

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6262621号
(P6262621)

(45) 発行日 平成30年1月17日(2018.1.17)

(24) 登録日 平成29年12月22日(2017.12.22)

(51) Int. Cl. F I
 HO4N 1/393 (2006.01) HO4N 1/393
 G06T 3/40 (2006.01) G06T 3/40

請求項の数 6 (全 22 頁)

(21) 出願番号	特願2014-178713 (P2014-178713)	(73) 特許権者	591128453 株式会社メガチップス 大阪府大阪市淀川区宮原一丁目1番1号
(22) 出願日	平成26年9月3日(2014.9.3)	(74) 代理人	100088672 弁理士 吉竹 英俊
(65) 公開番号	特開2015-89115 (P2015-89115A)	(74) 代理人	100088845 弁理士 有田 貴弘
(43) 公開日	平成27年5月7日(2015.5.7)	(72) 発明者	水野 雄介 大阪市淀川区宮原一丁目1番1号 株式会 社メガチップス内
審査請求日	平成29年5月22日(2017.5.22)	審査官	鈴木 明
(31) 優先権主張番号	特願2013-198092 (P2013-198092)		
(32) 優先日	平成25年9月25日(2013.9.25)		
(33) 優先権主張国	日本国(JP)		

最終頁に続く

(54) 【発明の名称】 画像の拡大縮小処理装置および画像の拡大縮小処理方法

(57) 【特許請求の範囲】

【請求項1】

入力画像データに対して画素を補間することでU/D倍(U、Dは自然数)に変換して出力画像データとする画像の拡大縮小処理装置であって、

前記画像の拡大縮小処理で使用される補間係数を算出する係数演算回路と、

前記入力画像データを構成する入力画素のそれぞれに対応するように前記係数演算回路から与えられる前記補間係数を乗算する乗算器と、

前記乗算器から出力される乗算済みデータの加算を繰り返し、前記入力画素の所定個数分について前記乗算済みデータの総和を取る加算器と、

前記所定個数分の前記入力画素について前記乗算済みデータの総和が得られるタイミングで前記乗算済みデータの総和を出力するセレクタと、

前記セレクタの出力をシフト演算することで、前記出力画像データのビット数を前記入力画像データのビット数に調整するシフト回路と、を備え、

前記係数演算回路で算出される前記補間係数は、その総和が、所定数の2の累乗と同じとなるように整数化されていることを特徴とする、画像の拡大縮小処理装置。

【請求項2】

前記係数演算回路は、

前記入力画像データに対して補間される画素の内挿点の座標を算出する座標演算回路と、

前記所定個数分の前記入力画素に対応する補間係数をそれぞれ出力する複数の係数演算

10

20

回路とを有し、

前記複数の係数演算回路のうちの1つは、前記所定数の2の累乗の値から他の係数演算回路で算出された前記補間係数の和を差し引くことで残り1つの補間係数を算出する、請求項1記載の画像の拡大縮小処理装置。

【請求項3】

前記座標演算回路は、

先に得られた内挿点の座標に D/U を加算することで次の内挿点の座標を算出し、そのうち、最初の内挿点の座標は、予め定めた初期値に D/U を加算することで座標を算出し、

算出された内挿点の座標の小数部分を切り捨てる演算を行って、その演算結果を前記内挿点と前記所定個数分の前記入力画素のうちの1つの入力画素との基準距離とし、前記基準距離に基づいて、前記内挿点と前記所定個数分の前記入力画素の他の入力画素との距離を算出し、前記基準距離を含めて内挿点入力画素間距離として前記他の係数演算回路に与える、請求項2記載の画像の拡大縮小処理装置。

10

【請求項4】

前記座標演算回路は、

現在の内挿点の座標と次の内挿点の座標とを整数化し、両者の差分値により、前記入力画素を前記所定個数分単位の選択範囲を、所定画素数ずつずらす、あるいはずらさないことを規定するスキップ数を決定する、請求項3記載の画像の拡大縮小処理装置。

