd Photographing) の利用が広がっている．例えば災害発生時に，建造物などの障害状況を人々がスマホで撮影してネットワーク上にアップするなどの活用が考えられる．MCP においては，ネットワーク上に数千万枚以上の画像がアップされ，似たような画像も多いため画像の冗長性が高い．一方，ユーザの要求する画像に厳密に一致する画像を配信する必要性は低いため，いかにして所望の画像に近い画像を配信できるかが重要である．画像の配信には CDN やエッジキャッシュなど，キャッシュ配信が用いられる機会が多い．キャッシュ内に要求する画像に近い画像が存在すれば，直接キャッシュ内の画像を活用することで，ネットワーク上の遠くから取得する必要がなく，画像の取得に要する時間を短縮できるしかしキャッシュの容量は有限であるため，容量を超過するとき，キャッシュに残す画像を選択するキャッシュ置換法が必要である．代表的なキャッシュ置換法は LRU (Least Recently Used) と FIFO (First In First Out) などである．そこで本稿では，キャッシュ内に存在する他の画像との類似度が最大の画像から優先的に削除するキャッシュ置換法を提案する．キャッシュ内における画像の類似度に基づいて画像データをグループ分けすることで，類似度計算に要する時間を低減し，また人気度も考慮することでキャッシュヒット率の向上も目指す．そして LRU と FIFO と性能を比較し，提案方式の有効性を示す．
キーワード　類似度，キャッシュ置換，冗長性

Cache Replacement Based on Similarity in Mobile Crowd Photographing

Qianyi DENG† and Noriaki KAMIYAMA†

† College of Information Science and Engineering, Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-0058
E-mail: †is0652sr@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

Abstract　In recent years, with the development of smartphones and the spread of social network services (SNS) such as Twitter and Facebook, the mobile crowd photography (MCP), in which photos uploaded from smartphones are used for various services, has been widely used. For example, when a disaster occurs, tens of millions of photos are uploaded onto the network. While tens of millions of images are uploaded to the network, there is little need to deliver images that exactly match the user's requirements, so it is important to be able to deliver images that are close to the desired images. Images are often delivered from cache servers such as CDNs and edge caches. Because of the large number of images on the network and the large number of similar images, there is a high degree of image redundancy. In this case, it is difficult to send the image requested by the user immediately. Therefore, the status of images in the cache is important. If the requested image is available in the cache, it can be used directly in the cache and sent to the user immediately, without the need to search the network. As a result, the way the cache is managed is also important. However, since the cache capacity is limited, a cache replacement method is needed to select images to be left in the cache when the cache capacity is exceeded. Typical cache replacement methods include LRU (Least Recently Used) and FIFO (First In First Out). In this paper, we propose a cache replacement method that preferentially deletes images with the largest similarity to other images in the cache. By grouping images based on their similarity in the cache, we aim to reduce the time required for similarity calculation and improve the cache hit rate by considering popularity. The performance of the proposed method is compared with that of LRU and FIFO, and the effectiveness of the proposed method is demonstrated.
Key words　Similarity, Cache Replacement, Redundancy

1. はじめに

近年のスマホのカメラ機能の向上や，ソーシャルメディア (Twitter や Facebook など) の利用機会の増加に伴い，ネットワーク上に写真をアップロードし，自分の生活シーンを他者と共有する mobile crowd photography (MCP) の活用が増えている．ネットワーク上には大量の画像データが存在し，似たような画像もたくさんあり，ネットワーク上に存在する画像データの冗長性が高くなる．そのためネットワーク上のストレージ資源の消費量が増加し，また検索の効率とユーザの体験品質も低下する．またユーザの要求する画像に厳密に一致する画像を見つけるのも困難である．

動画コンテンツや Web データのネットワーク上の配信には，ネットワーク上の様々な場所にキャッシュサーバを設置する CDN や，セルラーネットワークの基地局にキャッシュサーバを設置するモバイルエッジなど，キャッシュサーバが広く用いられる．しかしキャッシュの容量は有限であり，容量を超過するとき，キャッシュに残す画像を選択するキャッシュ置換法が必要となる．ユーザが画像データを要求したとき，最初にキャッシュ内に所望のデータが存在するか否かを確認する．要求画像が存在する場合，ユーザにはキャッシュから直接配信することで，遅延時間の低減と，ネットワーク資源の消費量の低減が可能である．キャッシュに所望のデータが存在しない場合は，ネットワーク上の検索を続行する．

代表的な 2 つのキャッシュ置換法は FIFO (First In First Out) と LRU (Least Recently Used) である．FIFO ではキャッシュ内のコンテンツを置き換えるとき，キャッシュに存在する時間が最長のコンテンツを消去し，新しいコンテンツを挿入する． LRU では，コンテンツが置き換えるとき，最後に要求されてからの経過時間が最大のコンテンツを消去する．しかし，これら 2 つの置換法はいずれもコンテンツがキャッシュ挿入された時刻のみを考慮したキャッシュ置換法である．キャッシュの冗長性を低減するには，コンテンツの内容を考慮する必要がある．

