コンピュータビジョン　2－3
（1979. 9. 20）

動画像解析のためのデータ構造とソフトウェア・システム

喜多伸之　谷内田正彦　福井達郎　辻三郎
（大阪大学基礎工学部）

1. まえがき

画像処理の分野では、従来は静止画像を対象とする研究がほとんどであったが、最近では時間的に連続する複数画面（動画像）を対象とする研究が増えている。特に、医学、生物学などの分野では動物体を対象とし、それらの動きや形の変化を解析することが多く、これらを計算機を使って効率良く処理することが望まれている。しかし、動画像解析において扱うデータ量は、静止画像を解析する場合に比べはるかに多いので、

1). データの格納に膨大な記憶容量を要する
2). 処理に時間がかかる

という点がしばしば問題となる。

ところが、動画像の解析においては、一般にそのすべての情報を必要とすることはなく、一部の情報のみを必要とする場合がほとんどである。たとえば、ある対象物にのみ着目して解析を行なう場合には、その対象物を含む領域の情報だけが必要であり、また、フレーム間でほとんど状態の変化がない場合は、フレームを適当に間引いて入力すれば十分である。これらのことより、(A) 格納すべき領域の選択、及び、(B) 格納すべきフレームの選択、を行なうことが動画像データを圧縮した形で格納するためにたいへん重要になってくる。

ここでは、これらの点を考慮して開発したデータ構造と、そのデータ構造を中心に、動画像を効率良くアクセス・転送・処理・ディスプレイできるソフトウェア・システムについて述べる。

2. データ構造

動画像のもつデータ量は膨大であり、そのすべてのフレームにわたって画面全体の情報を格納するには多くの記憶容量が必要となる。そこで、先に述べたように、次の2点を考慮して、画像データを圧縮した形でディスクに格納するためのデータ構造について述べる。

1). フレームの間引きを行なう
2). 各画面内で必要な領域のデータのみを格納する

まず、#2の要求をみたすために、ここでは画面全体を図1に示すように、正方形のセグメントに分割し、画像をセグメントの集合として扱っていく。これにより、重要なデータを含む領域のデータのみを抽出し、圧縮した形で格納することが可能となる。さらに、#1の要求、フレームの間引きを可能にするために、データ構造は

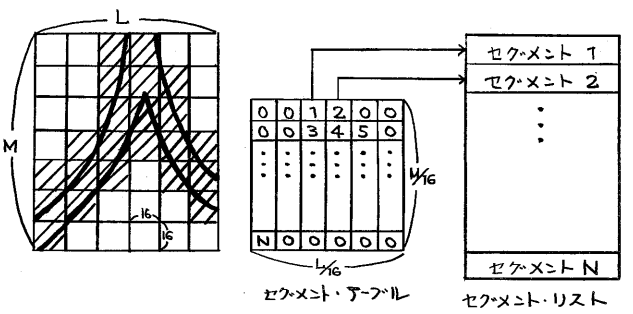

図1. 画面の分割

各フレームで独立させた。その結果、画像ファイルは図2に示すように、動画像全体の情報をもつメイン・ヘッダと、各フレームの情報をもつセグメント・テーブル、セグメント・リスト及び、縮小画像で構成される。以下、これらについてくわしく述べる。ここでは、セグメントの大きさを16×16画素にした。これは、1画素8ビットでセグメントを格納するのに256バイト要することになり、これがちょうどディスクの1レコードに相当するためである。セグメント・リストは、このような1つ256バイトのセグメントの集合である。セグメント・テーブルは、セグメント・リストにアクセスするさいに必要となる各セグメントへのポインターをもつ。セグメント・テーブルの構成は、図1のように、各セグメントに対し1つのセル(1ワード)をもち、各セルには、そのセルに対応するセグメントが格納されているときには、セグメント・リスト内でのアドレスを記録しており、格納されていないときは、Øを記録している。よって、セグメント・テーブルの大きさは、画面の大きさを$m \times l$としたとき、$m/16 \times l/16$となる。縮小画像とは、画面全体を空間平均して得られた画像で、ユーザーが各フレームにおいて必要と指定した場合のみ、画像ファイルに2次元構造で格納される。これは、とくに高分解能の画像を扱うことを考慮したもので、画面全体の情報を格納する必要が生じたときに有効である。これら3つにより、各フレームは構成されるが当然フレームによって、セグメント・リストの大きさが違うため、各フレームの大きさは異なってくる。そこで、画像ファイル内での各フレームの先頭アドレスを記憶しておく必要が生じる。それを行なうのが、メイン・ヘッダである。メイン・ヘッダは、画像ファイルの先頭にあり、画像ファイル内の総フレーム数、画面の大きさなど、すべてのフレームに共通な情報を格納しており、さらに、各フレームの先頭アドレスと、縮小画像の有無を示すテーブルをもつ。このようにして、1つの画像ファイルが構成される。