【請求項5】

前記他の係数演算回路は、

前記座標演算回路から与えられる前記内挿点入力画素間距離どうしの乗算および所定の係数の乗算を行って、入力された前記内挿点入力画素間距離に対応した補間関数を演算すると共に、前記補間関数の演算結果を四捨五入することで、入力された前記内挿点入力画素間距離に対応した補間係数を算出する、請求項3記載の画像の拡大縮小処理装置。

20

【請求項6】

入力画像データに対して画素を補間することで U/D 倍（ U 、 D は自然数）に変換して出力画像データとする画像の拡大縮小処理方法であって、

- (a) 前記画像の拡大縮小処理で使用される補間係数を算出するステップと、
- (b) 前記入力画像データを構成する入力画素のそれぞれに対応するように、前記ステップ(a)で得られた前記補間係数を乗算するステップと、
- (c) 前記ステップ(b)で得られた乗算済みデータの加算を繰り返し、前記入力画素の所定個数分について前記乗算済みデータの総和を取るステップと、
- (d) 前記所定個数分の前記入力画素について前記乗算済みデータの総和が得られるタイミングで前記乗算済みデータの総和を出力するステップと、
- (e) 前記ステップ(d)で得られる前記乗算済みデータの総和をシフト演算することで、前記出力画像データのビット数を前記入力画像データのビット数に調整するステップと、を備え、

30

前記ステップ(a)で算出される前記補間係数は、その総和が、所定数の2の累乗と同じとなるように整数化されていることを特徴とする、画像の拡大縮小処理方法。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像の拡大縮小処理で使用される補間係数を算出する係数演算回路を有した画像の拡大縮小演算装置に関する。

【背景技術】

【0002】

一般的な画像の拡大縮小処理装置としては、例えば特許文献1に開示されるように、まず、原画像を U 倍に拡大した後、 $1/D$ 倍に縮小するという手順で、整数倍だけでなく有理数倍も含めた U/D 倍任意拡大縮小処理を実現している。しかしながら、この方法では

50

、原画像をU倍に拡大するための演算が必要であると共に、U倍に拡大した画像データを格納するためのメモリが必要となり、演算およびメモリが冗長性を含むこととなる。

【0003】

一方、特許文献2には、原画像から中間画像(U倍画像)を求めずにいきなり最終画像(U/D倍画像)を求める技術が開示されており、冗長性が解消されているが、補間処理については従来的な手法を用いており、割り算器や浮動小数点演算器を用いる必要があると考えられる。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開平10-63826号公報

【特許文献2】特開2000-165664号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

以上説明したように、従来の画像の拡大縮小処理においては、割り算器や浮動小数点演算器を用いる必要があり、装置構成が複雑になるという問題があった。

【0006】

本発明は上記のような問題点を解消するためになされたもので、割り算器や浮動小数点演算器を用いることなしに画像を任意の倍数で拡大縮小できる画像の拡大縮小処理装置を提供することを目的とする。

【課題を解決するための手段】

【0007】

本発明に係る画像の拡大縮小処理装置の第1の態様は、入力画像データに対して画素を補間することでU/D倍(U、Dは自然数)に変換して出力画像データとする画像の拡大縮小処理装置であって、前記画像の拡大縮小処理で使用される補間係数を算出する係数演算回路と、前記入力画像データを構成する入力画素のそれぞれに対応するように前記係数演算回路から与えられる前記補間係数とを乗算する乗算器と、前記乗算器から出力される乗算済みデータの加算を繰り返し、前記入力画素の所定個数分について前記乗算済みデータの総和を取る加算器と、前記所定個数分の前記入力画素について前記乗算済みデータの総和が得られるタイミングで前記乗算済みデータの総和を出力するセレクタと、前記セレクタの出力をシフト演算することで、前記出力画像データのビット数を前記入力画像データのビット数に調整するシフト回路とを備え、前記係数演算回路で算出される前記補間係数は、その総和が、所定数の2の累乗と同じとなるように整数化されている。

