Vol.2017-CG-166 No.10
2017/3/13

二値特徴量のノルムでの並べ替えによる画像間の対応点探索の高速化

杉村昌彦[†1]　馬場孝之[†1]　上原祐介[†1]

概要：画像間の対応点探索は，画像や映像の検索などに重要な技術である．BRIEF などの二値特徴量を用いることで高速な探索が可能だが，それでも総当たりでの探索は処理時間が大きい．Locality Sensitive Hashing (LSH) などハッシュ法を用いた探索手法は高速化に有効である反面，メモリ消費量が大幅に増加するため適用が難しい場合がある．一方，特徴量をノルムの大きさ順に並べ替え，ノルムの近い特徴量の範囲で探索を行う Norm Ordered Matching (NOM) は，メモリ消費量がほとんど増加しない．しかし，二値特徴量のノルムは二項分布に従う傾向があるため，特徴量のビット数の半分の値の周辺に集中し，そこでの処理時間が大きくなる課題がある．本稿では，二値特徴量の一部のビット値を反転することで，特徴量間の距離を保存したまま，ノルムの分布を平坦化することで，処理を高速化する手法を提案する．評価実験により，提案手法が NOM に対して高速であり，LSH に対して，精度と速度の面で同等であり，メモリ消費量の観点で優れていることを示した．

キーワード：対応点探索，局所特徴量，二値特徴量，BRIEF，Norm Ordered Matching，Locality Sensitive Hashing

Fast Binary Descriptor Search for Image Matching by Norm Ordering

MASAHIKO SGUIMURA[†1]　TAKAYUKI BABA[†1]
YUSUKE UEHARA[†1]

1. はじめに

画像間の対応点探索は，画像や映像の認識や検索などコンピュータビジョンの様々な応用に重要な技術である．画像間の対応点探索の主なアプローチは，以下のプロセスによる．まず，二枚の画像から，等間隔に特徴点を決定する Dense Sampling やコーナーらしさを検出する FAST[1]などの特徴点検出よって，複数の特徴点を検出する．次に，SIFT[2]などの局所特徴量を特徴点周辺の部分的な画像から抽出する．そして，局所特徴量の特徴空間での距離を評価し，最近傍探索を行うことで，二枚の画像の特徴点の対応関係を求める．近年は，局所特徴量をバイナリコードで表現することでメモリ消費量の削減と，高速な距離計算を実現した，BRIEF[3]，BRISK[4]，ORB[5]などのさまざまな二値特徴量が提案されてきている．

しかし，高速な二値特徴量を用いたとしても，総当たりで距離計算を行う brute-force matching での最近傍探索は処理時間が大きく，実用的ではない場合が多い．この課題を解決する有効な手段として，Locality Sensitive Hashing (LSH) [6]に代表されるハッシュ法を用いた探索手法が提案されてきた．LSH は，特徴量にハッシュ関数を適用し，同じハッシュ値に分類された特徴量の範囲で探索を行うことで，高速に最近傍探索の近似解を得ることができる．しかし，LSH には精度向上に伴ってメモリ消費量が大幅に増加するというデメリットがあり，適用が難しい場合がある．

†1 (株)富士通研究所
FUJITSU LABORATORIES LTD.

一方，特徴量をノルムの大きさ順に並べ替え，ノルムの近い特徴量の範囲で探索を行う Norm Ordered Matching (NOM) [7]は，LSH ほどの高速化の効果は得られないが，メモリ消費量がほとんど増加しないというメリットがある．

本稿では，NOM の二値特徴量への適用を試みた．その際，二値特徴量のノルムが特徴量のビット数の半分の値の周辺に集中する傾向と，その結果として，そこでの距離計算の回数が多くなり，処理時間が大きくなるという課題が見えてきた．これに対して，ノルムの分布を効果的に平坦化することで，距離計算の回数を削減する Equalized NOM (ENOM)を提案し，その効果を確認した．

2. 関連研究

近年，BRIEF，BRISK，ORB など，様々な二値特徴量が提案され，画像間の対応点探索に用いられてきた．二値特徴量は，従来の SIFT などの実数ベクトルでの特徴量に比べ，メモリ消費量が小さく，高速に特徴抽出でき，ハミング距離計算により高速に距離計算できる点に優位性がある．

