Vol.2018-MPS-119 No.5
2018/7/30

類似の画像ファイルに対するデータ重複排除

小池 到[†1]　モハマド ノールシャフィカ[†1]　木下 俊之[†1]

近年，マルチメディアデータの普及に伴ってファイルのデータ量が増加し，データの重複部分をひとつにしてデータ量を削減する重複排除技術が用いられている．重複排除技術では対象ファイルを複数の領域に分割し（各領域はブロックと呼ばれる），ブロックごとに重複を見つけて重複しているブロックは削除される．可変長ブロック方式により，データの挿入や削除があっても重複排除の効果を維持することができる．

本研究では，類似度を用いて抽出した類似ファイルに対してのみ重複排除を行う類似ファイル抽出法を提案した．類似度は平均ハッシュ法を用いて計算する．類似度の低いファイルに対して重複排除を行わないことにより，重複排除の効果の低下を抑えながら重複排除の実行時間を短縮させることができる．90 個の画像ファイルを対象とした実験により，類似度が 0.75 以上の類似ファイルを抽出して重複排除を行うと重複排除によるデータ削減量の減り方は約 7.9%に抑えられて重複排除の実行時間を約 18.6%短縮できることを確認した．

Data Deduplication for Similar Image Files

Itaru Koike[†1]　Mohamad Zaini Nurshafiqha[†1]　Toshiyuki Kinoshita[†1]

Recently, massive data growth and duplicate data in enterprise systems have led to use of deduplication technique. The deduplication technique is a powerful storage minimization technique that can be adopted to manage maintenance issues in data growth. The target files for deduplication are divided into several parts (each part is called a block) and any duplicate blocks are eliminated. By using the variable-length block method, the effect of deduplication can be maintained.

In this research, we extracted files with high similarity by using file similarity, and proposed “similar file extraction method” that deduplicates only for these similar files. By deduplicating only files with high similarity, the execution time for deduplication can be minimized without reducing the effect of deduplication too much. The average hash method is used to determine the similarity in target image files. Experimental results clarified that the reduction of deduplication rate is suppressed to about −9% while the execution time for deduplication can be shortened to about −18% when deduplication is performed on files for similarity of 0.75 or more,

1. はじめに

近年，オーディオ，アニメーション，ビデオなどのマルチメディアデータの普及に伴って，エンタープライズシステムにおけるファイルのデータ量は大幅に増加している．これらのマルチメディアファイルには，厳密にまたは殆ど一致しているファイルが大量に存在している可能性がある．そこで，この重複したデータをひとつにまとめてデータ量を削減する重複排除技術が有効である．重複排除技術は，大規模なサーバやデータストレージにおいて重複するデータ（冗長データ）を排除することで膨大な量のデータを削減でき，ファイルバックアップ，仮想マシンストレージ，WAN 複製などに適用されている．

重複排除技術では，複数のレコードの一致を検出してそのうちのひとつを代表データとして保存し，他のデータは代表データを指し示すリンクに置き換えられる（図 1 参照）．こうして重複したデータをリンクに置き換えることにより，全体のファイルサイズを大幅に削減できる．その結果，データ保存の効率を大幅に向上させることができ，データの保守にかかるコストを削減できる．

本研究では，画像ファイルを対象にして類似したファイルを抽出し，類似ファイルに対してのみ重複排除を適用することで処理を効率化する類似ファイル抽出法を提案した．[10]では，類似ファイル抽出法をテキストファイルに適用した場合について報告した．テキストファイルでは，形態素解析によりセンテンスを単語に分解し，コサイン類似度を用いて類似ファイルを抽出した．本研究では，画像ファ

