

画像圧縮伸長の高速化方式

4U-11

安部井大、黒須康雄、横山佳弘、西川健一、井口博彰、堀川悦雄

(株)日立製作所 システム開発研究所

ストレージ システム事業部

1. はじめに

画像処理装置の小型、低価格化を計るため、標準パソコンをプラットフォームに画像処理ソフトを搭載した試作機を開発した。このために、画像処理の高速アルゴリズムの開発を行なった。

ここでは、特にMMR方式による画像の圧縮伸長の高速化方式について報告する。

2. 圧縮処理の高速化

圧縮処理では符号化ラインと参照ラインの2本のラインの変化点を検出する必要があるため、1本のラインにつき2回の変化点検出を行なうことになり、処理時間が大きくなる原因になる。

そこで、変化点の検出を1回で済ますために、符号化ラインとして処理している時に検出した1ライン分の変化点の情報を保持しておき、次のラインを符号化する際に参照ラインの変化点の情報として用いる。変化点の情報を保持しておくことで、変化点検出に要する処理時間は実質1/2になる。

また、変化点の検出では、標準のCPUはビット単位の処理に関する命令が乏しいために、画素の変化点を検出する場合も、1ビット単位で画素の色を検出する必要があり、処理に時間がかかる。

そこで、一般的画像は余白の部分が多いことと、専用の変換テーブルを用いることで高速化を計った。CPUが一度に処理できる最大量の32ビット分の画素を読み込み、その中に黒画素が含まれているかをテストする。黒画素が含まれていない場合には、32画素分白画素が連続しているとして処理し、次の32画素を読み込む。32画素の中に黒画素が含まれている場合には、32画素を

8画素ずつ4回に分けて変化点検出テーブルを参照する。変化点検出テーブルを参照することで、8画素の中の白画素と黒画素の連続する状態を知ることが出来る。例えば、図1に示すように、8画素の並びが”01101000”の場合に、テーブルを参照すると、連続した白画素或いは黒画素の個数が1、2、1、1、3であることが分かる。テーブルは8画素の並びに関する情報をバイトデータとして提供するので、この情報をもとに簡単に変化点の情報を作成することが出来る。また、テーブルを用いることで1ビットごとに画素の色を検出する必要がなく、変化点の情報がメモリをリードだけで得られるため、変化点検出処理を高速化することが出来る。

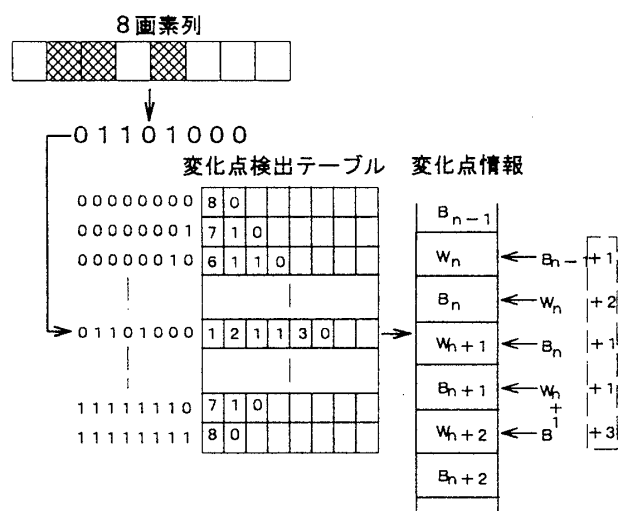


図1 変化点検出テーブル

Method of High Speed Image Compression and Decompression Processing

Hiroshi ABEI, Yasuo KUROSU, Yoshihiro YOKOYAMA, Kenichi NISHIKAWA, Hiroaki IGUCHI

Etsuo HORIKAWA

1.Systems Development Laboratory 2.Data Storage & Retrieval Systems Division HITACHI,Ltd.

3. 伸長処理の高速化

伸長処理においても、圧縮処理同様に参照ラインの変化点の情報が必要である。変化点検出にかかる時間を小さくするために、参照ラインの変化点の情報と符号を解読した結果から直接伸長するラインの画素を生成するのではなく、伸長するラインの変化点情報をCPUが取扱いやすい形式で保持する。画素の生成はその変化点情報をもとに行ない、また、次のラインを伸長する際の参照ラインの変化点情報としても用いる。この方式により、参照ラインの画素から変化点を検出する必要がなくなり、処理時間を削減できる。