NOM を二値特徴量へ適用した研究として，BRISK へハッシュ法を用いた探索手法を適用し，同じハッシュ値に分類された特徴量同士の探索に NOM を適用した試みがある[8]．しかし，NOM を適用に伴う上述の課題には対処していない．

3. LSH と NOM の BRIEF への適用

3.1 BRIEF

本稿では，最も基本的な二値特徴量である BRIEF[3]を扱

ⓒ2017 Information Processing Society of Japan

う．BRIEF に有効な手法は，他の二値特徴量に対しても有効性が期待できると考える．BRIEF では，特徴点周辺の矩形（パッチ）の中に，全てのパッチに共通の配置で画素の組を 128 組から 512 組設定する．そして，この画素の組の輝度差の符号をビット値（0 または 1）に割り当て，バイナリコードを生成し特徴量とする．特徴量のビット数は画素の組の数と同じであり，128 ビットから 512 ビットとなる．輝度差を計算するだけで済むため，非常に高速な特徴抽出が可能である．本稿では，文献[3]に基づきパッチのサイズを縦横 48 画素，特徴量のビット数を 128 ビットとした．

3.2 LSH の適用

LSH[6]を BRIEF による画像間の対応点探索に適用する場合の処理を以下に示す．

(1) ハッシュ関数による特徴量の分類

特徴量を表すバイナリコードの一部を切り出したバイナリコードを生成しハッシュ関数とする．ハッシュ関数が取る値をハッシュ値として，それぞれの画像の特徴量をハッシュ値毎に分類する．分類の細かさはハッシュ関数の長さで決まり，ハッシュ関数が 8 ビットの場合，256 種類のハッシュ値に特徴量を分類する．

(2) 探索

探索に際しては，それぞれの画像から同じハッシュ値を持つ特徴量を選択し，その範囲で，最も距離が小さい特徴量を探索する．

一般に，複数のハッシュ関数を定義し，いずれかのハッシュ関数によるハッシュ値が一致する特徴量の範囲で探索を行うことで，探索範囲を限定しすぎることによる誤対応を軽減する．

LSH の調整可能なパラメータは，ハッシュ関数の長さと，ハッシュ関数の数であり，これらを調整することで，精度と速度のトレードオフを制御できる．本稿では，ハッシュ関数の長さを予備評価で最も性能が高かった 8 に固定し，ハッシュ関数の数のみを変化させた．

3.3 NOM の適用

NOM[7]を BRIEF による画像間の対応点探索に適用する場合の処理を以下に示す．

(1) ノルムによる特徴量の並べ替え

それぞれの画像の特徴量に関して，ノルムを計算し，その大きさ順に並べ替える．二値特徴量の場合，ノルムは特徴量に含まれる 1 の数の合計であり，高速に計算できる．また，ノルムの値は，0 から特徴量のビット数（ここでは 128）の範囲の整数となる．

(2) 探索

探索に際しては，一方の画像の特徴量に対して，もう一方の画像の特徴量のうち，ノルムの差が一定の探索範囲に含まれる特徴量を選択し，その範囲で，最も距離が小さい特徴量を探索する．どの範囲を取っても，特徴量はメモリ上に順番に格納されているため，探索は高速に行える．

NOM の調整可能なパラメータは探索範囲であり，これを調整することで，精度と速度のトレードオフを制御できる．

3.4 LSH と NOM の比較

LSH と NOM の性能を比較するため，評価を行った．評価画像としては，画像間の対応関係の正解値が提供されている公開画像[9]から，平行移動（cars, trees, bikes），明度変化（cars），ぼかしの有無（trees, bikes）を含む 3 組を用いた（図 1，図 2，図 3）．画像サイズは，cars が横幅 900 画素，高さ 600 画素，trees と bikes が横幅 1,000 画素，高さ 700 画素である．

図 1　評価画像 cars1, cars2

図 2　評価画像 trees1, trees2

図 3　評価画像 bikes1, bikes2

特徴点を 6 画素毎に等間隔に検出し，cars から各 12,831 点，trees と bikes から各 16,799 点を得た．これらの特徴点から 128 ビットの BRIEF を抽出し，brute-force matching (以下 BF)，LSH，NOM を用いて画像間の対応点探索を実施した．算出された特徴点の対応関係と正解の画像間の対応関係の誤差が 6 画素以内であれば，特徴点の対応関係を正解とし，正解数を上述の特徴点の数で割ったものを正解率とした．