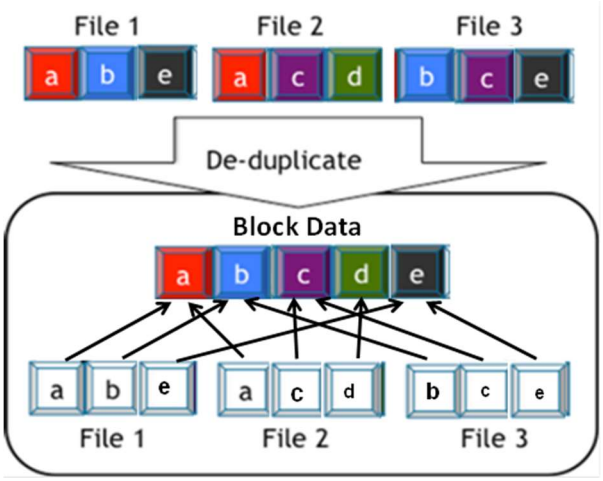
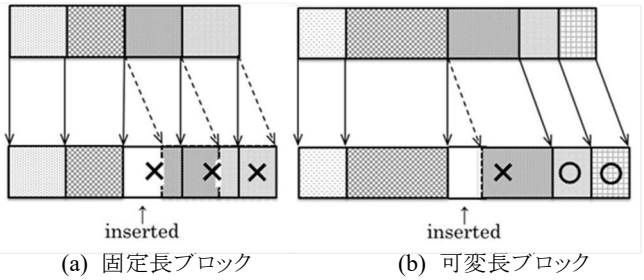


図 1 重複排除の考え方

†1 東京工科大学コンピュータサイエンス学部
School of Computer Science, Tokyo University of Technology



(a) 固定長ブロック　(b) 可変長ブロック

図 2　2 種類のブロック

イルを対象に平均ハッシュ法を用いてファイル類似度を計算した．実験により，提案した画像ファイルに対する類似ファイル抽出法は，重複排除の効果の低下を抑えながら重複排除の実行時間を短縮するのに有効な方法であることを確認した．

2. 2 種類のブロック

重複排除技術では，対象のファイルをいくつかの部分に分割し（各部分をブロックと呼ぶ），ブロックごとに重複排除を行う．ブロックごとに行うことにより，ファイル全体が一致していなくても重複排除を行うことができる．

ブロックには，長さが一定の固定長ブロックと長さが変更可能な可変長ブロックがある．固定長ブロック方式は，処理は簡単であるが，データの挿入や削除が行われるとデータがずれて元のファイルでは一致していたブロックを一致しているとは認識できなくなる（図 2（a））．一方，可変長ブロック方式は，処理は複雑になるが，ブロック長を調整することで挿入や削除によって位置がずれても重複ブロックとして認識することができる（図 2（b））．

図 3 に，可変長ブロックの境界を生成する方法を示す．まず，ウィンドウと呼ばれる一定長の小さな領域に対してハッシュ値が計算される．ウィンドウは，ブロックの境界の候補となる．指定された特定のビットパターン（このビットパターンを特異点と呼ぶ）がウィンドウのハッシュ値に含まれていれば，そのウィンドウの位置がブロックの境界となって新たなブロックが生成される．特異点のビットパターンがハッシュ値に含まれていなければ，その位置ではブロックは生成されず，ウィンドウを 1 バイトずらして再度境界の探索が行われる．重複排除の効果は，特異点のサイズに影響を受ける．我々の以前の研究 [5]で，最適なウィンドウサイズは約 32 バイトであり，最適な特異点サイズは 14～15 ビットであることを明らかにした．本研究においても，特異点サイズとウィンドウサイズにはこの値を使用した．

ウィンドウのハッシュ値に特異点のビットパターンが含まれているかどうかを判定するために，文字列検索アルゴリズムのひとつである Rabin-Karp アルゴリズムを用いる．可変長ブロック方式では，極端に大きなブロックや極端に小さいブロックが生成されないようにするために，最大ブロック長と最小ブロック長が設定される．ファイルが最小ブロック長よりも小さい場合は，ファイル全体がひとつのブロックとして生成される．次のブロックとの境界が最大ブロック長まで見つからなければ，最大長のブロックが生成される．重複排除は，この最大ブロック長および最小ブロック長の影響を受ける．

3. 関連研究

ブロック長を 4,000～16,000 バイト（最小ブロック長が 4,000 バイトで最大ブロック長が 16,000 バイト）とした時の固定長ブロック方式による重複排除の効果は[1]で検討され，可変長ブロック方式による効果は[3]で議論された．最小ブロック長が 4,000 バイトより大きい場合は[2]で報告された．[4]では，固定長ブロック方式と可変長ブロック方式を組み合わせた二段階重複排除法が提案された．

我々の前回の研究[5]では，特異点サイズと重複排除率の関係を分析した．[6]では，ファームウェアファイルとオーディオデータファイルに対する重複排除の効果を調べた．[7]では，本研究と同じ類似ファイル抽出法をテキストファイルの重複排除に適用した場合について検討した．

