

7N-6

音声エディタ (2)
— メモリー及びファイル管理方式 —

○大槻朋子 黒田明裕

日本アイ・ビー・エム株式会社 サイエンス・インスティテュート

1. はじめに

音声データは、テキストデータなどと比べて、データ量が膨大である。この問題は従来から音声データの圧縮という観点から様々な提案がなされてきた。しかし、一方では圧縮による音質劣化のため、PCMなどのデータ量が多いが良質の音声データの用途も無視できない。このような良質の音声データを編集する場合の問題点の一つはエディタの実行時間である¹⁾。本報告では、この問題を解決するための音声エディタのメモリー管理/ファイル管理法をMS-DOS上での実現例と共に述べる。

2. データ管理

音声エディタの編集は最終的にはPCMなどの音声データに反映させる必要がある。しかし、これらのデータは10Kbytes/sec程度とデータ量が大きいため、直接編集を行うことは非効率的である。そこで、実際のデータ移動を必要としない編集方式を可能にする論理データ構造の表現方法を用いた。編集に伴う変更は音声データの論理データ構造を記述する構造記述子(図1 Main Memory 上部)に反映される。さらに、MS-DOS上のファイルは、ひとまとまりのデータ(1クラスター = 2Kbytes)のつながりを示すポインタのテーブル(FAT)で管理されているため、ファイルのsaveの際には構造記述子に従ってFATを書きかえるだけでよい。

ところで、実際の編集の際にはワーキングデータなどの視覚化データ(データ量はPCMの1/100程度)が必要であり¹⁾、これに対しても編集を行う必要がある。そこで、構造記述子は視覚化データの構造を記述するものとし、最終的なsaveの際に、視覚化データを音声データへのマッピングに利用しつつFATの書きかえを行う方式を用いた。以下、これらのデータ管理方式の詳細をメモリー及びファイルについて述べる。

2-1. メモリー管理

メモリー管理は図1に示したように音声データへのマッピングを示す視覚化データとその論理データ構造を示す構造記述子からなる。ところで、音声データは本来単なるバイト・パットの連鎖であり、表現すべき構造は時間軸上での前後関係のみである。このような観点から音声データの論理構造を表現する方式として図1に示したものを用いた。1つの節は2つのポインタ

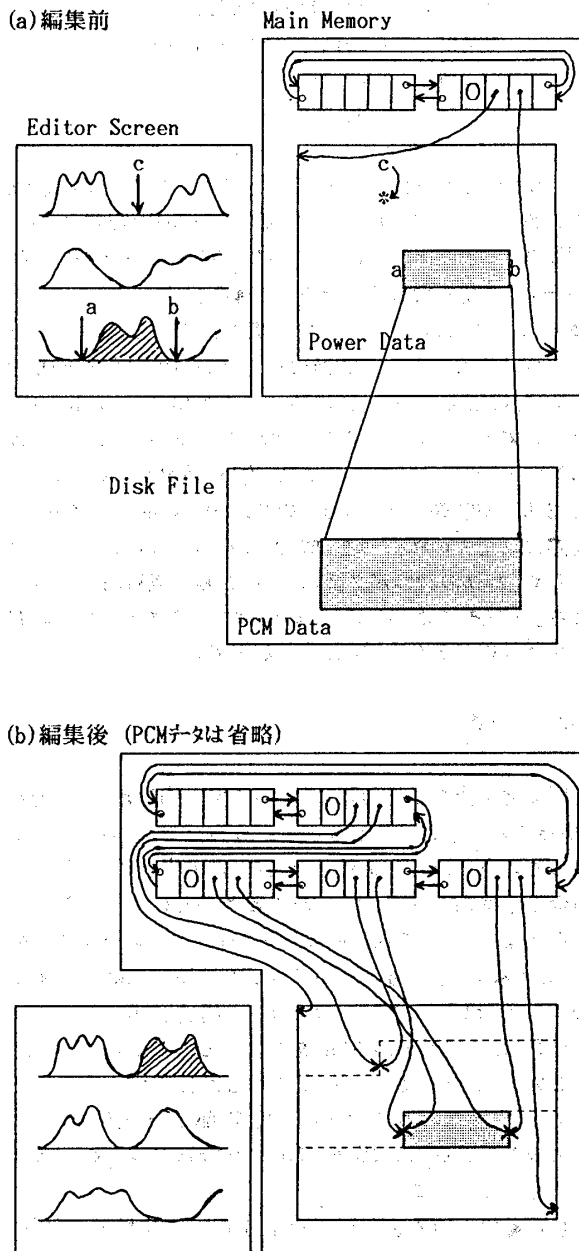


図1. moveコマンド実行に伴うメモリーおよびエディタ画面の変化

を持ち、それらはデータの始まりと終りを指す。これにより、基本となる情報の単位は編集中にエーサの指定に応じて定義されることになり、不必要なリンクを持つ必要がなくなる。構造記述子、第2項は音声ファイルの区別を示すものであり、別のファイルからの引用(get)を可能にしている。

moveコマンドの実行を例に以下の場合を仮定してメモリ管理の詳細を述べる。

- 音声データ : 8KHzサンプリングの12bit PCM
- 視覚化データ : 8bit/ピクセル(177x4:128points)
- コンピュータ : PC-AT
- OS : MS-DOS