一般に画像データは曖昧性が高く，多くの場合，ユーザの要求する画像に厳密に一致する画像を配信する必要性は低い．そのためキャッシュ内に存在する画像データ間の冗長性が低く，できるだけ多様な画像データがキャッシュ内に存在することが望ましい．

そこで本稿では，キャッシュ内に存在する他の画像との類似度が最大の画像から優先的に画像データを削除するキャッシュ置換法を提案する．キャッシュ内における画像の類似度に基づいて画像データをグループ分けすることで，類似度計算に要する時間を抑制し，また人気度も考慮することでキャッシュヒット率の向上も目指す．本稿の貢献を以下にまとめる．

- MCP において，ユーザからの要求画像との類似度の高い画像データのヒット率が向上するよう，キャッシュ画像セットの冗長性の低減を目的とした画像データのキャッシュ置換法を提案する．
- 冗長性の高い画像データの判別の計算処理量を抑制するため，キャッシュ内で画像を類似性に基づき複数のグループに分類する方式を提案する．
- 実際の画像データセットを用いた計算機シミュレーション評価により，提案方式の有効性を明らかにする．

以下，2. 節では関連研究について述べ，3. 節で提案方式の詳細を述べる．そして 4. 節で性能評価結果を示し，最後に 5. 節で全体をまとめる．

2. 関連研究

MCP の画像データの冗長性を削減した通信方式として，近年，様々な研究が行われているが，ネットワーク上を流れるトラヒック量の削減を目的とした送信画像の選択法 [1] [2] [3] と，どの画像データをキャッシュするかというキャッシュ判断法 [4] [5] [6] に分類できる．

災害時にレスキュー隊が迅速に救助活動を行うには，被災者の周囲の状況を迅速に共有することが重要である．そのため被災者が撮影した写真を共有することが有効だが，限られた電力・NW (Network) 資源を節約するには，冗長な写真の送信を回避する必要ある．そこで Zuo らは，バッチで写真をレポジトリに送信する前に，写真のレポジトリと比較した冗長性とバッチ内の冗長性から送信すべき写真を選択する方式を提案している [1]．また Hua らは，災害時にユーザがスマホで撮影した写真の特徴量だけをクラウドサーバに送信し，クラウドで収集済みの写真データとの類似性を判断して回答し，ユーザは類似性が低い場合にだけ画像データを送信する方式を提案している [2]．提案方式では，さらにネットワーク内で類似性の高いデータを同じフローに集約して転送する．また Chen らは，いくつかの属性に基づき MCP で収集された写真を木構造で整理して，冗長度の高いものを除いて少数の写真に絞りこむ方式を提案している [3]．しかしこれらの方式では，キャッシュを用いた画像データの配信は考慮されていない．

キャッシュを考慮した MCP の優先制御方式として，Dinh らは，IoT (Internet of Things) サービスに対しての貢献度の高い，冗長性の低いデータを優先的にキャッシュする方式を提案している [4]．同様の情報をもつ複数のデータセットの単位でテーブルを管理し，既に同じセット内のデータがキャッシュされている場合にはキャッシュしないことでキャッシュの利用率を高める．また Neglia らは，要求コンテンツ x の代わりに類似コンテンツ y を配信したときのユーザのコスト $C_a(x, y)$ が与えられたときに，コストを最小化するよう配信する類似コンテンツの選択法を提案している [5]．さらに Sermpezis は，5G の small cell (SC) にキャッシュが設置されている状況で，ユーザの配信要求に対しキャッシュミスした際に，関連コンテンツがある場合 (Soft cache hit) は，それをユーザに推奨して了承されれば配信することを提案している [6]．Soft cache hit ratio (SCHR) が最大化するよう，キャッシュするコンテンツを選択する．しかしこれらの研究は配信する類似性の高いコンテンツの選択法や，キャッシュ挿入判断法であり， MCP の冗長性を考慮したキャッシュ置換法の研究はみられない．

3. 提案方式

提案方式はまず SIFT (scale-invariant feature transform) アルゴリズム [7] を用いて，１つの画像に対して複数の特徴点を抽出する．これらの特徴点は画像の内容を表すが，これらの特徴点の特徴量を Bloom filter (BF) [8] の入力とすることで，単一の特徴ベクトルに変換する．そして得られた特徴ベクトルを用いて，E2LSH (exact euclidean locality sensitive hashing) アルゴリズム [9] [10] によって，画像データを類似度に基づき複数のグループに分類する．そしてユーザからの配信要求に対して，要求データの特徴ベクトルとキャッシュ内に存在する各画像データの特徴ヘクトルとの距離が最小となるキャッシュ内の画像データを要求ユーザに配信する．さらに特徴ベクトルの差異と画像の人気度の両方を考慮して配信画像データを選択することも提案する．すなわち提案方式は，(1) 画像の特徴量の抽出，(2)Bloom filter による単一特徴ベクトルの生成，(3)E2LSH に基づく画像分類，(4) 距離と人気度を考慮したキャッシュ置換，の 4 つのステップから構成される．

3.1 画像の特徴量の抽出

任意の２つの画像間の類似性を判定するには，各画像の特徴量を定義する必要がある．同一の画像を拡大・縮小させたり，回転させた場合も，類似性の高い画像と判定する必要があるため，画像の特徴量には，画像のスケールと回転に対して不変であることが求められる．さらに画像の特徴量には，さまざまな視点の変化，ノイズの追加，照明の変化などに対してもロバス

トであることが求められる．SIFT アルゴリズムは，特徴点を効果的なローカル記述子 (local descriptor) として使用することで，光度変換 (photometric changes) と幾何変換 (geometric variation) に対してロバストな特徴があり，実際のアプリケーションで広く採用されている [7].