符号の解読は、図2に示すような符号解読テーブルを用いて行なう。符号の長さは最大7ビットであるので、圧縮データの未解読の先頭位置から7ビットをアドレスとして、テーブルを参照する。符号解読テーブルからは、アドレスの符号に該当する符号内容と、その符号のビット長を知ることが出来る。このテーブルを用いて解釈した符号の内容と参照ラインの変化点情報から伸長するラインの変化点情報を生成する。また、テーブルから得た符号のビット長は、テーブルのアドレスとして用いた7ビットの中で先頭の何ビットを符号として解読したかを示す。例えば、符号ビット長が6であれば、解読した符号のビット長が6ビットで、7ビット目から次の符号が始まっていることを示している。従って、次の符号を解読する時は、前回アドレスとして用いた先頭から6ビット進んだ位置を先頭とする7ビットのデータで符号解読テーブルを参照する。

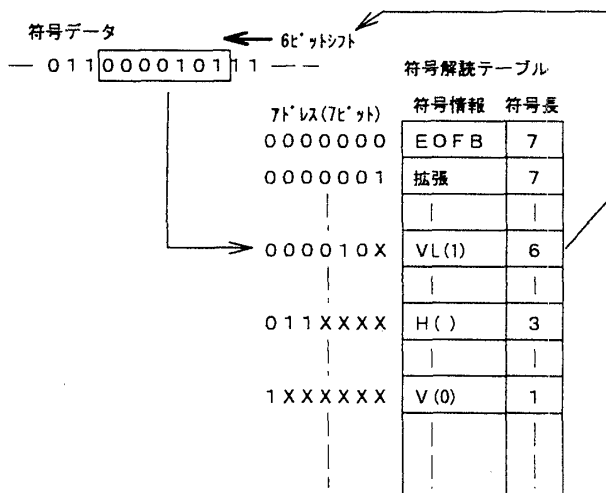


図2 符号解読テーブル

上記処理では、一つの符号を解読する度にテーブルを参照する必要がある。しかし、一般に圧縮率を高くするため、発生頻度が高い符号は短いビット長が割り当てられていることに着目し、符号長の短い符号は一度のテーブル参照によって、複数の符号を解読出来るようにする。例えば、参照ライン上の変化点と同じ位置で変化することを示す符号（記号:V(0)）には“1”が割り当てられている。従って、7ビットをアドレスとしてテーブルを参照する場合には、V(0)の符号が7個連続しているのを一度のテーブル参照で解読することが可能である。

4. 結果

図3に示すように、変化点情報保存方式とテーブルによる変化点検出を採用することで、200dpi、A4サイズのCCITT標準原稿を圧縮処理する場合、処理時間を当初の約1/4にすることが出来た。

また、図4に示すように、変化点情報保存方式と複数符号解読方式を採用することで、伸長処理時間を当初の約1/4にすることができた。

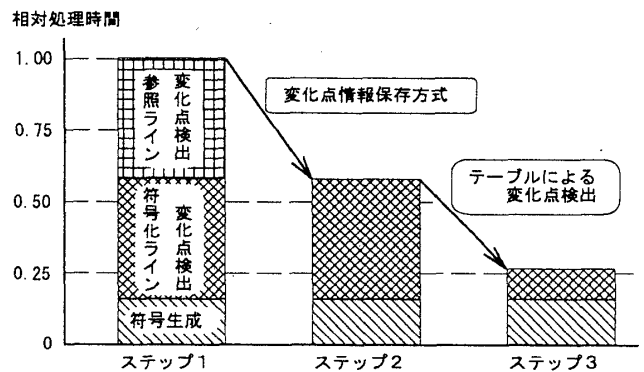


図3 圧縮処理時間

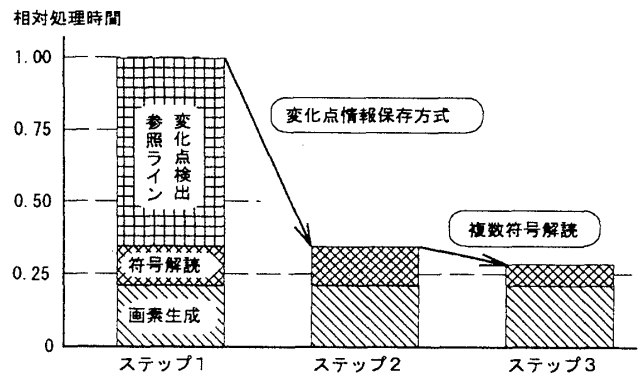


図4 伸長処理時間

5. 参考文献

1) 釜江尚彦、吹抜敬彦：デジタル画像通信，産業図書（1985）