LSH のハッシュ関数の数を 4，8，12，16 の 4 段階，NOM の探索範囲を±6，8，10，12 の 4 段階に変化させ，それぞれの正解率，処理時間を計測した．また，それぞれのメモリ消費量を見積もった．環境としては，Xeon E3-1246 v3

ⓒ2017 Information Processing Society of Japan

3.5GHz とメモリ 16GHz を搭載した計算機上で，CPU をシングルコアで利用し，バイナリコードの距離計算を高速に実行するため SSE4.2 のビットカウント命令を利用した．cars の正解率と処理時間の関係を図 4 に，正解率とメモリ消費量の関係を図 5 に示す．trees と bikes に関しては，cars とほぼ同じ傾向であるため，省略する．

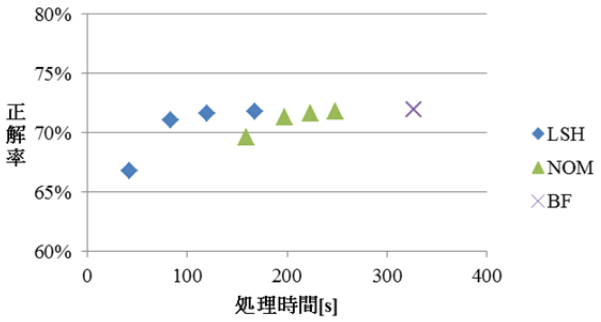

図 4　正解率と処理時間の関係（cars）

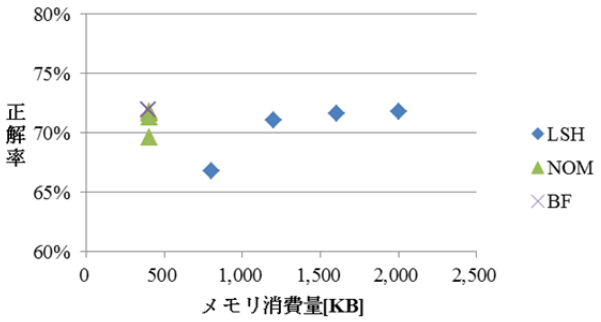

図 5　正解率とメモリ消費量の関係（cars）

図 4 から，正解率と処理時間の観点で，LSH が最も高性能で，NOM は BF と LSH の中間的な性能であることがわかる．

また，図 5 からわかるように，LSH は，正解率の向上に伴い，メモリ消費量が BF に比較して 2 倍から 5 倍に増加する．これは，LSH では，全ての特徴量をハッシュ値毎に分類する必要があるため，この情報を格納するためのメモリが，特徴量の数に比例して必要となり，さらに，ハッシュ関数の数に比例して，その量が増加するためである．一方，NOM のメモリ消費量は精度に関係なく BF とほとんど変わらない．これは，NOM では，特徴量を並べ替えるだけなので，処理に伴う余分なデータをほとんど必要としないためである．

この結果を，実用面から検討してみる．例えば，画像間の対応点探索を用いた類似画像検索システムを構築し，データベースに 1,000 万枚の画像を持つ場合を想定する．今回の BF でのメモリ消費量が 200KB/枚程度であることから概算すると，必要なストレージは，BF の場合 2TB で済むのに対して，LSH を適用する場合には 10TB 必要になることがわかる．また，ストレージからメインメモリへのデータ転送にも，概して 2 倍から 5 倍の時間がかかることになる．これらは，システム全体でのコストと性能に大きく影響する．また，別の例として FPGA などのハードウェアに画像間の対応点探索を実装し高速化する場合を考える[10]．ハードウェアによる高速化の効果は，データ転送や集積度の観点で，扱うデータ量の影響を大きく受けるため，LSH を用いる場合，十分な高速化の効果を得られない可能性がある．

一方，NOM は，基本性能では LSH に劣るが，メモリ消費量の増加を懸念することなく BF を置き換えることができるため，様々な場面で利用できる実用的な手法であると言える．