3.1.1 特徴点の検出

特徴点を正確に抽出して検出するためには，それらを位置とスケールを特定する必要がある．一般にスケールスペース検索では，特徴点はローカルピーク (Local peak) に存在する．これらの特徴量が変換後も安定して維持されるよう，これらの特徴量をフィルタリングすることが可能である．さらに，場所とスケールに関し画像をスキャンすることで，場所とスケールを選択し，潜在的な特徴点を見つけることができる．[2] で提案されている SmartEye では，ガウスピラミッド (Gaussian pyramid) を構築し，DoG (difference-of-Gaussian) 画像のローカルピークを検索することで，これらの操作を実現する．

特徴点を効率的に特定し選択するために，潜在的な各候補位置はサブピクセル精度で局所化され，さらに除去される．さらに向きによる安定性を考慮するため，各候補位置において複数の回転操作を行い，局所画像勾配方向に基づいて，各候補点に複数の方向を割り当て，各特徴点に割り当てられた方向，スケール，位置を基準にしてイメージを変換する．

3.1.2 特徴点の表現

高速検索性能をサポートする特徴点の効率的な表現のために，特徴点の記述子を定める．この記述子は識別可能で，簡潔で，かつ変換に対して不変である．コンパクトな特徴ベクトルを生成するためには，パッチのローカル勾配イメージを計算して正規化し，事前に計算された固有空間に投影する必要がある．この固有空間は，自然の風景の画像から抽出された多数の特徴点に由来する．実際には，記述子の数は画像から検出された特徴点の集合の基数に依存する．

3.2 Bloom filter を用いた特徴量の集約

抽出した画像の特徴量を用いて，次節で述べるように LSH (locality sensitive hashing) [11] を用いて，類似度に基づき画像をグループ化する．ただし LSH の入力には単一の特徴量ベクトルを用いる必要があるため，前節で述べた方法で抽出した複数の特徴量を，Bloom filter を用いて単一の特徴量ベクトルにマッピングする [2]．すなわち Bloom filter で生成した単一の特徴量ベクトルを LSH の入力とすることで，メモリスペースと計算時間を低減する．2 つの類似した画像の Bloom filter ベクトルには，状通するビットが多くの部位で含まれる．

Bloom filter は n 個アイテムのデータセット $S = \{a_1, a_2, ..., a_n\}$ に対して，k 個の独立したハッシュ関数 $\{f_1, ..., f_k\}$ と，m ビットのビットマップが用意される．最初はビットマップのすべてのビットが 0 に設定される． 各アイテムの特徴ベクトルをキーとして k 個の各ハッシュ関数から得られる k 個のハッシュ値に該当するビットマップのビットを 1 にセットする．そのため各アイテムは m ビットのビットマップで構成される m 次元のベクトル $[1, ..., m]$ 空間上の 1 点にマッピングされる．ハッシュ関数 f_i は，アイテム a を一様にランダムな方法で m 配列のビット位置の 1 つにマップすることができる．メンバシップクエリでは，すべての $f_i(a)$ が 1 の場合，アイテム a はセット S のメンバーと見なされる．それ以外の場合はメンバーではない．アイテム a がメンバーではないにもかかわらず誤ってメンバーと見なされる擬陽性が生じる．n 個のアイテムを持つデータセットに対して，Bloom filter が m ビットビットマップと k 個のハッシュ関数を持つ場合，偽陽性はおよそ $(1 - e^{-\frac{kn}{m}})^k$ で与えられる．偽陽性確率の下限界は，$k = (m/n) \ln 2$ のときで，$(1/2)^k$ または $(0.6185)^{m/n}$ である [12].

3.3 E2LSH に基づく画像分類

LSH (locality sensitive hashing) を使用して，類似度が高い (特徴量が近い) 画像集合を高い確率で同じハッシュバケットまたは近接ハッシュバケットにマップする [11]．LSH 関数群はデータの局所性の認識能力があり，類似性に基づきデータを集約することが可能である．すなわち似た特徴を有する類似画像は，より高い確率で同じバケットにハッシュ化できる [11]．ここで S をドメイン，$||*||$ を S に含まれる任意の要素間の距離メトリック，R と cR を距離に関する閾値，P_1 と P_2 を確率の下限と上限，$h(x)$ を S の要素 x のハッシュ値，Pr_H をハッシュ関数 H から生成されるハッシュ値に対する実現確率とすると，任意の $p, q \in S$ に対して以下が成立するとき，LSH 関数群 $H = \{h : S \to U\}$ は (R, cR, P_1, P_2)-sensitive である．

- $If ||p, q|| \leq R$ then $Pr_H[h(p) = h(q)] \geq P_1$
- $If ||p, q|| \geq cR$ then $Pr_H[h(p) = h(q)] \leq P_2$