【0008】

本発明に係る画像の拡大縮小処理装置の第2の態様は、前記係数演算回路が、前記入力画像データに対して補間される画素の内挿点の座標を算出する座標演算回路と、前記所定個数分の前記入力画素に対応する補間係数をそれぞれ出力する複数の係数演算回路とを有し、前記複数の係数演算回路のうちの1つは、前記所定数の2の累乗の値から他の係数演算回路で算出された前記補間係数の和を差し引くことで残り1つの補間係数を算出する。

【0009】

本発明に係る画像の拡大縮小処理装置の第3の態様は、前記座標演算回路が、先に得られた内挿点の座標にD/Uを加算することで次の内挿点の座標を算出し、そのうち、最初の内挿点の座標は、予め定めた初期値にD/Uを加算することで座標を算出し、算出された内挿点の座標の小数部分を切り捨てる演算を行って、その演算結果を前記内挿点と前記所定個数分の前記入力画素のうちの1つの入力画素との基準距離とし、前記基準距離に基づいて、前記内挿点と前記所定個数分の前記入力画素の他の入力画素との距離を算出し、前記基準距離を含めて内挿点入力画素間距離として前記他の係数演算回路に与える。

【0010】

本発明に係る画像の拡大縮小処理装置の第4の態様は、前記座標演算回路が、現在の内

10

20

30

40

50

挿点の座標と次の内挿点の座標とを整数化し、両者の差分値により、前記入力画素を前記所定個数分単位の選択範囲を、所定画素数ずつずらす、あるいはずらさないことを規定するスキップ数を決定する。

【0011】

本発明に係る画像の拡大縮小処理装置の第5の態様は、前記他の係数演算回路が、前記座標演算回路から与えられる前記内挿点入力画素間距離どうしの乗算および所定の係数の乗算とを行って、入力された前記内挿点入力画素間距離に対応した補間関数を演算すると共に、前記補間関数の演算結果を四捨五入することで、入力された前記内挿点入力画素間距離に対応した補間係数を算出する。

【0012】

本発明に係る画像の拡大縮小処理方法は、入力画像データに対して画素を補間することでU/D倍（U、Dは自然数）に変換して出力画像データとする画像の拡大縮小処理方法であって、前記画像の拡大縮小処理で使用される補間係数を算出するステップ（a）と、前記入力画像データを構成する入力画素のそれぞれに対応するように、前記ステップ（a）で得られた前記補間係数を乗算するステップ（b）と、前記ステップ（b）で得られた乗算済みデータの加算を繰り返し、前記入力画素の所定個数分について前記乗算済みデータの総和を取るステップ（c）と、前記所定個数分の前記入力画素について前記乗算済みデータの総和が得られるタイミングで前記乗算済みデータの総和を出力するステップ（d）と、前記ステップ（d）で得られる前記乗算済みデータの総和をシフト演算することで、前記出力画像データのビット数を前記入力画像データのビット数に調整するステップ（e）とを備え、前記ステップ（a）で算出される前記補間係数は、その総和が、所定数の2の累乗と同じとなるように整数化されている。

【発明の効果】

【0013】

本発明によれば、補間係数の総和が、所定数の2の累乗と同じとなるように整数化されているので、割り算器や浮動小数点演算器を用いることなしに画像を任意の倍数で拡大縮小処理を行うことができる。