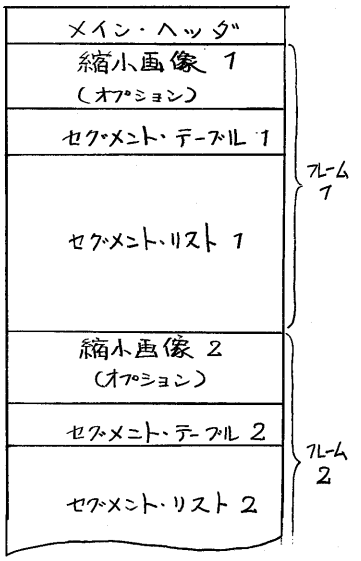

図2. 画像ファイル

このデータ構造のもつ特徴は、画面をセグメントに分割して扱っていることであり、これにより、有効な画像データの圧縮が可能となっている。さらに、こうした特殊な構造の画像データへのアクセスが容易に行なえるように工夫されていることである。

3. 入力システム

当システムの使用する計算機は、ミニコンYHP2108Aであり、動画像を扱い易いように、外部メモリとして、256Kバイトの画像メモリを装備している。また、画像の入力装置は、35mm映画フィルムより、1画素8ビットで最大1024×1024の大きさの画像を入力できる。

ここで述べる入力システムは、動画像を第1フレームについては領域の選択を行ない、第2フレーム以降については1フレームの間引きを行なったあと、さらに

領域の選択を行ないながら、ディスク上に画像ファイルとして入力する機能をもつ。

第1フレームと、第2フレーム以降では入力手順が異なるので、ここでは、まず第1フレームの入力手順について述べる。

1). 画像ファイル名、画面の大きさなどのパラメータの入力

これにより、システムは、ディスク上に新しく画像ファイルをクリエイトし、さらに、メイン・ヘッダに必要な情報をかきこむ。

2). 画面全体のデータの入力とモニタ

入力された画面全体のデータは、一度ディスク上のテンポラリー・ファイルに格納される。さらに、入力状態はカラーディスプレイに表示される縮小画像(画像メモリ上に入力と同時につくられる)によりモニタされる。

3). 縮小画像の解析による領域の選択

領域の選択は、縮小画像を解析することにより行なうが、動画像によりその対象物はさまざまな特徴をもつため、その方法はユーザーにより工夫される必要があるが、現在、システムとしては縮小画像を空間微分してその微分値の高いところを選択するように設計してある。

4). インタラクティブな修正による最終的な領域の決定

3)において選択された領域は、縮小画像とともにカラーディスプレイ上に表示され、満足のいくものであればそのままで、そうでなければインタラクティブに修正することにより最終的な格納領域の決定を行なう。

5). 画像ファイルへの格納

4)において決定された領域の画像データをセグメント・リストの形に変換し、セグメント・テーブルと共に画像ファイルに格納する。このとき、ユーザーにより必要と指定されれば、縮小画像も格納する。

以上の手順で第1フレームの入力が行なわれる。

次に、第2フレーム以降の入力手順について述べる。

1). 画面全体の入力とモニタ

第1フレームと同様、画面全体のデータを一度テンポラリー・ファイルに格納する。

2). フレームの選択

フレームの選択も縮小画像を用いて行なう。入力中のフレームを間引くかどうかは、画像メモリに格納されている前画面の縮小画像との各画素でのグレイレベルの差分をとり、その差分があるいき値をこえる画素数が多いかどうかにより決定し、多い場合は3)にすすみ、少ない場合は次のフレームにすすんで1)にもどる。こうして、前画面との間で状態の変化が少ないフレームは間引かれる。

3). 領域の選択

前画面と入力中の画面の縮小画像とで、セグメントごとに、そのセグメント内の各画素でのグレイレベルの差分をとり、その差分がユーザーの指定したいき値をこえる画素数が多いセグメントを格納領域と決定する。こうして、前画面との間で状態の変化が大きい領域が選択される。