本稿ではハッシュ関数 H を $h_{a,b}(v) = \lfloor \frac{a \cdot v + b}{\omega} \rfloor$ と定義する．ここで a は v と同じ次元のランダムなベクトルであり， b は $[0, \omega)$ の範囲のランダム値である [9]．得られたハッシュ値 $g(v) = (h_1(v), h_2(v), ..., h_k(v))$ をハッシュテーブルに直接格納すると，所要メモリが大きくなり，また検索が困難になる [11]．この問題を解決するため， E2LSH アルゴリズム [11] を使用し，次の 2 つのハッシュ関数を用いる．

$h_1 : Z^k \to \{0, ..., T-1\}$

$h_2 : Z^k \to \{0, ..., C\}$

ここで T はハッシュテーブルの大きさで，C は大きな素数である．h_1 をハッシュテーブルのインデックスとして，h_2 をリンクテーブルのキーとして使用し，以下のように定義する．

$$h_1(a_1, a_2, ..., a_k) = \left((\sum_{i=1}^{k} r_i a_i) mod\ C \right) mod\ T \quad (1)$$

$$h_2(a_1, a_2, ..., a_k) = (\sum_{i=1}^{k} r'_i a_i) mod\ C \quad (2)$$

ただし r_i と r'_i はランダムな整数で，C を $2^{32} - 5$ に設定する．

各ハッシュバケット g_i は z_k にマップされ，関数 h_1 は通常のハッシュポリシーのハッシュ関数であり，関数 h_2 はチェーンテーブル内のハッシュバケットを決定するために使用される．ハッシュバケット $g_i(v) = (x_1, x_2, ..., x_k)$ をチェーンテーブルに格納する際，ベクトル $(x_1, x_2, ..., x_k)$ ではなく，$h_2(a_1, a_2, ..., a_k)$ のフィンガープリントが格納される．この結果，ハッシュバケットの消費ストレージサイズを大幅に削減できる．

3.4 距離と人気度を考慮するキャッシュ制御

ネットワーク上に存在する複数のキャッシュサーバが各々，独立して自律的な判断で，画像データ到着時のキャッシュ置換処理を行うことを想定する．これまでに述べた方法で，キャッシュ内には複数のグループに分類された画像データがキャッシュされているとする．

ユーザが要求した画像データがキャッシュされている場合は，その画像をユーザにキャッシュから配信する (キャッシュヒット)．要求データがキャッシュ内に存在しない場合は，キャッシュ内に存在する画像データの中で，要求画像との特徴量との距離が最小となる，類似度が最大の画像データをユーザに配信する (ソフトキャッシュヒット)．この際，要求画像データと配信された画像データの特徴量の差異をキャッシュヒット距離と定義する．

キャッシュ内に存在する画像データ間の距離が大きい方が，多様な画像データをキャッシュに持たせ，ユーザの要求に対して，要求データと距離 (類似度) が近く，キャッシュヒット距離が小さい画像をキャッシュから送信できる可能性が向上する．そこで各画像に対しキャッシュ内の他画像との距離に関するコスト (C_d) を定義する．2 つの画像の距離は 2 つの画

像の特徴ベクトルのハミング距離 (Hamming distance) である [13]．挿入画像の特徴ベクトルと，グループの中心点との距離が最小のグループに，挿入画像を加え，このグループの中心点を更新する．グループの中心点はグループ内のすべての画像の特徴ベクトルの平均値である．そして中心点からグループ内の各画像の距離を計算する．中心点からの距離は近いほうが画像の冗長性が高いため，m 個の画像がキャッシュ内に存在するとき，i 番に冗長性が高い画像の距離コスト C_d を $C_d = (i-1)/(m-1)$ で定義する．

またキャッシュヒット率の向上ためには人気の高い画像を優先的にキャッシュ内に残すことが望ましいことから，m 個の画像がキャッシュ内に存在するとき，キャッシュ内の j 番目に人気の高い画像データに関するコスト C_p を $C_p = (j-1)/(m-1)$ で定義する．そしてこれら二つのコスト C_p と C_d を組み合わせ，画像の総コスト (C_{total}) を次式で定義する．

$$C_{total} = \omega * C_p + (1-\omega) * C_d \qquad (3)$$

ただし ω は C_p の重みである．画像を置換する際は，C_{total} が最大の画像を削減する．

4. 性 能 評 価

4.1 性能評価尺度

提案方式の有効性を確認するため，キャッシュヒット率 (cache hit ratio)，キャッシュ画像データ間の平均距離 (cache average distance)，キャッシュヒット距離 (cache hit distance) の 3 つの尺度を評価する．

4.1.1 キャッシュヒット率

キャッシュヒット率 η_h は，ユーザの要求画像そのものがキャッシュ内に存在 (キャッシュヒット) する確率で，シミュレーション期間中のキャッシュヒットの回数 H と，同じ期間中の総要求回数 R に対し，$\eta_h = H/R$ で定義する．要求データとの類似度を考慮しないで，要求されたデータのみを配信する従来のキャッシュの研究では，キャッシュヒット率がキャッシュ置換方式の効果に関する重要な指標である．キャッシュヒット率が高いほうが，キャッシュを有効に活用できるため望ましい．