【図面の簡単な説明】

【0014】

【図1】U/D倍任意拡大縮小処理を模式的に示すブロック図である。

【図2】画像をU倍に拡大する場合の伝達関数を示す図である。

【図3】画像を1/D倍に縮小する場合の伝達関数を示す図である。

【図4】画像をU/D倍に拡大する場合の伝達関数を示す図である。

【図5】画像をU/D倍に縮小する場合の伝達関数を示す図である。

【図6】キュービックコンポリューションによる補間を説明する図である。

【図7】本発明に係る実施の形態の画像の拡大縮小処理装置の基本構成を示すブロック図である。

【図8】U/D倍する場合のPitchと初期値との関係を説明する図である。

【図9】U/D倍する場合のPitchと初期値との関係を説明する図である。

【図10】入力画像の画素データの選択範囲をスキップさせる動作を説明する図である。

【図11】入力画像の画素データの選択範囲をスキップさせる動作を説明する図である。

【図12】入力画像の画素データの選択範囲をスキップさせる動作を説明する図である。

【図13】入力画像の画素データの選択範囲をスキップさせる動作を説明する図である。

【図14】係数演算回路の構成を示すブロック図である。

【図15】座標演算回路の構成を示すブロック図である。

【図16】係数演算回路の構成を示すブロック図である。

【図17】係数演算回路の構成を示すブロック図である。

【図18】座標演算回路の変形例の構成を示すブロック図である。

【図19】座標演算回路の変形例の構成を示すブロック図である。

【図20】座標演算回路の変形例の構成を示すブロック図である。

10

20

30

40

50

【図 2 1】プロセッサの構成を説明するブロック図である。

【発明を実施するための形態】

【0015】

< 始めに >

実施の形態の説明に先立って、本発明の前提となる画像の拡大縮小処理について説明する。

【0016】

< 伝達関数を用いた例 >

まず、伝達関数を用いた画像の拡大縮小処理の例として、冗長性を排除した U / D 倍任意拡大縮小処理の手順について説明する。

10

【0017】

図 1 は、U / D 倍任意拡大縮小処理を模式的に示すブロック図であり、入力画像データ $x(n)$ がアップサンプラ UP に入力され、アップサンプリングされた画像データがローパスフィルタであるデジタルフィルタ DF に与えられ、デジタルフィルタ DF で帯域制限を受けた画像データ $d(n)$ がダウンサンプラ DW に与えられ、ダウンサンプリングされた出力画像データ $y(n)$ が出力される構成となっている。

【0018】

ここで、アップサンプラ UP は、原画像信号の各信号間に画素 $U - 1$ 個分の 0 値を挿入する処理を行い、ダウンサンプラ DW は、画像データから D 画素ごとに残し、残りを間引く処理を行う。また、デジタルフィルタ DF のインパルス応答は $h_{UD}(n)$ であり、図 1 の処理は畳み込み演算を示す以下の数式 (1) で表される。

20

【0019】

【数 1】

$$y(n) = \sum_{k=-\infty}^{\infty} x(k) h_{UD}(Dn - Uk) \cdots (1)$$

【0020】

ここで、画像を U 倍に拡大する場合の理想的なフィルタのインパルス応答 $h_U(n)$ の伝達関数 $H_U(z)$ は、 z 軸 ($z = \exp(j\omega T)$) 上の表現で図 2 のように表され、また、 $1 / D$ 倍に縮小する場合の理想的なフィルタのインパルス応答 $h_D(n)$ の伝達関数 $H_D(z)$ は図 3 のように表される。なお、図 2 および図 3 において、横軸は正規化角周波数 (ω) を表し、縦軸は振幅を表している。

30

【0021】

これに対し、 U / D 倍に拡大する場合、すなわち $U > D$ の場合の理想的なフィルタのインパルス応答 $h_{UD}(n)$ の伝達関数 $H_{UD}(z)$ は図 4 のように表される。また、 U / D 倍に縮小する場合、すなわち $U < D$ の場合の理想的なフィルタのインパルス応答 $h_{UD}(n)$ の伝達関数 $H_{UD}(z)$ は図 5 のように表される。なお、図 4 および図 5 において、横軸は正規化角周波数 (ω) を表し、縦軸は振幅を表している。

【0022】

40

なお、図 4 と図 2 に示す画像を U 倍に拡大する場合の伝達関数 $H_U(z)$ は同じであり、共に遮断角周波数は π / U であるので、 U 倍に拡大する場合のフィルタを、 U / D 倍に拡大する場合のフィルタとして使用可能である。また、図 5 の画像を U / D 倍に縮小する場合の伝達関数 $H_{UD}(z)$ の遮断角周波数は π / D になる。

