(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3727768号
(P3727768)
(45)発行日　平成17年12月14日(2005.12.14)　(24)登録日　平成17年10月7日(2005.10.7)

(51)Int.Cl.[7]　F I
G06T 15/70　G06T 15/70　A

請求項の数 2　(全 11 頁)

(21)出願番号　特願平9-296451
(22)出願日　平成9年10月15日(1997.10.15)
(65)公開番号　特開平11-120379
(43)公開日　平成11年4月30日(1999.4.30)
審査請求日　平成14年7月22日(2002.7.22)

(73)特許権者　591095856
株式会社ハドソン
北海道札幌市豊平区平岸三条五丁目４番２２号
(74)代理人　100093517
弁理士　豊田　正雄
(72)発明者　三浦　高志
北海道札幌市豊平区平岸３条５丁目１番１８号株式会社ハドソン内
(72)発明者　熊岡　忍
北海道札幌市豊平区平岸３条５丁目１番１８号株式会社ハドソン内

審査官　村松　貴士

最終頁に続く

(54)【発明の名称】コンピュータ画像の表示方法及び表示装置

(57)【特許請求の範囲】

【請求項１】

フレーム毎に作成された３次元データに基づいて作成した画像を連続的に表示して動画表示を行うコンピュータ画像の表示方法において、

２つの特定フレーム間に表示されるフレームの３次元データが前記２つの特定フレームデータから補間近似により作成された近似データが元の３次元フレームデータとの差が一定の値以下の場合に、前記２つの特定フレーム間の元の３次元データを前記２つの特定フレームデータをキーフレームデータとして補間する係数ベクトルからなるフレームデータで置き換えて動画表示を行うデータを作成し、

画像再生時には、前記キーフレーム間のフレームデータを前記係数ベクトルを用いて補間して作成した画像を前記キーフレームのデータより作成される画像の間で順次表示して動画表示を行う

ことを特徴とするコンピュータ画像表示方法。

【請求項２】

フレーム毎に作成された３次元データに基づいて作成した画像を連続的に表示して動画表示を行うコンピュータ画像表示装置において、

２つの特定フレーム間に表示されるフレームの３次元データが前記２つの特定フレームデータから補間近似により作成された近似データが元の３次元フレームデータとの差が一定の値以下の場合に、前記２つの特定フレーム間の元の３次元データを前記２つの特定フレームデータをキーフレームデータとして補間する係数ベクトルからなるフレームデータ

で置き換えて動画表示を行うデータを用い、
画像再生時には、前記キーフレーム間のフレームデータを前記係数ベクトルを用いて補間して作成した画像を前記キーフレームのデータより作成される画像の間で順次表示して動画表示を行う手段を備えた
ことを特徴とするコンピュータ画像表示装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、フレーム毎に作成された画像データを連続的に表示して動画表示を行うコンピュータ画像表示装置に用いる画像データの削減方法に関する。詳しくは、コンピュータアニメーションについて元のアニメーションの質を落とさずにデータを削減する方法に関する。

【０００２】

【従来の技術】

コンピュータで画像データ、特にアニメーションを表示する場合、キャラクタの動きを滑らかに表現するためには、フレーム数を増やす必要がある。従来技術においては、キャラクタの滑らかな動きを再生するために、多数のフレームを使ってキャラクタの各点における位置情報や関節情報をアニメーションデータとして作成している。

【０００３】

通常、ゲーム機などにおけるアニメーションでは１秒間に３０フレームが表示されるが、そのフレームに対応したデータをすべて用意しておき、実行時に表示を行っている。このために、変化率の小さなアニメーションデータもあらかじめ用意しておかなければならず、冗長なデータもそうでないデータも同じようにアニメーションデータとして準備しておく必要があった。

【０００４】

例えば、従来のアニメーション表示では各フレームごとにデータ（x,y,z）を作成し、実機に表示していた。この場合に簡単な直線的な動きのキャラクタ動作も、各フレームごとに３成分データを必要とするために、冗長なデータ列となっている。