4.1.2 キャッシュ平均距離

提案方式は 3.4 節で述べたように，キャッシュミス時にはキャッシュ内に存在する画像の中で，ユーザの要求画像との類似度が最大となる画像を配信する．そのため提案方式の有効性を測る尺度として，キャッシュ内に存在する任意の画像組の特徴量の差異の平均値 (キャッシュ平均距離)η_d を用いる．ここではキャッシュ平均距離を，キャッシュ内の画像の集合を $\boldsymbol{P}$，キャッシュ内の画像数を p，画像 x の特徴量ベクトルを v_x，キャッシュ内の全画像の特徴量の平均ベクトルを $\hat{v}$ とするとき，$\eta_d = \sum_{x \in \boldsymbol{P}} ||v_x, \hat{v}||/p$ で定義する．キャッシュ平均距離が大きなほど，キャッシュ中により多様な画像データが含まれることを意味し，キャッシュヒット距離の低減が期待できる．数値評価では，計算機シミュレーションの終了時点にて算出した結果を示す．

4.1.3 平均キャッシュヒット距離

提案方式は，ユーザの要求する画像に厳密に一致する画像がキャッシュ内に存在しない場合，キャッシュ内の画像データの中から要求画像に最も類似するものを配信する．そのためキャッシュの性能を測る，キャッシュヒット率に代わる評価尺度として，平均キャッシュヒット距離 η_{hd} を用いる．シミュレーション期間中に発生した要求の集合を $\boldsymbol{R}$，全要求数を r，要求 s の要求画像の特徴量ベクトルを v_s，要求 s に対して配信された画像の特徴量ベクトルを $\tilde{v}_s$ とするとき，η_{hd} を $\eta_{hd} = \sum_{s \in \boldsymbol{R}} ||v_s, \tilde{v}_s||/r$ で定義する．要求画像と完全に一致する画像がキャッシュ内に存在し，キャッシュヒットした場合は，$v_s = \tilde{v}_s$ となる．

さらに要求画像がキャッシュに存在せず，類似する画像を配信した場合 (ソフトキャッシュヒット) の，配信画像と要求画像との差異の平均値を平均ソフトキャッシュヒット距離 η_{shd} とし，シミュレーション期間中のソフトキャッシュヒットした要求の集合を $\boldsymbol{R_{sh}}$，全ソフトキャッシュヒット数を h_s とするとき，η_{shd} を $\eta_{shd} = \sum_{s \in \boldsymbol{R_{sh}}} ||v_s, \tilde{v}_s||/h_s$ で定義する．η_{hd} も η_{shd} も小さなほど，ユーザの要求画像と類似度の高い画像が配信されることを意味するため望ましい．

4.2 評 価 条 件

計算機シミュレーションでは，画像データとして，4 つのデータセットを用いる．データセット 1 は [15] から取得した 101 種類の画像 8,677 枚である．データセット 2 と 3 は [16] から取得した，各々，149 種類の画像 11,025 枚，および 105 種類の画像 6,312 枚である．データセット 4 は [17] から取得した 20 種類の画像 10,132 枚である．データセット 1，2，3 は異なる種類の画像から構成されるが，データセット 4 は類似した画像からなるデータセットである．これら各データセットの画像には約 100 のカテゴリの画像が含まれており，各データセットに含まれるカテゴリの例を表 1 にまとめる．

表 1　category of datasets

dataset1	accordion, airplanes, anchor, ant, ... barrel, bass, beaver, binocular, bonsai, ...
dataset2	hot-dog, hot-tub, hourglass, house-fly, ... ibis, ice-cream-cone, iguana, ipod, ... kangaroo, kayak, ketch, killer-whale, knife, ...
dataset3	ak47, american-flag, ... backpack, baseball-glove, bat, bear, ... cactus, cake, calculator, camel, ...
dataset4	Apple Braeburn, Apple Crimson.S, Apple Golden, ... Cantaloupe, Cherry Rainier, Cherry Wax Black, ...

単一のキャッシュサーバに対し，ユーザから 1 万回の要求を発生させ，各要求に対しパラメタ 0.8 の Zipf 分布 [14] に従う確率でランダムに要求画像を選択する．ただし各データセットの画像の人気順位はランダムに設定する．各データセットには約 100 の種類の画像が含まれ，キャッシュ内の画像を約 100 のグループに分けるために，E2LSH アルゴリズムでグループに分ける際のハッシュ関数 $h_{a,b}(v)$ のパラメタ ω を 100 に設定する．提案方式に加え，比較方式として，LRU と FIFO をキャッシュ置換法に用いた場合を評価する．

4.3 キャッシュヒット率

図 1 に，4 つの各データセットを用いた場合の各方式のキャッシュヒット率 η_h をキャッシュ容量に対してプロットする．ただし提案方式においては，2 つの評価尺度間のバランスを決める重み ω を 4 つの各値に設定した場合の結果を示す．提案方式は他のキャッシュ画像と類似度が高い画像を優先的に削除するため，高人気の画像を削除する可能性がある．そのため LRU や FIFO と比較して，提案方式のキャッシュヒット率は低下する．提案方式においては，人気度の重み ω が増加するほど，人気度がより高い画像がキャッシュに残る可能性が増加し，キャッシュヒット率が増加する．LRU は人気度を考慮しているため，人気度がより高い画像がキャッシュに残る可能性が FIFO より高く，LRU のキャッシュヒット率は最も高い．