【0023】

このように、デジタルフィルタ DF をインパルス応答 $h_{UD}(n)$ のフィルタとすることで、画像の拡大用フィルタと縮小用フィルタの 2 つを備える必要がなくなり、演算量を削減することができる。また、一旦、 U 倍に拡大する必要がないので、 U 倍に拡大した画像データを格納するためのメモリも不要となる。

【0024】

50

ここで、画像を任意の倍数で拡大縮小する場合、フィルタを適切に選ばないと画素周期 U で碁盤目状のひずみ（チェス盤ひずみ（checkerboard distortion））が発生する。

【0025】

これに対しては、以下の数式（2）で表されるように、インパルス応答 $h_{UD}(n)$ を画素 U 個ごとの区切りとみなして、それぞれの区切りの 0 番目の総和、1 番目の総和 ~ $U - 1$ 番目の総和が、それぞれ一定値になるような伝達関数を設定することでチェス盤ひずみを回避することができる。

【0026】

【数2】

$$\sum_{k=-\infty}^{\infty} h_{UD}(-Uk) = \sum_{k=-\infty}^{\infty} h_{UD}(1-Uk) = \dots = \sum_{k=-\infty}^{\infty} h_{UD}(U-1-Uk) = \text{一定} \dots (2)$$

10

【0027】

< 補間関数を用いた例 >

伝達関数を用いて、画像の拡大縮小を行う方法の他に、キュービックコンボリューションや線形補間、3次補間、ラグランジュ補間、スプライン補間、ランツォシュ補間などの補間関数を用いて画像データを内挿（補間）し、画像の拡大縮小を行う方法がある。

【0028】

例えば、特開2001-189850号公報では、キュービックコンボリューションを用いた内挿の例が説明されている。以下、キュービックコンボリューションを用いた内挿について説明する。

20

【0029】

【数3】

$$h(t) = \begin{cases} (a+2)|t|^3 - (a+3)|t|^2 + 1, & 0 \leq |t| < 1 \\ a|t|^3 - 5a|t|^2 + 8a|t| - 4a, & 1 \leq |t| < 2 \\ 0, & 2 \leq |t| \end{cases} \dots (3)$$

30

【0030】

上記数式（3）はキュービックコンボリューションで用いる補間関数 $h(t)$ を表す数式である。

【0031】

数式（3）において、 a は補間関数の性質を制御するための変数（ $-0.5 \sim -2$ ）であり、 t は隣接する画素から内挿点までの距離を表す。

【0032】

また、図6にはキュービックコンボリューションの内挿点の一例を示している。図6においては、直列に等間隔で並んだ4点の画素 P_i 、 P_{i+1} 、 P_{i+2} および P_{i+3} がある場合に、画素 P_{i+1} と画素 P_{i+2} との間に内挿点 X が内挿される場合を示している。

40

【0033】

そして、内挿点 X からの画素 P_i 、 P_{i+1} 、 P_{i+2} および P_{i+3} までの距離をそれぞれ t_i 、 t_{i+1} 、 t_{i+2} および t_{i+3} とし、画素間の距離を 1.0 として示している。それぞれの画素に対する補間関数 $h(t_i)$ 、 $h(t_{i+1})$ 、 $h(t_{i+2})$ および $h(t_{i+3})$ は以下の数式（4）で表される。なお、距離 t_i 、 t_{i+1} 、 t_{i+2} および t_{i+3} は内挿点入力画素間距離と呼称する場合もある。

【0034】

【数4】

$$\begin{cases} h(t_i) = a|t_i|^3 - 5a|t_i|^2 + 8a|t_i| - 4a \\ h(t_{i+1}) = (a+2)|t_{i+1}|^3 - (a+3)|t_{i+1}|^2 + 1 \\ h(t_{i+2}) = (a+2)|t_{i+2}|^3 - (a+3)|t_{i+2}|^2 + 1 \\ h(t_{i+3}) = a|t_{i+3}|^3 - 5a|t_{i+3}|^2 + 8a|t_{i+3}| - 4a \end{cases} \quad \dots (4)$$