4. 類似ファイル抽出法

重複排除の対象ファイル数が多いほど重複部分をより多く見つけられて重複排除の効果を高めることができるが，ファイル数の増加は重複を見つけるための実行時間の増加

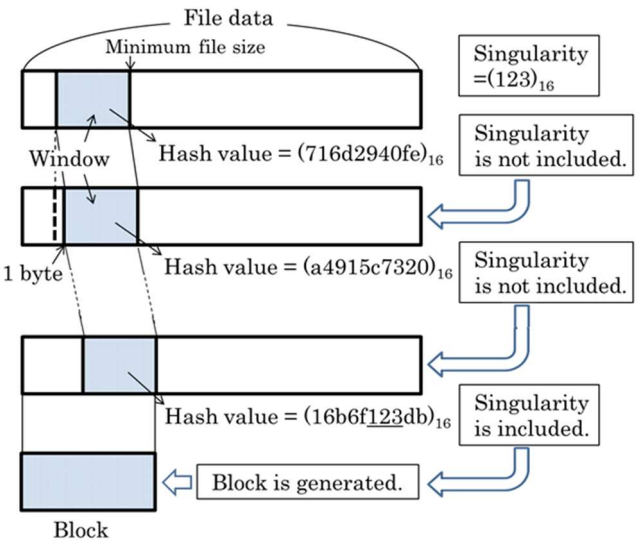


図 3　ブロックの境界探索

を引き起こす．本研究では，対象ファイル群から類似度の高いファイルを抽出し，抽出したファイルに対してのみ重複排除を行う類似ファイル抽出法を提案した．類似度の低いファイルには重複データは少ないと考えられるので，これを重複排除の対象から除いても重複排除の効果が大きく低下することはないと考えられる．そこで類似度の低いファイルを除いて類似度の高いファイルに対してのみ重複排除を行うことで，重複排除の効果の低下を抑えながら重複排除の実行時間の短縮を図る．

類似ファイルを抽出するために，画像ファイルが互いにどの程度類似しているかを示すファイル類似度を算出する．ファイル類似度は平均ハッシュ法を用いて計算する． 図 4 に平均ハッシュ法によるファイル類似度の計算手順を示す．まず対象とするファイルの双方をグレースケール（白黒）にし，8×8 のメッシュで 64 個の正方形に分割する．次に各正方形が白っぽいか黒っぽいかによって 1 または 0 のハッシュ値を割り当て（各正方形の輝度が閾値に達していれば 1，達していなければ 0），双方のファイルの対応する正方形のハッシュ値の排他的論理和をとる（ハッシュ値が一致していれば 1，一致していなければ 0）．排他的論理和が 1 の個数は双方のファイルの白黒の傾向が似ている正方形の個数だから，この 1 の個数を正方形の総数（＝64 個）で割った値を両ファイルの類似度とする．

5. 実験結果

実験は，bmp 形式の 90 個の画像ファイルを対象として行った．その中には，犬，猫，海，山などの動物や風景，静物などの様々な種類の画像が含まれる．実験には次のパラメータを使用した．

・画像ファイル数：90 個
・画像フォーマット：bmp
・最大ブロックサイズ：4,000 バイト
・最小ブロックサイズ：500 バイト
・特異点サイズ：14 ビット

平均ハッシュ法により，対象とした 90 個の画像ファイルのファイル類似度を求めるとすべて 0.60 以上になる．図 5 に，類似度が 0.60 以上から 0.90 以上まで 0.05 刻みで 7 グループを抽出した結果を示す．重複排除は各グループに属するファイルに対してのみ行う．当然，類似度が大きくなるにつれて画像の数は減少する．類似度が 0.60 以上のグループには 90 個のすべての画像が含まれ，類似度が 0.70 以上では 82 個が，0.75 以上では 67 個が含まれる．類似度が 0.75 と 0.80 の間で，含まれる画像の数はほぼ半分に減少する．類似度が 0.95 以上では 7 個の画像だけとなり，それらは殆ど同じパターンまたは形状を有する．図 6 の 2 つの画像は，類似度 0.95 の画像の例である．