4.4 キャッシュ平均距離

図 2 に 4 つの各データセットを用いた場合の各方式のキャッシュ平均距離 η_d をキャッシュ容量に対してプロットする．提案方式はキャッシュ内のデータの多様性を向上させるので，提案方式のキャッシュ平均距離は LRU や FIFO と比較して大幅

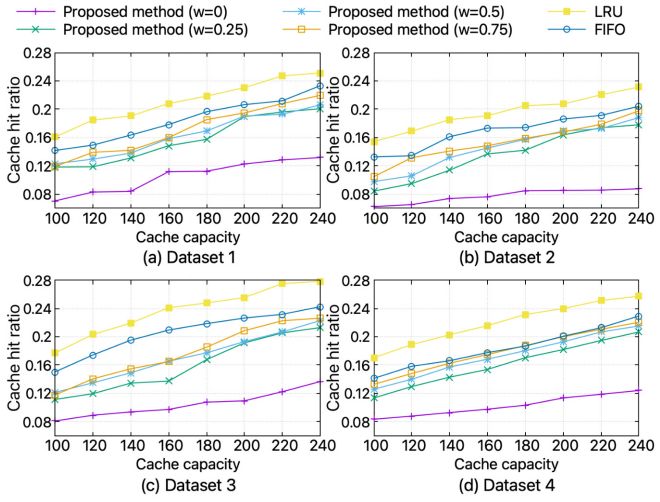

図 1　Cache hit ratio against cache capacity

に増加し，提案方式の有効性が確認できる．人気度の重み ω の増加に伴い，距離の重みが減少し，キャッシュに冗長な画像が存在する確率が増加し，キャッシュ平均距離は減少する．LRU は人気度を考慮しているため，似たような写真がキャッシュに残る可能性が FIFO より高く，LRU のキャッシュ平均距離は最も低い．キャッシュ容量の増加に伴い，より多数の画像がキャッシュ内に存在するため，提案方式のキャッシュ平均距離は緩やかに減少する．

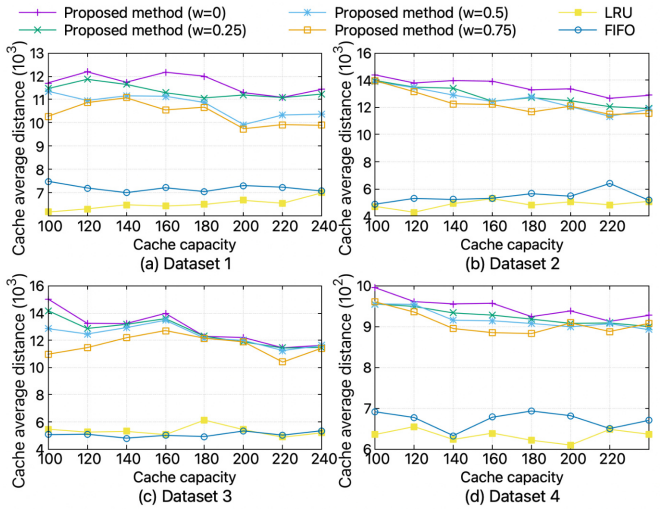

図 2　Cache average distance against cache capacity

4.5　平均キャッシュヒット距離

図 3 に 4 つの各データセットを用いた場合の各方式の平均キャッシュヒット距離 η_{hd} をキャッシュ容量に対してプロットする．提案方式はキャッシュ内のデータの多様性を向上させるので，提案方式の平均キャッシュヒット距離は LRU と FIFO より小さく，提案方式の有効性が確認できる．提案方式においては，人気度の重み ω の増加に伴い平均キャッシュヒット距離が減少する．前節で示したように，ω の増加に伴いキャッシュ平均距離は減少し，キャッシュデータの冗長性が増加するが，一方で高人気の画像がキャッシュ内に存在する可能性が向上する．平均キャッシュヒット距離の計算には，キャッシュヒット時の距離 0 の配信も含めるので，キャッシュヒット率増加の効果がより大きく，平均キャッシュヒット距離は ω の増加に伴い減少する．

類似画像から構成されるデータセット 4 を用いた場合，ω が 0 の提案方式の平均キャッシュヒット距離は LRU と FIFO より大きい．類似画像が多くキャッシュ内に存在する場合は，画像間の距離を考慮する効果が比較的小さい．提案方式は，多様な画像が存在する場合に，よりキャッシュヒット距離の低減効果が大きく，ユーザの要求画像と類似した画像をキャッシュから配信できる効果が高い．