【０００５】

【発明が解決しようとする課題】

従来技術で説明したように、質の高いアニメーションを表示するためには単位時間あたり一定数以上のフレームが必要となる。しかし、そのためには多くの表示データが必要となり、コンピュータのメモリを圧迫する原因となっている。とくに３次元アニメーションデータの場合には、滑らかな動きを再生しようとするとそのデータ量が膨大になり、他の目的にメモリを活用しようとしてもできないという問題が発生することがある。

【０００６】

本発明の目的はこのような条件下でもアニメーションなどの画像の質を落とさずにデータ量を削減する方法を開発することである。

【０００７】

【課題を解決するための手段】

本発明は上記課題を解決するために、フレーム毎に作成された画像データを連続的に表示して動画表示を行うコンピュータ画像のデータ作成において、２つの特定フレーム間に表示されるフレームのデータが前記２つの特定フレームデータから補間近似により作成された近似データが元のフレームデータとの差が一定の値以下の場合に、前記２つの特定フレーム間のデータを前記２つの特定フレームデータをキーフレームデータとして代表して記録する。

【０００８】

そして、このようにして作成されたデータを用いて、キーフレーム間のフレームデータを近似により補間して作成した補間フレームデータを前記キーフレームのデータの間で順次表示して動画表示を行う。

【０００９】
本発明をアニメーションを例にしてさらに具体的に説明する。アニメーションの表示の対象となるキャラクタに対して、キーフレーム間のキャラクタの動きをもとに、各キーフレームにおけるキャラクタの対応点を直線近似（１次元近似）あるいは曲線近似（多次元近似）することによって、途中のフレームを作成してコンピュータ画像として表示する。
【００１０】
従来のアニメーション表示では各フレームごとに３次元のデータ（x,y,z）を作成し、画面表示していた。本発明では、冗長なフレームのデータ（三次元アニメーションの場合は３成分データ）をそれぞれ独立してキーフレームで補間することで、冗長的フレームを間引き、全体のデータ量を減らす。
【００１１】
たとえば、図１に示すようにキャラクタＡＢがフレームf_1～f_5で表される動作をしているとすると、途中のフレームf_2とf_4はf_1、f_3、f_5によって補間できる。このとき、f_1、f_3、f_5がキーフレームであり、f_2とf_4は補間によって作成されるフレームすなわち中間フレームである。
【００１２】
キーフレームと削減するフレーム（中間フレーム）との関係は許容誤差によってコントロールする。すなわち、キーフレームから中間フレームを決定するには、近似式によって決定される中間フレームのデータ点が誤差のしきい値内に収まるように区間を分割を行い、近似式によって算出される中間フレームの精度（質）を一定に保つようにする。以上の方法により、中間フレームf_2とf_4は元のデータから間引くことができ、しかも質を一定に保つことができる。
【００１３】
【発明の実施の形態】
本発明の実施の形態を図で説明する。まず最小二乗法を用いた直線近似の場合について説明する。３次元データの各成分は独立しているから、問題を簡単化するために１成分のみに着目して近似する。すなわち、与えられたデータを次のようなｎ個の２次元ベクトルとして表す。
$v_i = (f_i, r_i)$　　$(i=1,2,\cdots\cdots,n)$
ここではf_iはｉ番目のフレーム番号であり、r_iはf_iに対応した位置座標である。上記のデータ列を直線近似するために、
$r_i = a f_i + b$
とおく。表記を簡潔にするためにさらに行列を用いて数１のように置き換える。
【００１４】
【数１】

$$A\boldsymbol{k} = \boldsymbol{r}$$

ただし

$$A = \begin{bmatrix} f_1 & 1 \\ f_2 & 1 \\ \vdots & \vdots \\ f_n & 1 \end{bmatrix}, \quad \boldsymbol{k} = \begin{pmatrix} a \\ b \end{pmatrix}, \quad \boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

【００１５】
このとき標本点を近似するために最小二乗法を用いた直線近似（線形近似）を行う。直線近似を行う場合、最小二乗法のアルゴリズムは疑似逆行列を用いた線型方程式の解法と同値になる。そこでまず、疑似逆行列A^+を以下のように定義する。

