

近年、パーソナル・コンピュータのハードウェア機能はいちぢるしく発展し、グラフィックをふんだんに使用したゲーム等が多数開発されている。これらパーソナル・コンピュータで、大量のグラフィックデータを扱う場合に、最も問題となるのが、そのデータサイズである。

ここで、画像データの自己相関性に着目した、ゲーム用グラフィックデータ圧縮方法を開発したので報告する。

データ圧縮の概要

画像データの圧縮方法に、最も早くから取り組んできたのは、TV、ファクシミリ等の2次元・画像データの通信分野である。

ファクシミリ等では、通信の信頼性・経済性を高めるため、通信に要する時間を短縮する必要があった。このために、画像データサイズの圧縮が必要になってきた。

画像圧縮とは、原画像の情報量を損なわずに、あるいは人間の目からみて、殆ど損なわないように、原画像データのデータ量を何らかの方法によって減らす事である。一般的に、データの圧縮を行うには、元データを単純な符号(コード)化するのが有効である。ファクシミリでは、画面の白黒の長さを、例えば次のような単純なコード文字列

・ ・ 白 1 0 ・ 黒 1 ・ 白 1 0 2 ・ 黒 4 ・ ・

というように符号化すれば、元のドット数に比べ、劇的にデータ量を減らす事が出来る。

この符号化においては、原画像に対する忠実度によって、基本的に2つの方法がある。すなわち、

* 超越科学研究所・ワークスキッチュ
マッドサイエンス学会正会員
Laboratory of Hyper-Science
Yamanashi JAPAN

である。

可逆符号化とは、圧縮した画像データを復元すると、まったく元の画像データが再現するものであり、ファクシミリにおける画像圧縮はこれを採用している。

一方、不可逆符号化とは、圧縮したデータからは、元の画像が完全には復元できない。しかし、細部の情報が欠落しても、人間の視覚にとっては、全体の印象があまり変わらないという特性を利用して、情報の削除・圧縮を行うものであり、TV、スチルカメラ等の自然画像の圧縮によく用いられている。

自己相関法によるデータ圧縮

パソコンゲームにおける画像データは、グラフィックツールを用いて作成した、コミック・アニメライクな画面である場合が多い。

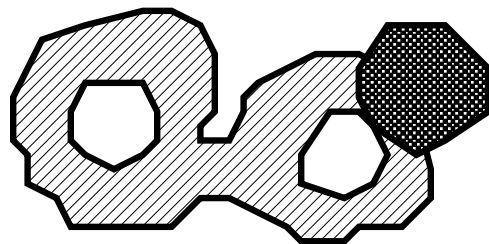


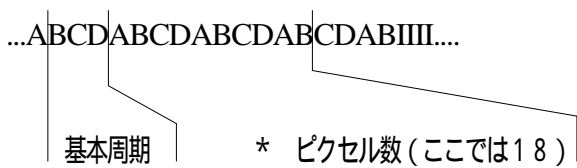
図1 画像データの例

このような画像では、単色に塗られている領域が比較的大きく、このために、画像をピクセル(画面の1ドット。画素ともいう。)単位で観察すると、上下左右のデータが同じカラーであるという場合がほとんどである。

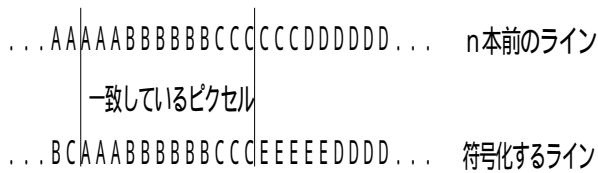
また、中間色を現すのによく使われるタイリング(ピクセル単位の一松模様)も、その周期が縦横16ピクセル以内である場合がほとんどである。

このため、ピクセルの横方向のデータのつながり、縦方向のデータのつながりを利用すれば、効率の良い画像データ圧縮が実現できる。

図2に、具体的なデータ圧縮の方法を示す。ここでは、ピクセルの色をABC・・・で簡略表示している。



(1) 横方向の符号化



(2) 縦方向の符号化

図2 縦・横方向の符号化

この例より、自己の画像データと相関を取る事によって、符号化が可能な事がわかる。すなわち、図2(1)の場合においては、次のような符号化コード、

- 始点X, Y : (X座標)、(Y座標)
- 符号種類 : 横方向相関
- 相関ピクセル長さ : 4ピクセル
- 画像ピクセル長さ : 18ピクセル
- ピクセル色パタン : ABCD

を用いる事で、効率のよいデータ圧縮を行うことができる。これは、縦方向の符号化も同様である。なお、符号化データは、本研究においては、4ビットのニブルサイズを基本にしている。実際のデータ設計(ビットフィールド設計)は、煩雑になるため省略する。

この他、画像データによっては、圧縮をかけない方が、データサイズの観点から有効である場合も考えられる。このため、本研究により開発した画像圧縮プログラムは、2ステップの画像圧縮(コンパイル)を行うようにしている。(図3)

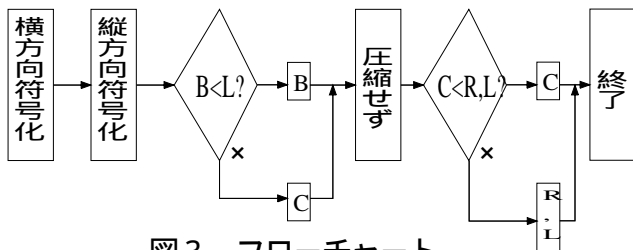


図3 フローチャート

また、縦方向相関を効率良く行うために、画像の水平線(640ピクセル相当)を記憶するメモリーを16ライン分、作業用を4ライン分確保している。

結果と考察

本研究に基づくプログラムの開発は、NEC PC-9801UX上にて、TURBO-C Ver.2.0により行った。画像データの作成・編集には、(株)ツアイトのZ's STAFF Kid98を用い、画像取り込みには16ハーフトーン・スキャナ PC-in503Hを利用した。

論文と共に発表されるプログラムフロッピーには、画像再生用プログラムLOADER2.EXEと、各種画像データ、バッチファイルのみ納められている。

画像圧縮の効率を、実際の画像データについて実験した結果を表1に示す。

表1 圧縮データ実験

ファイル名	B_KO.*1 (ベタファイル)
圧縮前のサイズ	計128Kbyte
圧縮後のサイズ	37046byte
圧縮率	28.9%

このように、本プログラムにおいては、非常に高い効率で、データを圧縮する事が可能である。また、各符号化された画像データは、始点の座標X, Yを持っているため、コンパイルの仕方によって、どのような位置からでも再生させる事が可能である。

まとめ

ゲーム用にターゲットを置いた、画像圧縮プログラムを自己相関法に基づき開発した。これにより、高効率・高機能な画像圧縮が実現できた。本研究が、高度なグラフィックを利用したゲーム開発にささやかなる貢献ができれば、研究者として、これに勝る喜びはない。