表 1 に，平均ハッシュ法による類似度に基づくグループ毎の重複排除結果を示す．重複排除の効果は重複排除率とデータ削減量によって示される．類似度が 0.60 以上のグループの重複排除率は 6.56%，0.70 以上では 6.61%であった．類似度が 0.75 以上では重複排除率は 7.25%になり，これは最高の重複排除率である．この類似度 0.75 以上のグループでは重複排除によるデータ削減量（重複排除の効果）は約 7.9% 減るが，重複排除の実行時間は約 18.6% 短縮される．これは，重複排除の効果の低下を抑えながら重複排除の実行時間を短くするという類似ファイル抽出法の目的が，この範囲で達成されていることを示している．一方，類似度が 0.8 以上のグループでは重複排除率は 5.61%に低下し，0.85 以上，0.90 以上，0.95 以上では重複排除率は 1.0%未満と大幅に低下する．

図 7 に，類似度と重複排除によるデータ削減量，重複排除の実行時間の比較を示す．類似度が増加するにつれて，重複排除によるデータ削減量と重複排除の実行時間ともに減少する．類似度が 0.70 以上と 0.75 以上のグループでは，重複排除によるデータ削減量の減り方よりも重複排除の実行時間の短縮効果の方が大きい．これは，類似ファイル抽出法がこの範囲で有効であることを示している．