4. Equalized NOM (ENOM)の提案

上の結果を踏まえ，本稿では，NOM のメリットを維持しつつ，精度と速度を改善することを狙い，Equalized NOM (ENOM)を提案する．

4.1 NOM の課題

まず，NOM を BRIEF による画像間の対応点探索に適用した場合の課題を述べる．図 6 に，cars1 の特徴量のノルムの分布を示す．ノルムが取りうる範囲 0 から 128 に対して，64 の周辺に値が集中していることがわかる．

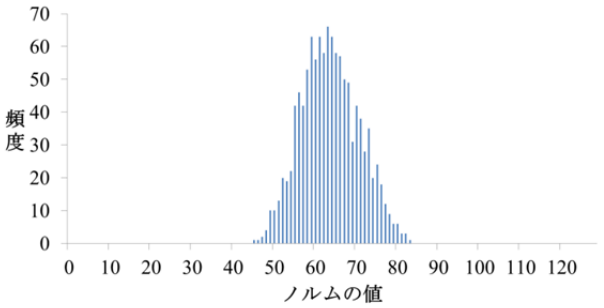

図 6　特徴量のノルムの分布（cars1）

これは，BRIEF の仕組みを考えると自然なことである．前述したように，BRIEF はパッチ内に配置された複数の画素の組の輝度差の符号から生成されるバイナリコードである．対象となる画像が十分複雑で，パッチ内の画像が十分なバリエーションを持つ場合，輝度差の符号は約 50%の確率でプラスまたはマイナスになる．同様に，そこから生成されるバイナリコードのそれぞれのビット値も約 50%の確率で 0 または 1 になる．その結果，バイナリコードのビット値の和であるノルムの値は 2 項分布に従い，バイナリコードの長さの半分に値が集中する．

二枚の画像から抽出した特徴量のノルムの傾向がこれに従う場合，ノルムの値で範囲を限定して探索しても，ノルムの値 64 の周辺での組み合わせが多大になるため，大き

ⓒ2017 Information Processing Society of Japan

な処理時間がかかることになる．

4.2 基本的なアイデア

これを抑制するため，NOM の改良版である ENOM を提案する．ENOM の基本的なアイデアは，全ての特徴量に含まれる全てのビット値に関して，本来 50%程度である 1 の頻度を減らす（または増やす）ことで，64 周辺に分布している特徴量のノルムを小さい（または大きい）方向へ移動することで，ノルムの分布を平坦化することである．

ENOM では，全ての特徴量に対して，同じビット位置の値を反転することで，これを実現する．この方法の良い点は，二値特徴量の性質上，この操作を行っても，特徴量同士の距離が変化しないことである．これにより，画像間の対応点探索の精度にほとんど影響を与えることなく，ノルムの分布を平坦化することができる．また，特徴量のビット値を反転するだけなので，メモリ消費量の増加はほとんどなく，NOM のメリットを維持することができる．

4.3 アルゴリズム

ENOM は，上述の NOM の処理の前に，以下の平坦化処理を追加することで実現できる．

(1) **1 の数の集計**

対象とする全ての画像中の全ての特徴量を通して，バイナリコード内のビット位置毎に 1 の数を集計する．

(2) **反転するビット位置の決定**

1 の数が特徴量の数の半分より多い，つまり，全ての特徴量を通して 1 の発生頻度が 50%を超えているビット位置を特定する．

(3) **値の反転**

全ての特徴量に対して，(2)で決定したビット位置の値を反転する．

上で生成した特徴量を用いて，NOM を実行することで，ENOM を実施できる．

ENOM を画像検索などへ適用する場合は，画像データベースに格納された画像の特徴量に対してビット位置の決定と値の反転を行い，クエリ画像が与えられた際には，クエリ画像の特徴量に対しても同じビット位置の値を反転させれば良い．

上の平坦化処理を cars1 に適用した際の特徴量のノルムの分布の変化を図 7 に示す．大幅に平坦化できていることがわかる．

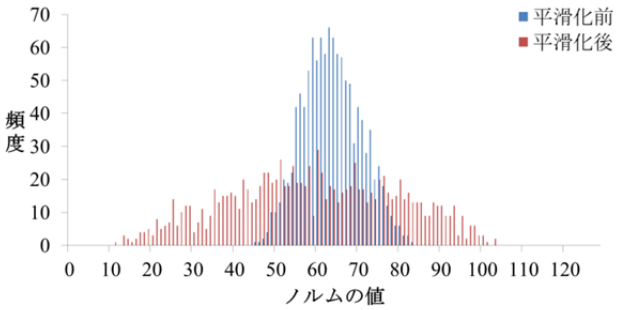

図 7　平坦化前後の特徴量のノルム（cars1）

5. 評価と結果

ENOM の効果を示すため，画像間の対応点探索と，それを用いた類似画像検索のタスクで評価を行った．どちらのタスクでも，ENOM は精度と速度の面で LSH と同等の性能を達成している．