編集が全く行なわれていないファイルにおいて図1-(a)の矢印a,bで示した区間の斜線部に相当する音声データを矢印cの位置に移動する。編集前の状態は図1-(a)で記述される。編集後は、図1-(b)に示したように実際のデータおよびPCMデータの移動を行なうことなく、構造記述子のリンクの書き換え、および節の付加のみで編集を行なえる。従って、その実行時間はほとんど問題にならない。例えば、1分のPCMデータの最後部のデータを先頭に持ってくる場合、1wordのメモリからレジスタへのmove命令の所要時間を1μsecとすると1分のPCMデータ720Kbytesのmoveでは、約1秒必要となる。もちろんMS-DOSが管理できるメモリエリアは640Kbytesであるから、実際には全てのPCMデータをメモリ上に読みこむことはできないため、ファイルのアクセスが必要になり、さらに実行時間が長くなる。本方式では、3つのリストの付加ですむため、所要時間は10⁻³秒程度となり、ほとんど無視できる。

2-2. ファイル管理

編集終了後のsaveの際には、不必要に物理データと論理データ構造のギャップを大きくしないという方針で、構造記述子に従って音声データ・ファイルを書き替える。この場合にも音声データの全てを移動させず、主にFATの書き換えにより行なう。

例えば図1のようなmoveが行なわれたデータをsaveする場合、まず視覚化データからのマッピングにより、a,b,cに対応する音声データの場所を求める。すなわち、構造記述子が指す視覚化データの要素番号からFATを順にたどることにより対応するクラスターを求める。いま、この対応が、図2のようであったとする。すなわち、クラスター4-6の斜線部のデータをクラスター2の*印の部分にmoveした場合、音声データ・ファイルのsaveの手順は以下ようになる。

- (1) 編集の境界を含むクラスター2,4,6のデータを境界で分割し、一方を新しいクラスター9,10,11にコピーする。
- (2) 図4の()で囲んだFATの書き換えにより、データの並び換えを行なう。

1分の音声データに1カ所のmoveの編集が行なわれている場合、実際に音声データ全てをsaveする場合にはデータのアクセス時間を50Kbytes/secとすると約15秒必要である。これに対し本方式では6カ所のFATの書き換えと3個のクラスター(PC-AT 20Mbytes Hard Diskの場合、FATが示す1クラスターの容量は2K

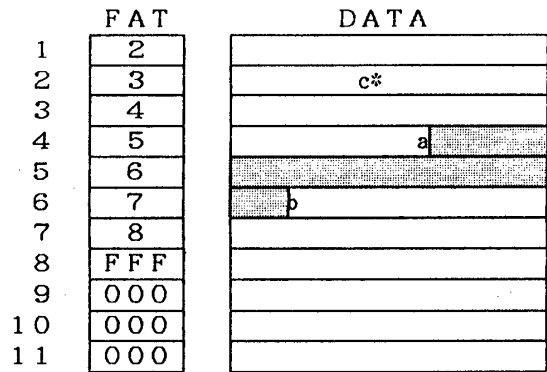
bytes)のコピー計6Kbytesの書きこみで行なうことができ1/120の実行時間(約0.1秒)に短縮できる。

FATの書き換えを中心にファイルのsaveを行なうため、図2-(b)に斜線で示した不必要なデータ領域が生じる。例えば、1分の音声データに10カ所の編集を行なった場合、ファイルの量は数%増加する。ただし、これらの部分は別の機能により必要に応じて取り除くことができる。

3. まとめ

音声データにおけるメモリ/ファイル管理法を提案し、これをPC-AT上を実現することにより、十分実用にあう速度で編集コマンドが実行できることを確認した。

(a) 変更前



(b) 変更後

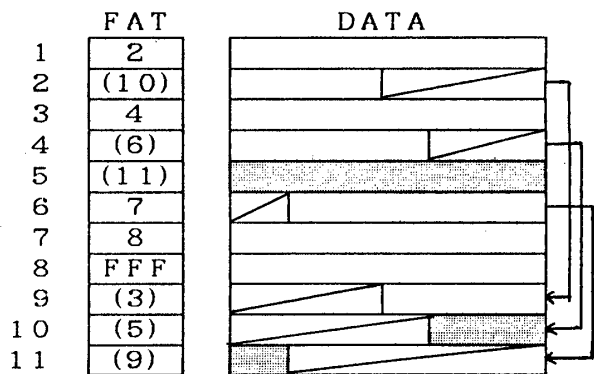


図2. 編集後のFATおよびデータの変更

参考文献

- (1) 黒田,大槻, "音声データ(1) -システム及び機能の概要-", 第33回情報処理学会全国大会.