4). インタラクティブな領域の修正

第1フレームと同様、カラーディスプレイに表示された縮小画像と格納領域

をみて、インタラクティブに修正を加え、最終的に格納領域を決定する。

5). 画像ファイルへの格納

4)での決定に従い、第1フレームのとき同様、データ構造を変換してディスク上の画像ファイルに格納する。必要なら縮小画像も格納する。

以上の手順をくり返すことにより、領域の選択、フレームの間引きという2点を満足しながら、動画像のディスクへの効率良い格納が実現される。

4. 出力システム

ここでは、ディスクに格納された画像ファイルの任意のフレームの任意の領域を表示できる出力システムについて述べる。

システムの画像表示装置としては、カラーディスプレイおよび、プリンタ・プロッタがあるが、これらはいずれも、画像メモリに2次元的な配列で格納された画像を表示するよう設計されているため、圧縮された形でディスクに格納された画像を表示するためには、データ構造を変換して画像メモリ上に格納する必要がある。さらに、表示装置はいずれも最大256×256の画像を1画素4ビットでしか表示できないため、256×256より大きい画像については、画面全体を表示することができないので、局部的に表示する必要があり、さらには、1画素8ビットの画像を1画素4ビットの画像に変換しなければならない。

当システムはこれらの機能を備えており、とくに1画素8ビットの画像を1画素4ビットの画像に変換するさいには、その画像のグレイレベルのヒストグラムをみて、任意のグレイの範囲で16レベルに変換できる。この方法についてくわしく述べる。まず、蓄積型ディスプレイに表示された画面全体のグレイレベルのヒストグラムが、図3のように、0～255レベルまでなめらかに分布していたとする。このとき、画面全体を表示するならば、0～255レベルを16等分して16レベルに変換すれば、もっとも有効な表示が行なえるであろう。ところが、この画面のある領域だけを表示したい場合、その領域のグレイレベルのヒストグラムが、図4のように、あるグレイの集団にかたよって分布したとする。このようなとき、前と同様の変換をしたならば、その表示画像はほとんど情報をもたないものとなる。そこで、ここでは、変換後の0レベルの上限と、15レベルの下限を指定することにより、図4のように局部的な16レベルへの変換を実現できるようにした。これにより、画面の局部の表示が、効果的に行なえるようになった。

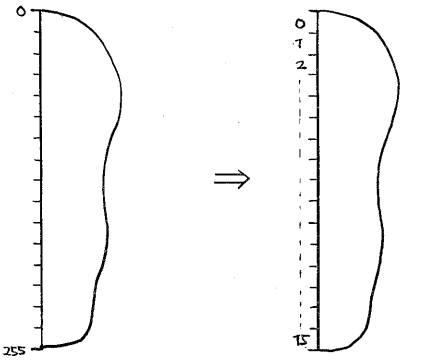

図3. グレイの変換(a)

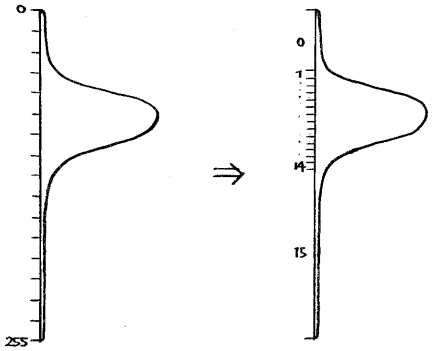

図4. グレイの変換(b)

5. 処理システム

ディスクに格納された動画像を効率良く解析するためには、ユーザー・プログラムにより、画像データが容易にアクセスされる必要がある。ここで述べるシステムはユーザーに、容易に効率良く画像データをアクセスする方法を提供する。

このシステムの特徴は、ユーザーが動画像全体のデータが常に計算機の内部メモリに格納されているかのように考えてプログラミングできることである。

システムは図5に示されるように、内部メモリ、画像メモリ、ディスクのための3つのデータ構造をもち、これらのデータ構造間のデータの流れを管理し、ディスク、画像メモリを使って内部メモリに必要なデータを供給する。