5.1 画像間の対応点探索

cars, trees, bikes の 3 組の評価画像に対して，上述の BF, LSH, NOM の適用に加え，ENOM を適用し，正解率と処理時間を計測した．ENOM の探索範囲は±6，8，10，12, 14, 16 の 6 段階に変化させた．それぞれの結果を，図 8，図 9，図 10 に示す．どの場合でも，ENOM は NOM よりも二倍程度高速化できている．また，trees において若干 LSH に劣るもの，概ね LSH と同等の性能を示している．

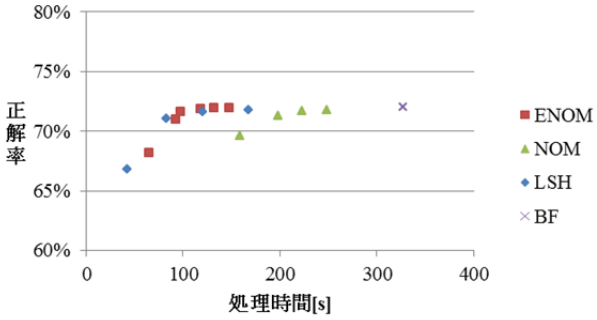

図 8　正解率と処理時間（cars）

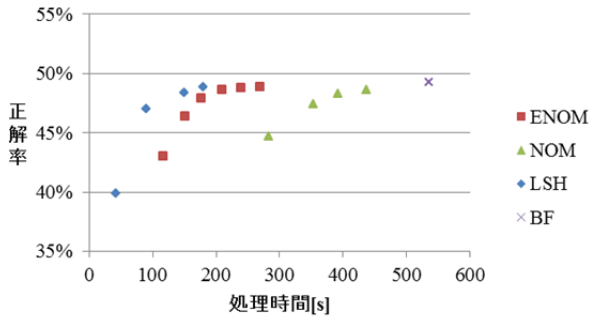

図 9　正解率と処理時間（trees）

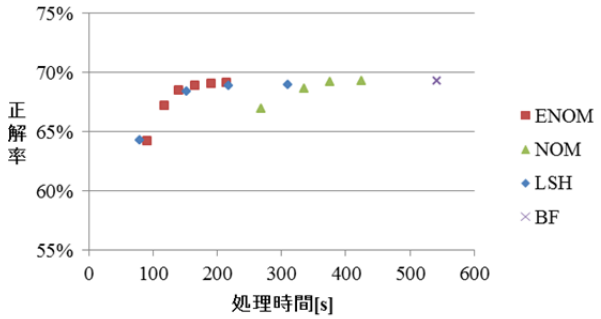

図 10　正解率と処理時間（bikes）

5.2 類似画像検索

次に，類似画像検索のタスクでの評価を行った．建物や景色などの観光地 708 種類の写真 708 枚をデータベース内の画像とし，異なる時に撮影された 37 種類の観光地の写真 400 枚（1 種類につき 10 枚程度）をそれぞれクエリ画像とした．

検索手法としては，クエリ画像とデータベース内の画像との対応点探索を行い，対応する特徴点の配置の整合性を評価することで類似度を算出し，類似度の大きい順にデータベース内の画像の順位を決定した．

図 11 に，BF，LSH，ENOM を適用した際の正解率と処理時間を示す．正解率は，正解の観光地の画像が 1 位に検索されたクエリ画像の数を，クエリ画像の総数 400 で割った値である．また，処理時間は，クエリ画像 400 枚とデータベース内の画像 708 枚の組み合わせ 283,200 回の画像間の対応点探索にかかった時間である．この図から，ENOM と LSH がほぼ同等の性能を示していることがわかる．

また，データベース内の画像 708 枚に関して，ENOM の平坦化処理を行う前後の特徴量のノルムの分布を図 12 に示す．データベース内の画像に対して，効果的に平坦化できていることがわかる．