$A^+ = (A^T A)^{-1} A^T$

【００１６】
この問題では疑似逆行列は２×２の正方行列となるので簡単に書き下すことができ、数２のように表せる。

【００１７】
【数２】

$$\det(A^T A) = \begin{pmatrix} \sum_{i=1}^{n} f_i^2 & \sum_{i=1}^{n} f_i \\ \sum_{i=1}^{n} f_i & n \end{pmatrix}$$

$$A^+ = \frac{1}{\det(A^T A)} \begin{pmatrix} nf_1 - \sum_{i=1}^{n} f_i & \cdots & nf_n - \sum_{i=1}^{n} f_i \\ \sum_{i=1}^{n} f_i^2 - f_1 \sum_{i=1}^{n} f_i & \cdots & \sum_{i=1}^{n} f_i^2 - f_n \sum_{i=1}^{n} f_i \end{pmatrix}$$

【００１８】
数２を用いると、近似的な係数ベクトルｋ´は、以下の数３のように求められる。このｋ´が最小二乗の意味で最適な係数ということになる。

【００１９】
【数３】

$$k' = (A^T A)^{-1} A^T$$

【００２０】
以上は数学的な考え方であるが、３成分データをプログラム的に削減する場合には探索アルゴリズムとして以下の２点を考慮する必要がある。
１．与えられたしきい値以下の誤差で近似可能な区間を選択し、一つの直線でなるべく長い区間を近似する。
２．キーフレームと３成分を対応させて扱う必要があるので、３成分をすべて加味して誤差評価をする。
【００２１】
さらに区間選択は再帰的なアルゴリズムによって行う。それにはまず、全体の点列の最初と最後の点を端点とし、両端点に収まる範囲を区間として設定し、許容誤差以内での直線近似で可能な限り両区間を伸ばしていく。
【００２２】
たとえば、図２のように二つの区間では十分に近似できない場合には、その区間に対しては同様の区間選択アルゴリズムを適用して区間の細分化を行う。
【００２３】
また二つの区間だけで近似できる場合には、左の区間を優先する。図３のように区間が重なるような場合には、右の区間の左端点を、左区間の右端点と一致するように移動する。なお図２、図３における黒点は連続して表示されるフレーム上の対応する点である。
【００２４】
図４は上記のアルゴリズムをフローチャートにまとめたものである。ただし、図４では、上記のアルゴリズムと若干異なり、簡単にするために開始の点ｓから順に近似的直線係数ベクトルｋ´を求めている。
【００２５】
すなわち、図５の例に示すように最初はｓ＝０、ｅ＝７から始め、点集合
Ｐ＝（p_s，p_2，……，p_e）
に対して係数ベクトルｋ´を求め、求められた近似直線l_1が誤差ε以内に収まっていないときにはｅ＝６として新たな近似直線l_1’を求める。その誤差がε以内かどうかを判断し、誤差以内に収まる範囲を探索する。
【００２６】
最終的に誤差の範囲内に収まる区間を区間１（図の例ではｓ＝０、ｅ＝４の範囲）とする。次に開始の端点ｓ＝４、ｅ＝７として同様な操作を行い、直線l_2を求める。以上の操作をｅ＝７（図４のフローチャートではｅ＝ｎ）になるまで繰り返す。図５の例ではｓ＝４、ｅ＝６が区間２、ｓ＝６、ｅ＝７が区間３を形成しているが、キーフレームを重視するなら、両端から近似直線を求めていたったほうがよい場合がある。しかし、通常は大差ない。区間の探索には２分法と呼ばれる方法を用いることもできる。
【００２７】
誤差評価について説明する。ある区間が近似（この場合は直線近似）できるかどうかを判断するには、まず最小二乗の意味で最適な係数を求める。すなわち、この時点で近似直線が得られるわけである。次に近似直線を用いて各キーフレームでの成分がどのような値になるのかを推定する。推定値と真のデータ点の値の誤差（差の絶対値）が真値のｃパーセント以下ならば、その点は十分に近似されているとする（ｃは図４の誤差εに相当）。逆にｃパーセント以上の誤差となった場合には、その区間は近似（直線近似）するには誤差が大きすぎるために、小区間に分割しなければならないことを意味している。
【００２８】
以上の判定をｘ成分、ｙ成分、ｚ成分すべてに対して行い、どの成分も許容できるようで