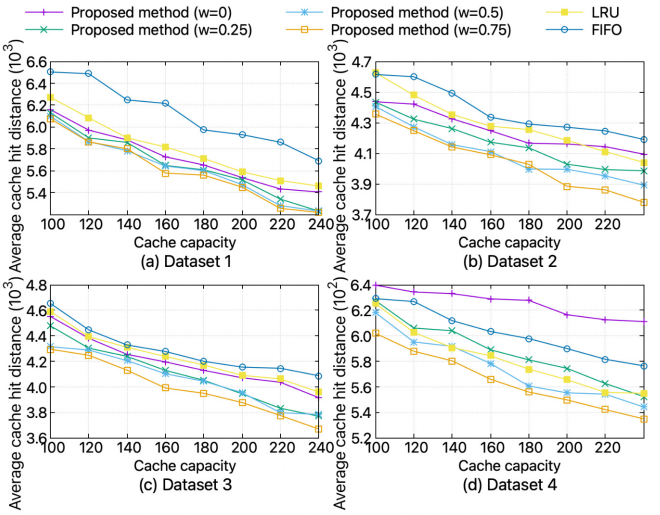

図 3　Average cache hit distance against cache capacity

4.6　平均ソフトキャッシュヒット距離

図 4 に 4 つの各データセットを用いた場合の各方式の平均ソフトキャッシュヒット距離 η_{shd} をキャッシュ容量に対してプロットする．平均ソフトキャッシュヒット距離の算出にはキャッシュヒット時の距離は含めないため，キャッシュ内に画像データの多様性が高いほうがソフトキャッシュヒット距離が小さい．そのため，人気度の重み ω が増加するほど距離の重みが減少し，キャッシュ内に多様性が減少するため，η_{shd} は増加する．キャッシュ画像の冗長性を考慮しない LRU と FIFO と比較して，提案方式は平均ソフトキャッシュヒット距離を大きく低減する．

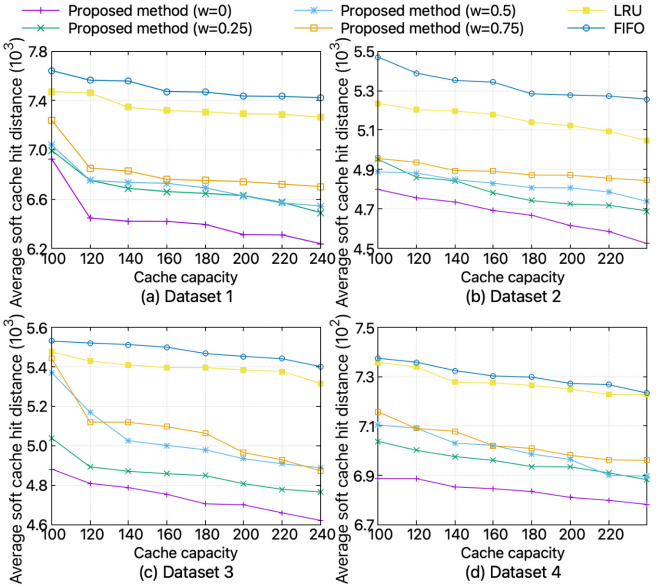

図 4　Average soft cache hit distance against cache capacity

4.7　複数のデータセットを混ぜた場合の結果

図 5 に，多様な画像データを含む 3 つのデータセット 1，2，3 の画像データを全て用いた場合の各方式の 4 つの評価尺度を各々，キャッシュ容量に対してプロットする．多様な画像データが混在し，画像の多様性が大きな場合，提案方式の，LRU や FIFO と比較した，キャッシュ平均距離の向上効果と，平均

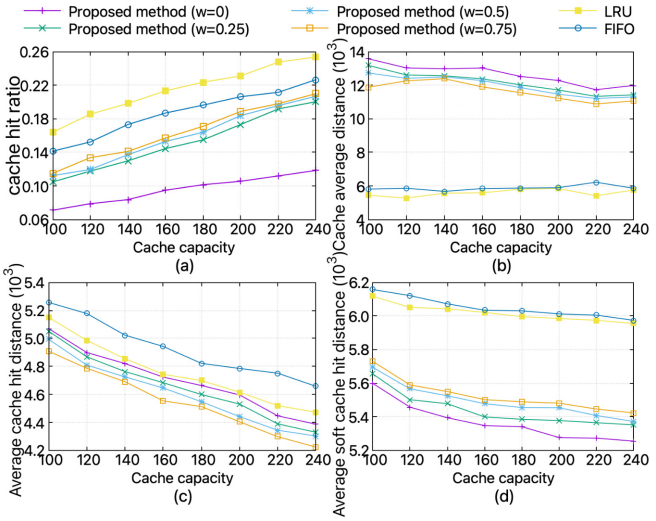

図 5　(a) Cache hit ratio, (b) cache average distance, (c) average cache hit distance, and (d) average soft cache hit distance, against cache capacity when mixing three datasets