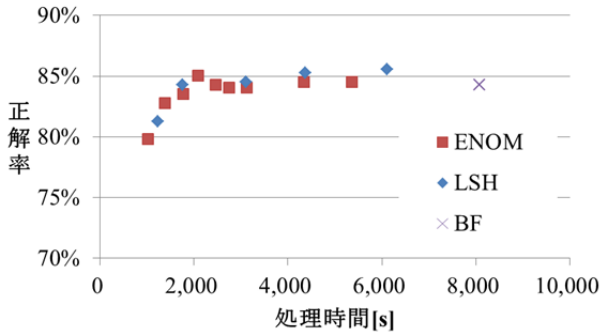

図 11　類似画像検索による評価結果

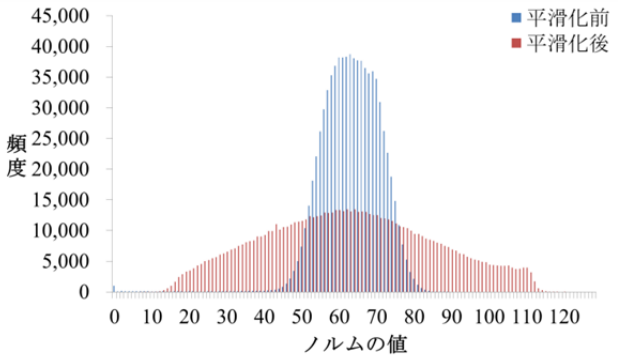

図 12　データベース内の画像に対する平坦化の効果

6. おわりに

本稿では二値特徴量による画像間の対応点探索を高速化するために，ENOM を提案した．ノルムが特徴量のビット数の半分の値の周辺に集中することにより処理時間が大きくなる NOM の課題に対して，二値特徴量の一部のビット値を反転することでノルムの分布を平坦化し，処理を高速化する手法を提案した．評価実験により，提案手法 ENOM が，NOM に対して二倍程度高速であることを示した．また，既存手法として有力な LSH に対して，精度と速度の面で同等であり，また，メモリ消費量の観点で優れていることを示した．

今後の課題としては，本手法が有効に利用できる条件と他の手法との使い分けの基準の明確化，BRIEF 以外の二値特徴量での効果検証，さらに，LSH 以外の探索手法との比較評価などがある．

参考文献

[1] Rosten, E and Drummond, T.. Machine Learning for High-speed Corner Detection. European Conference on Computer Vision (ECCV). 2006, pp. 430-443.
[2] Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 2004, vol. 60, no. 2, pp. 91-110.
[3] Calonder, M., Lepetit, V., Strecha, C., and Fua, P.. Brief: Binary robust independent elementary feature. European Conference on Computer Vision (ECCV). 2010, pp. 778-797.
[4] Leutenegger, S., Chli, M., and Siegwart, R. Y.. BRISK: Binary Robust Invariant Scalable Keypoints. International Conference on Computer Vision (ICCV). 2011, pp. 2548-2555.
[5] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G.. ORB: An efficient alternative to SIFT or SURF. Conference on Computer Vision and Pattern Recognition (CVPR). 2012, pp. 510-517.
[6] Indyk, P. and Motwani, R.. Approximate nearest neighbors: Towards removing the curse of dimensionality. Proc. 30th ACM Symposium on Theory of Computing. 1998, pp. 604-613
[7] Yousef, M. and Hussain, K. F.. Fast Exhaustive-Search equivalent pattern matching through hierarchical partitioning. Journal of Visual Communication and Image Representation. 2013, vol. 24, no. 5, pp. 592-601.
[8] Kamel, A., Mahdi, Y. B., and Hussain, K. F.. Multi-Bin search: improved large-scale content-based image retrieval. International

ⓒ2017 Information Processing Society of Japan

Journal of Multimedia information Retrieval. 2015, vol. 4, Issue 3, pp. 205-216.
[9] “Affine Covariant Features”. http://www.robots.ox.ac.uk/~vgg/research/affine, (参照 2017-02-18).
[10] Matsumura, H., Sugimura, M., Yamasaki, H., Tomita, Y., Baba, and T., Watanabe, Y.. An FPGA-accelerated partial duplicate image retrieval engine for a document search system. Applications of Computer Vision (WACV). 2016, pp. 1-7.

ⓒ2017 Information Processing Society of Japan