あれば、次のデータ点の評価に進む。最終的にすべてのデータ点で許容範囲に収まっている場合、その区間で近似（直線近似）が行えると結論づけられる。それが図４の“Ｅ≦ｔｈ”の判定であり、そのときの真のデータ点ｓ～ｅが近似（直線近似）される区間となる。

【００２９】

歩くような速度、あるいは高速の動きでも直線的な場合には、直線近似で十分である。しかし、曲線的に変化する場合には、直線近似では区間分割が多くなり、必ずしも有利でない。このようなときには、曲線式近似が有利になる。とくにゲームにおけるアニメーションの振り付けを作成するときにはスプライン補間を用いると便利なことが多い。近似式の作り方も以下で示すように基本的に直線近似と同じ方法で行うことができる。

【００３０】

まず直線近似の場合と同様に、簡単化するために一成分のみに着目し、与えられたデータを次のようなｎ個の２次元ベクトルと表す。

$v_i = (f_i, r_i)$　　$(i=1,2,\cdots\cdots,n)$

ここでは上記のデータ列を３次式

$r_i = a f_i^3 + b f_i^2 + c f_i + d$

で近似する。表記を簡潔にするためにさらに行列を用いて数４のように置き換える。

【００３１】

【数４】

$$A\boldsymbol{k} = \boldsymbol{r}$$

ここで

$$A = \begin{bmatrix} f_1^3 & f_1^2 & f_1 & 1 \\ f_2^3 & f_2^2 & f_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ f_n^3 & f_n^2 & f_n & 1 \end{bmatrix}, \quad \boldsymbol{k} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \quad \boldsymbol{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

【００３２】

したがって、直線近似の場合と同様に、近似的な係数ベクトルｋ´は数５と求められる。このｋ´を用いて最小のデータ点で許容誤差を満足するアニメーションデータを作成することになる。