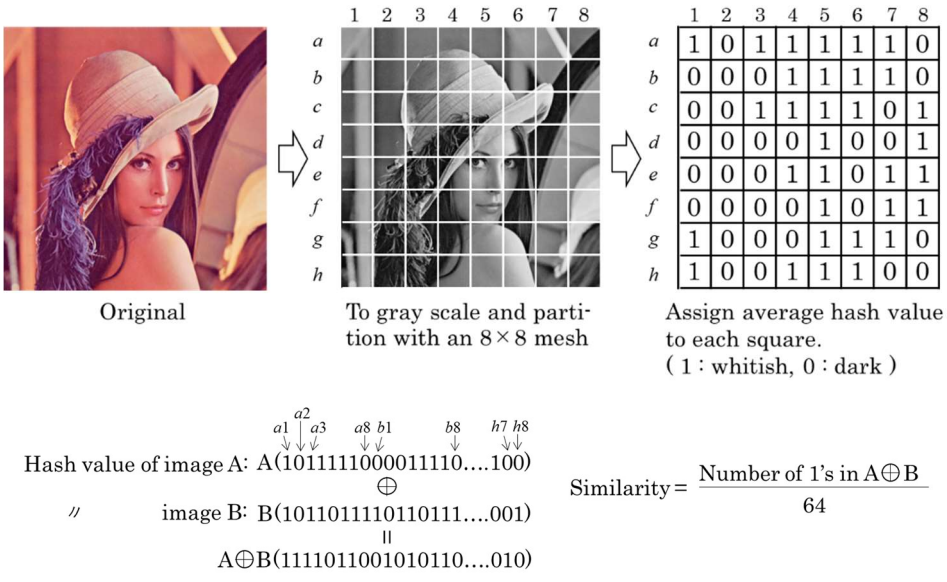


図 4　平均ハッシュ法

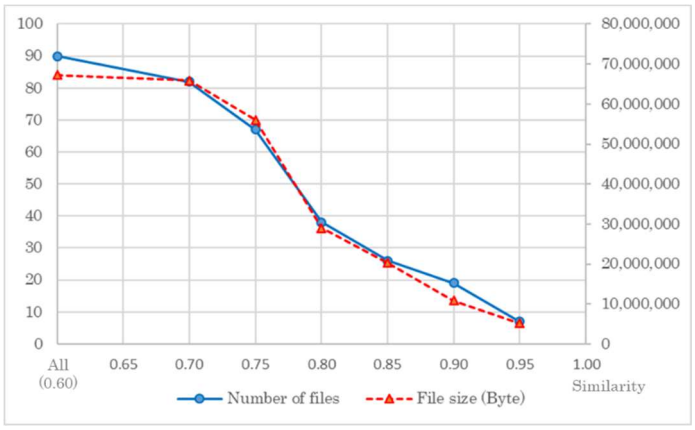


図 5　類似度による画像ファイルのグループ分け

図 6　類似度 0.95 の画像例

表 1　類似ファイル抽出法による重複排除結果

Attribute	Similarity range	Number of files	File size (Byte)	Exec. time (sec.)	Number of blocks	Reduce size (Byte)	Dedup. Rate (%)
All files	0.60 or more	90	67,199,393	467	49,276	4,408,406	6.56
Extracted files	0.70 or more	82	65,877,139	447	48,367	4,353,578	6.61
	0.75 or more	67	56,016,335	380	41,210	4,060,851	7.25
	0.80 or more	38	28,933,533	183	23,085	1,622,290	5.61
	0.85 or more	26	20,285,874	138	15,930	127,669	0.63
	0.90 or more	19	10,878,093	72	8,834	48,208	0.44
	0.95 or more	7	5,136,391	37	3,711	48,208	0.94

一方，類似度が 0.80 以上では，重複排除の実行時間の短縮効果が重複排除によるデータ削減量の減り方より小さく，この範囲では類似ファイル抽出法は効果が出ていないことを示している．

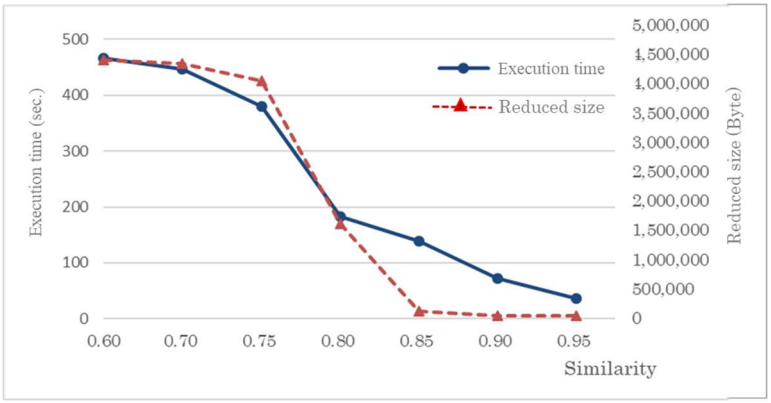


図 7　類似度ごとの重複排除によるデータ削減量と実行時間

6. 終わりに

類似度の高いファイルに対してのみ重複排除を行う類似ファイル抽出法を提案した．この方法は，類似度の低いファイルに対しては重複排除を行わないことで，重複排除の効果の低下を抑えながら重複排除の実行時間を短縮する．実験により，類似度が 0.75 以上のファイルに対して重複排除を行うと，重複排除の実行時間の短縮効果が重複排除によるデータ削減量の減り方を上回り，重複排除をより効率的に行えることが示された．

今後は，画像の図柄だけでなく色の類似性も考慮した類似度や，AI 技術を利用した類似度を用いる方法を考えたい．

参考文献

[1] Q. He, Z. Li, X. Zhang, “Data deduplication techniques,” Future Information technology and Management Engineering 2010 (FITME), vol.1, pp.430-433, Oct. 2010
[2] C. Constantinescu, J. Glider, D. Chambliss, “Mixing Deduplication and Compression on Active Data Sets,” Data Compression Conference 2011 (DCC), pp.393-402, March 2011
[3] A.N. Yasa, P.C. Nagesh, “Space savings and design considerations in variable length deduplication,” ACM SIGOPS Operating Systems Review, Vol.46 Issue 3, pp.57-64, Dec. 2012
[4] H. Yamasaki, I. Koike, T. Kinoshita, “Analysis of double layered deduplication efficiency,” IPSJ SIGMPS Technical Report, 2014-MPS-97 No.9, March 2014 (in Japanese)
[5] M. Noorafiza, M. Hirose, M. Takaya, I. Koike, T. Kinoshita, “Optimum Singularity Size in Data Deduplication Technique,” Proceedings of the 2015 International Conference on Scientific Computing (CSC2015), pp.101-105, July 2015
[6] M.Z. Nurshafiqah, H. Yoshii, F. Enomoto, I. Koike, T. Kinoshita, “Data Deduplication for Audio Data Files,” Proceedings of 32th International Conference on Computers and Their Applications (CATA2017), pp.17-21, April 2017
[7] M.Z. Nurshafiqah, N. Miyamoto, H. Yoshii, R. Kodama, I. Koike, T. Kinoshita, “Data deduplication for Similar Files,” Proceedings of the 2017 International Conference on Scientific Computing (CSC2017), pp.37-42, July 2017