ソフトキャッシュヒット距離の低減効果は，より顕著となる．

4.8　人気度重み ω の影響

図 6 に，データセット 1 を用いた提案方式の 3 つの評価尺度を，人気度の重み ω に対してプロットする．人気度の重み ω が増加するに従い，人気度がより高い画像がキャッシュに残る可能性が増加するため，キャッシュヒット率が増加する．一方で ω の増加に伴い距離の重みが減少するため，キャッシュ内の画像データの多様性が減少し，キャッシュ平均距離は減少する．平均キャッシュヒット距離はキャッシュヒット率とキャッシュ平均距離の両方の影響を受け，ω の増加に伴う増大要因と減少要因が存在するため，ω が 0 や 1 に近い両極端の場合は平均キャッシュヒット距離が増加する．キャッシュ容量が小さな場合はキャッシュヒット率の影響が大きく，ω が 0.8 程度のときに平均キャッシュヒット距離は最小となる．一方でキャッシュ容量が大きな場合はキャッシュ平均距離の影響が大きく，ω が 0.2 程度のときに平均キャッシュヒット距離は最小となる．

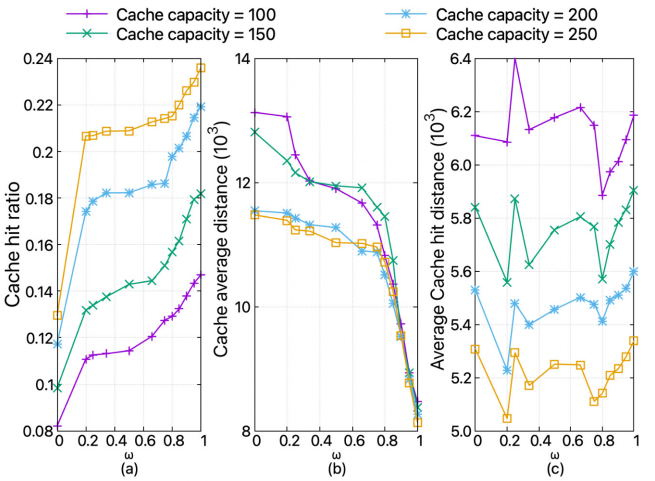

図 6　(a) Cache hit ratio, (b) cache average distance and (c) average cache hit distance against ω

5.　ま　と　め

本稿では MCP において，ユーザからの要求画像との類似度の高い画像データのヒット率の向上を目的に，キャッシュ画像セットの冗長性が低減するよう，冗長性の高い画像データを優先的にキャッシュから削除するキャッシュ置換法を提案した．冗長性の高い画像データ判別の計算処理量を抑制するため，キャッシュ内で画像を類似性に基づき複数のグループに分類する方式を提案した．そして実際の画像データセットを用いた計算機シミュレーション評価により，提案方式の有効性を明らかにした．

性能評価により，提案方式は多様な画像が存在する場合に，ユーザの要求画像と類似した画像をキャッシュから配信できる効果がより高くなること，また提案方式は画像データの冗長性と人気度の両方を考慮することで，キャッシュ配信される画像とユーザの要求画像データの類似度の向上が可能であることを明らかにした．

謝辞　本研究成果は，JSPS 科研費 21H03436 の助成を受けたものである．ここに記して謝意を表す．

文　　献

[1] P. Zuo, et al., Bandwidth and Energy Efficient Image Sharing for Situation Awareness in Disasters, IEEE Trans.Parallel and Dist. Sys., 30, 1, Jan. 2019

[2] Y. Hua, W. He, X. Liu, and D. Feng, SmartEye:Real-time and efficient cloud image sharing for disaster environments, IEEE INFOCOM 2015

[3] H. Chen, et al., A Generic Framework for Constraint-Driven Data Selection in Mobile Crowd Photographing, IEEE IoT Journal, 4, 1, Feb. 2017

[4] N. Dinh, et al., Sensing Content Correlation-aware In-network Caching Scheme at the Edge for Internet of Things, ACM ICN 2019

[5] G. Neglia, et al., SimilarityCaching: Theory and Algorithms, IEEE INFOCOM 2020

[6] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, Soft Cache Hits: Improving Performance through Recommendation and Delivery of Related Content, IEEE Journal on Selected Areas in Communications, vol. 36, no. 6,pp. 1300-1313, 2018

[7] D. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of ComputerVision, vol. 60, no. 2, pp. 91-110, 2004

[8] B. Bloom, “Space/time trade-offs in hash coding with allowable errors”, Communications of the ACM, vol. 13, no. 7, pp. 422-426, 1970

[9] A. Andoni, P. Indyk. E2lsh: Exact Euclidean locality-sensitive hashing. http://web.mit.edu/andoni/www/LSH/. 2004

[10] A. Andoni,et al., Practical and Optimal LSH for Angular Distance,NIPS 2015.

[11] P. Indyk and R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, STOC 1998

[12] Waggener, Bill. Pulse Code Modulation Techniques. Springer, ISBN 9780442014360. Retrieved 13 June 2020

[13] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, Locality-Sensitive Hashing Scheme Based on p-Stable Distributions, Symp. Computational Geometry 2004

[14] A. Broder and M. Mitzenmacher, Network applications of Bloom filters: a survey, Internet Mathematics, vol.1, pp.485-509, 2005

[15] Powers, David M. W., Applications and explanations of Zipf’s law., Joint conference on new methods in language processing and computational natural language learning. Association for Computational Linguistics. pp. 151 – 160.

[16] https://data.caltech.edu/records/mzrjq-6wc02

[17] https://data.caltech.edu/records/nyy15-4j048

[18] https://www.kaggle.com/datasets/moltean/fruits?resource=download