【００３３】

【数５】

$$\boldsymbol{k}' = (A^T A)^{-1} A^T \boldsymbol{r}$$

【００３４】

実際に実行時に補間処理で求めた係数ベクトルｋ´を用いてフレーム番号f_iに対応したr_iを

ｒ＝Ａｋ´
の計算式で求めることによって、キーフレームｆ$_1$からキーフレームｆ$_n$までの間を補間できる。すなわち、この間のフレームは近似式で求めることができ、実際のデータから削減できる。この方法は直線近似あるいは曲線近似にかかわらず、いったん係数ベクトルを求めておけば、同様な方法で補間できることも本発明の特徴である。
【００３５】
【実施例】
ある物体（キャラクタ）にアニメーションをさせる場合、本来ならばすべてのフレームに対してそのキャラクタの形状を定義しておかなければならない。しかし、それではデータ量が膨大になってしまう。そこで本発明では、ある特定なフレームに対してのみ形状を定義し、他のフレームは数学的手法によって形状データを作成して表示しようというものである。この形状を定義するフレームを本発明ではキーフレームとよんでいる。
【００３６】
たとえば、ゲームにおける通常のアニメーションにおいては１秒間に３０フレームを表示するが、それを１、１０、２０、３０フレームのみのをキーフレームとして形状を定義しておき、その間のフレームはキーフレームから数学的に算出するというものである。図６の例では人が歩く動作をＡ、Ｂ、Ｃの３フレームのみをキーフレームとして形状データを用意しておき、その間のフレームは本発明の実施の形態で述べたような直線近似あるいはスプライン近似で算出される。
この近似式によって算出されるフレームを本発明では中間フレームとよんでいる。
【００３７】
図６の例の場合には、“Ａ→Ｂ→Ｃ→Ｂ→Ａ→Ｂ→Ｃ→……”というフレームの繰り返しになるから、その間の中間フレームは直線近似式または曲線近似式によって算出されることになる。
【００３８】
本発明の手法を用いて野球のバッターの腰の動きを表すパスデータを、実際のデータと直線近似式で求めたデータとを比較解析したデータがあるので、実施例として紹介しておく。元の形状データ数（フレーム数）は４５のものを許容誤差0.01（=1%）で削減すると、１８個まで削減できる。このときのｘ、ｙ、ｚ成分に対応した元のデータと補間データとの動きの比較をグラフで表したものが図７、図８、図９である。ｘ、ｙは地面に平行な座標値、ｚは地面に垂直方向＜高さ方向＞の座標値である。図８に示すように動作の開始のところのｙ成分だけが真のデータとの若干のズレがあるものの、全体的に一致している。このことは、４５フレームを１８フレーム（１８個のキーフレーム）にしても十分に元の形状を再現できることを表している。
【００３９】
さらに精度を落として許容誤差を0.1（=10%）にした場合には、１４フレームまで削減できた。このときの真のデータと近似データとを同様に比較表示すると、図１０、図１１、図１２のようになる。この場合のｙ成分における真のデータと近似データのズレが動作の開始から中程まで広がっているが、全体のズレは小さい。すなわち、４５フレームを１４のフレームまで削減しても十分に近似表現ができることを表している。
【００４０】
では、何フレームまで削減できるか。そのテスト結果を図１３に示してある。図のグラフは誤差とフレーム数の関係を示したものであり、横軸が誤差、縦軸がフレーム数である。この結果からもわかるように、フレーム数が１４を境に誤差が急激に大きくなっている。すなわち実用上、誤差は0.05以下の範囲で調整することが賢明であるといえる。しかし、これはあくまでもこの例の場合であり、アニメーションによってその許容誤差は異なる。
【００４１】
【発明の効果】
これまでのアニメーションにおいては、実機に表示するデータをすべてフレームごとに定義していた。このため、膨大なデータ量を必要とし、アニメーションデータがメモリを圧

迫していた。この点を解決したのが本発明のデータ点削減方法である。すなわち、本発明を用いることによって中間フレームの形状データを数学的に算出できるから、冗長的な中間フレームの形状データをもつ必要がなく、データ量を大幅に減らすことができる。

【００４２】

しかも許容誤差によってデータ点の削減数を調整できるために、この誤差の設定の仕方によってアニメーションの質を一定に保つことができる。このことは、アニメーションの質を落とさずにデータ量の削減度合いを調整することができることを意味している。例えば、本発明の実施例で説明したように、本発明の方法を用いないときには４５フレーム分の形状データが必要であったものを、本発明の方法を用いることによって１４フレームまでに削減することができる。すなわち、実施例の場合には約３分の１までデータ量が減らせたことになる。とくに、キャラクタの動きの変化率が少ないものに対してはさらに多くのデータを削減できる。