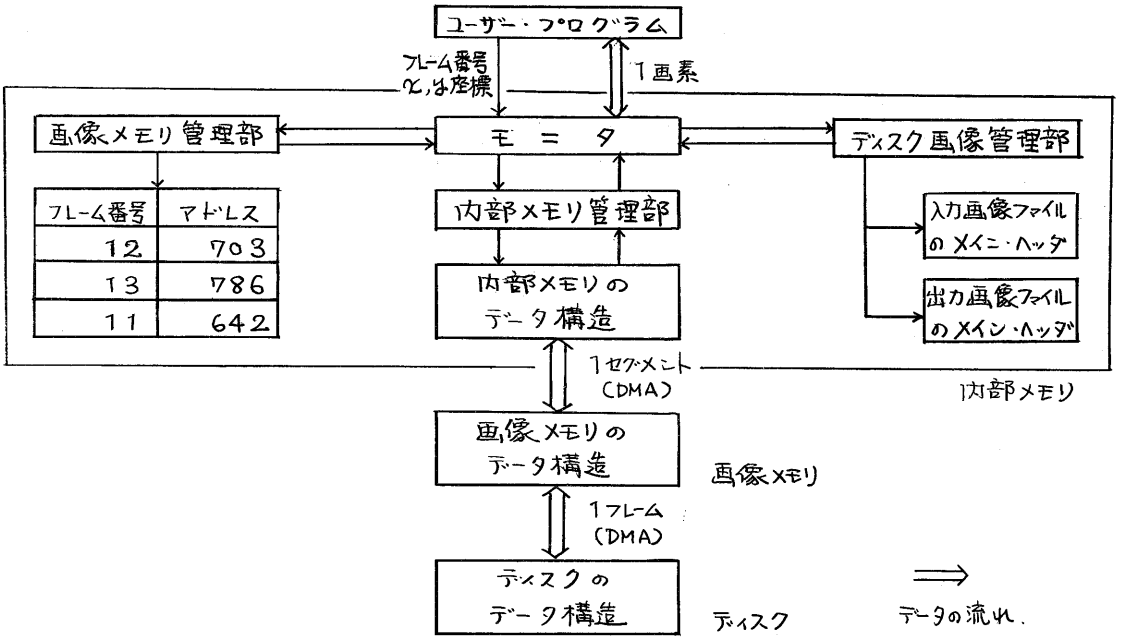

図5. システムの構造

3つのデータ構造のうち内部メモリは、解析中の画像ファイルの数個のセグメントのデータを格納しユーザー・プログラムにデータを提供する。画像メモリは内部メモリとディスク間の高速バッファメモリとして働き、数フレームの画像データを格納する。ディスクは、内部メモリにセグメントを供給するための仮想メモリとしての役割りを果たす。画像データは、これら3つのデータ構造の間を、システムの管理にしたがって流れるのであるが、内部メモリと画像メモリとの間ではセグメント単位で、画像メモリとディスクとの間ではフレーム単位で、それぞれDMA転送される。

内部メモリのデータ構造は、4つのプッシュダウン・スタックと、12のセグメント・リストより構成される。セグメント・リストは、ユーザー・プログラムがアクセスしているセグメントのデータを格納する。動画像の解析においては、時間微分などのように、同時に複数のフレームの画像データをアクセスするため、セグメント・リストは、複数のフレームのセグメントのデータを格納することに

なる。4つのプッシュダウン・スタックのうち3つは、ユーザー・プログラムにデータを供給するための入力フレームに、1つはユーザー・プログラムにより処理結果として新しくつくられる出力フレームにわりつけられ、セグメント・リストに格納されているセグメントのフレーム内でのアドレスと、セグメント・リストへのポインターをもつ。