【図面の簡単な説明】

【図１】本発明の課題を解決するための手段におけるキーフレームと中間フレームの意味を説明するための図である。

【図２】本発明の実施の形態における直線近似と区間分割の意味を説明するための図である。

【図３】本発明の実施の形態における直線近似と区間分割の意味を説明するための図である。

【図４】本発明の実施の形態におけるデータ点削減アルゴリズムをフローチャートで失明した図である。

【図５】本発明の実施の形態における図４のフローチャートを補足説明するための図である。

【図６】本発明の実施例におけるキーフレームの表示例である。

【図７】本発明の実施例における元のデータ点と許容誤差0.01で近似したデータ点との動作比較をグラフで表したものである（縦軸はｘ成分、横軸はフレーム番号）。

【図８】本発明の実施例における元のデータ点と許容誤差0.01で近似したデータ点との動作比較をグラフで表したものである（縦軸はｙ成分、横軸はフレーム番号）。

【図９】本発明の実施例における元のデータ点と許容誤差0.01で近似したデータ点との動作比較をグラフで表したものである（縦軸はｚ成分、横軸はフレーム番号）。

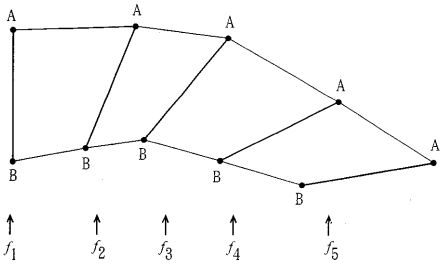
【図１０】本発明の実施例における元のデータ点と許容誤差0.1で近似したデータ点との動作比較をグラフで表したものである（縦軸はｘ成分、横軸はフレーム番号）。

【図１１】本発明の実施例における元のデータ点と許容誤差0.1で近似したデータ点との動作比較をグラフで表したものである（縦軸はｙ成分、横軸はフレーム番号）。

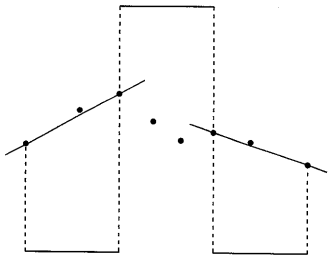
【図１２】本発明の実施例における元のデータ点と許容誤差0.1で近似したデータ点との動作比較をグラフで表したものである（縦軸はｚ成分、横軸はフレーム番号）。

【図１３】本発明の実施例における誤差とキーフレーム数の関係をグラフで表したものである（縦軸はフレーム数、横軸は誤差）。

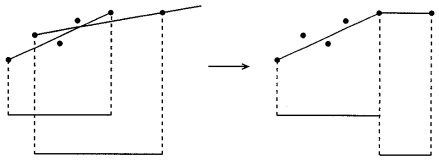
【図１】



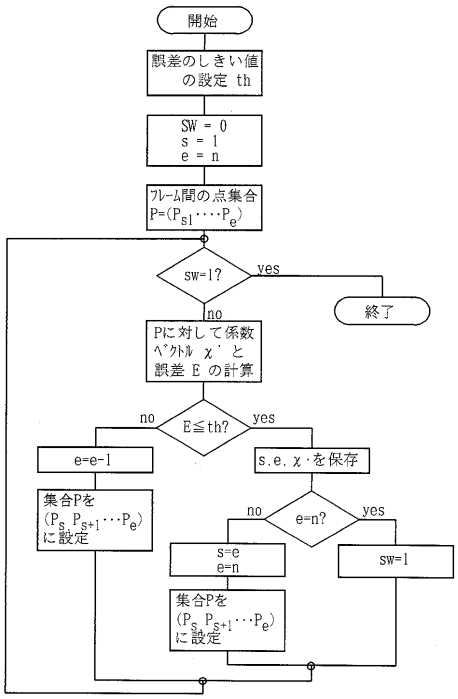
【図２】



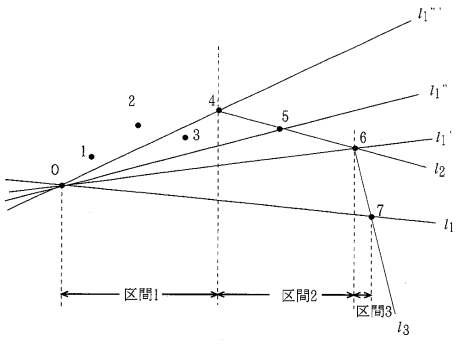
【図３】



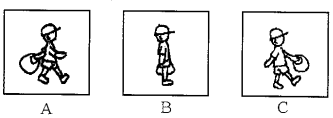
【図４】



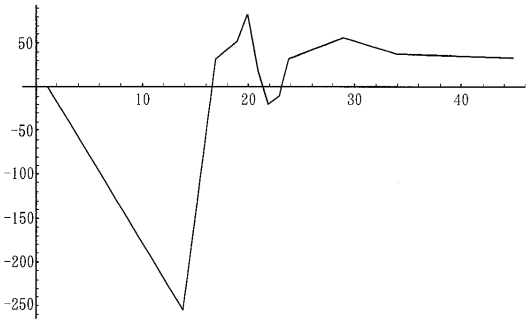
【図５】



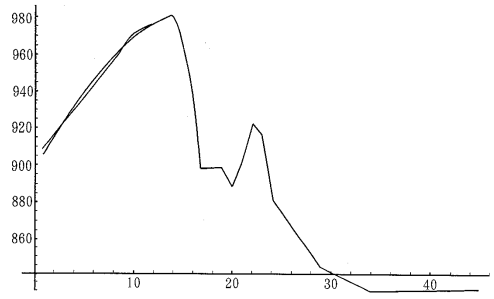
【図６】



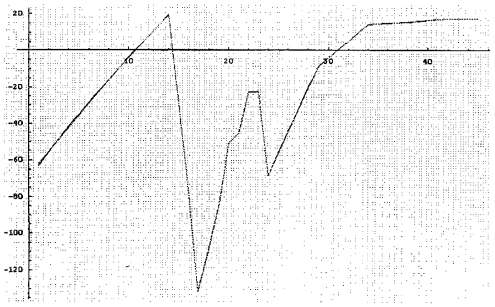
【図７】



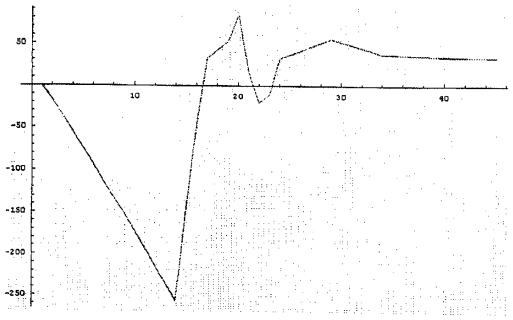
【図８】



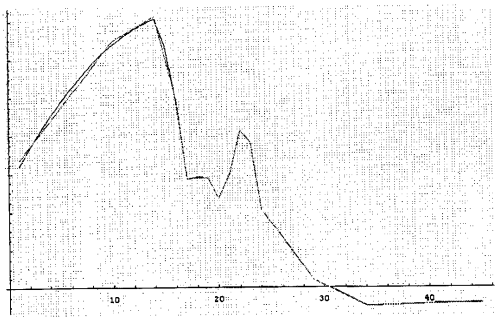
【図９】



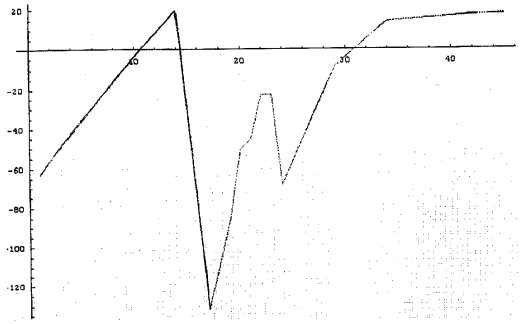
【図１０】



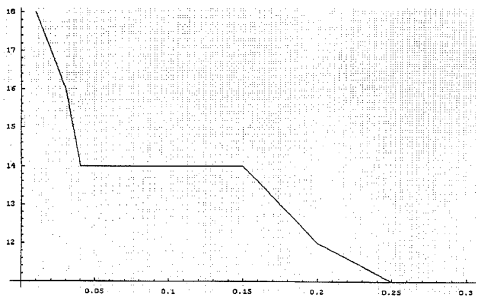
【図１１】



【図１２】



【図１３】



フロントページの続き

(56)参考文献 特開平０６－１４９６９４（ＪＰ，Ａ）
特開平０６－２７６４８２（ＪＰ，Ａ）
特開平０８－２３７６６５（ＪＰ，Ａ）
特開平０４－２８８７９０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.[7]，ＤＢ名)
G06T 13/00 - 17/50
H04N 5/91 - 5/956