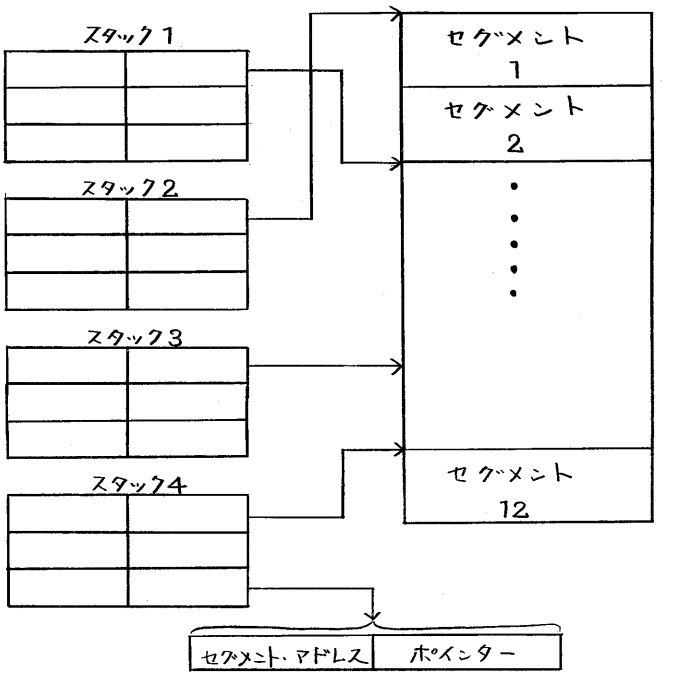

図6. 内部メモリのデータ構造

仮想メモリとしてのディスク上での画像ファイルのデータ構造は前述のとおりであり、すべての画像ファイルはそのファイル名で、ユーザー・プログラムによりアクセスされる。ディスク画像管理部(図5)は、ディスク上の画像ファイルを管理し、ユーザー・プログラムにより、入力画像ファイル名が与えられれば、その画像ファイルをオープンしメイン・ヘッダをディスクより内部メモリによみこむ。また、出力画像ファイル名が与えられれば、新しくディスク上にその画像ファイルをクリエイトし、内部メモリにメイン・ヘッダとして必要な情報をかきこむ。

内部メモリとディスク間の高速バッファメモリとしての画像メモリのデータ構造は、画像ファイルを構成するセグメント・テーブルとセグメント・リストよりなる。前にも記したように動画像解析においては、同時に複数枚のフレームをアクセスするが、画像メモリの大きさ(256Kバイト)により、同時に画像メモリに格納されるフレーム数は制限される。画像メモリ管理部(図5)は、画像メモリに格納されているフレームの、フレーム番号と、画像メモリ内のそのフレームへのポインターをもつスタックにより構成される。

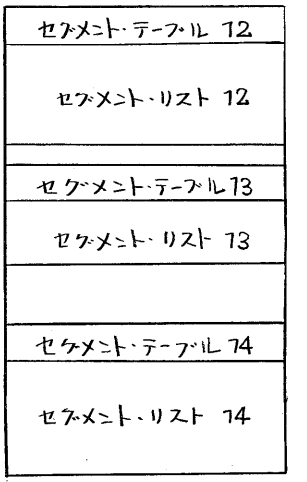

図7. 画像メモリのデータ構造

システムのモニタは、これら3つの管理部により構成されるわけだが、その動きを、ユーザー・プログラムが画像データをよみだすときのデータ構造間のデータの流れを例に説明する。

まず最初に、ユーザー・プログラムにより、解析すべき動画像ファイル名がシステムに与えられ、ディスク画像管理部がその画像ファイルのメイン・ヘッダを内部メモリによみこむ。次に、よみだしたいデータのフレーム番号とx, y座標

がモニタに与えられたならば、モニタはx, y座標より、そのデータを含むセグメントのアドレスと、そのデータのセグメント内でのアドレスを計算する。

次に、そのデータをもつフレームにわりつけられたプッシュダウン・スタックの先頭をしらべる。もし、そのセグメント・アドレスが先に計算したセグメント・アドレスに一致しているなら、セグメント・リストに格納された対応するセグメントのデータより、求めるデータをユーザー・プログラムに供給する。

ここで、もし、プッシュダウン・スタックの先頭のセグメント・アドレスが、目的のセグメント・アドレスと異なっていたならば、プッシュダウン・スタックを底までしらべていき、もし一致するセグメント・アドレスがあったならば、それを先頭と交換してから先と同様にして、ユーザー・プログラムにデータを供給する。

一方、プッシュダウン・スタックの底までしらべても、一致するセグメント・アドレスがなかった場合、つまり、内部メモリに求めるセグメントのデータが格納されていない場合には、目的のセグメントのデータを画像メモリより内部メモリのセグメント・リストにDMA転送する。このとき、プッシュダウン・スタックの内容は1つダウンされ、いま転送されたセグメントはスタックの先頭にくるため、ここで、ユーザー・プログラムにデータの供給が可能となる。

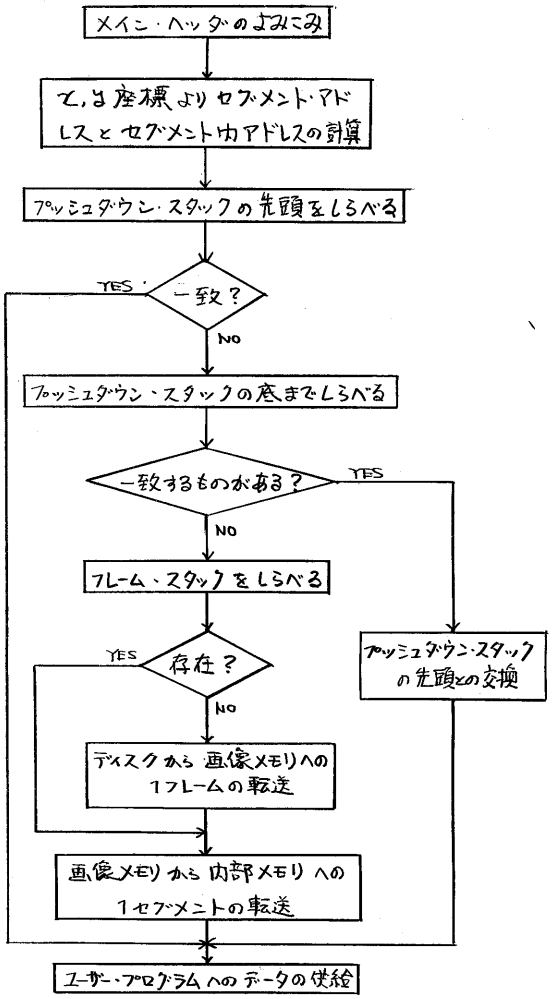

図8. システムの働き

ところで、画像メモリより内部メモリにセグメントを供給する場合であるが、先にも述べたように、画像メモリに目的のフレームが常に格納されているとは限らない。よって、画像メモリ管理部のフレーム・スタックにより、格納されているフレームをしらべ、目的のフレームがなかった場合には、ファースト・イン/ファースト・アウトに従って、ディスクより画像メモリに目的のフレームがDMA転送される。このようにして、モニタの働きにより、ユーザー・プログラムにデータが供給されるわけであるが、わかりやすいように、上で述べた例をフローチャートにして図8.に示しておく。

この例で示されるように、画像データは3つのデータ構造間をシステムのモニタの管理に従って流れるわけであるが、最終的には、ユーザー・プログラムによ

リアクセスされたデータをもつセグメントは常に、内部メモリ内のプッシュダウン・スタックの先頭に格納されることになり、さらには、その周辺のセグメントのデータも、内部メモリのセグメント・リストに格納されている。このことが、システムのオーバー・ヘッド・タイムをできるだけ短くするために工夫された点であり、有効に働く。なぜなら、画像データの連続したアクセスは、先にアクセスした点の近傍の点で行なわれるのが普通であり、まったく入けはなれた位置の点がアクセスされる場合はほとんどないためである。

このように、ユーザーはシステムのモニタに、フレーム番号と、x座標を与えるだけで、つまり、内部メモリに画像ファイルのすべての画像データが格納されていると考えて、プログラムをかくことができるのである。

6. むすび

動画像を効率良く格納するためのデータ構造と、そのデータ構造をもった動画像の入力、出力、及び処理を行なうためのソフトウェア・システムについて述べた。

ここで述べたデータ構造は、各画面をセグメントに分割して扱うという特殊な形をとっているが、これにより、画面の重要な領域のデータだけを抽出して画像ファイルに格納することが可能となり、かなりの記憶容量の節約が実現された。現在までに入力された例によると、従来の方法で格納した場合の、1/2から1/4程度に圧縮された。処理システムについては、特殊なデータ構造をもつ画像データを扱うにもかかわらず、オーバー・ヘッドを最少におさえ、かなり効率良く処理できるシステムが実現された。処理システムの大きさは、3つの管理部、及び内部メモリのデータ構造で、約9Kバイトである。また、このシステムで動画像を解析するために要する時間は、同じ大きさの画面で同じ処理を行なっても、その画面の格納領域の大小により異なるが、たとえば、256×256の大きさで格納領域が1/4ほどの画面に対し、単純な時間微分を行なった場合、その処理時間は約45秒であった。

今後は、システムに機能を追加し、より使い易く効率の良いシステムに改良していきたいと考えている。特に、入力システムに関して、現在はその入力装置として、高分解能画像入力装置だけをもっているが、ハードウェア的には、TVカメラ、VTRが計算機とつながっているので、現在の入力システムに、これらTVカメラ、VTRからも動画像を入力できる機能を追加したいと考えている。