10

【0035】

上記数式(4)で表される補間関数 $h(t_i)$ 、 $h(t_{i+1})$ 、 $h(t_{i+2})$ および $h(t_{i+3})$ を画素 P_i 、 P_{i+1} 、 P_{i+2} および P_{i+3} の座標にそれぞれ掛け合わせた総和が内挿点 X の位置となる。これを表す数式が、以下に示す数式(5)である。

【0036】

【数5】

$$X = P_i \times h(t_i) + P_{i+1} \times h(t_{i+1}) + P_{i+2} \times h(t_{i+2}) + P_{i+3} \times h(t_{i+3}) \quad \dots (5)$$

20

【0037】

<実施の形態>

<拡大縮小処理装置の基本構成>

図7は本発明に係る実施の形態の画像の拡大縮小処理装置100の基本構成を示すブロック図である。なお、以下においては、拡大縮小処理装置100は、図1に示したU/D倍任意拡大縮小処理を行うものとして説明する。

【0038】

図7に示されるように、画像の拡大縮小処理装置100は、DMA(Direct Memory Access)バス19を介して主記憶装置30に格納された入力画像データ $x(n)$ を直接に読み出すDMA読み取り装置(DMAR)5と、DMAバス19を介して主記憶装置30に出力画像データ $y(n)$ を直接に書き込むDMA書き込み装置(DMAW)6とをデータ入出力部として備えている。

30

【0039】

DMAR5が読み出した1画素分の入力画像データ $x(n)$ は、セレクタ1を介してピンポンバッファ10に書き込まれる。ピンポンバッファ10は、セレクタ1の2つの出力と、セレクタ2の2つの入力との間に互いに並列に接続されたバッファ11およびバッファ12を有しており、バッファ11および12は書き込み用と読み取り用に順番に入れ替わるように制御される。なお、ピンポンバッファ10は、デュアルポートSRAMで構成することも可能である。デュアルポートSRAMは、データを入出力するためのポートを2つ有し、一方のポートからデータを読み出すのと同時に他方のポートからデータを書き込むことが可能であり、バッファ11および12を有する場合と同様の動作が可能である。

40

【0040】

セレクタ1は、バッファ11および12のうち、書き込み可能な方のバッファを選択し、入力画像データ $x(n)$ を書き込む。セレクタ2は、バッファ11および12のうち、入力画像データ $x(n)$ が書き込み済みのバッファから画像データを読み出す。

【0041】

従って、例えば、バッファ11に入力画像データ $x(n)$ が書き込まれているタイミン

50

グでは、バッファ 12 から、既にかき込まれた入力画像データ $x(n)$ が読み出されていることとなる。

【0042】

なお、セクタ 1 による書き込み可能なバッファの選択は制御装置 7 によりセクタ 1 が制御されることで実行され、セクタ 2 による書き込み済みのバッファの選択は制御装置 7 によりセクタ 2 が制御されることで実行される。

【0043】

セクタ 2 を介してピンポンバッファ 10 から読み出された入力画像データ $x(n)$ は、乗算器 8 に入力され、また、デジタルフィルタ DF (図 1) のインパルス応答 $h_{UD}(n)$ を補間係数として演算する係数演算回路 20 からは、セクタ 3 を介して補間係数が出力され、乗算器 8 に入力される。

10

【0044】

なお、係数演算回路 20 では、拡大、縮小の倍率にそれぞれ対応した補間係数、例えば、2 倍、3 倍、あるいは $1/2$ 倍、 $3/4$ 倍などに対応した補間係数を制御装置 7 が係数演算回路 20 を制御して算出することができる。

【0045】

乗算器 8 では、入力画像データ $x(n)$ と補間係数 $h_{UD}(n)$ の乗算を行い、乗算済みデータが加算器 9 に与えられる。加算器 9 には、フリップフロップ 13 内に格納されている加算済みデータがセクタ 4 を介して与えられ、加算器 9 で、乗算器 8 から与えられる乗算済みデータと加算される。

20

【0046】

乗算済みデータが加算された加算済みデータは、フリップフロップ 13 に格納されることでフリップフロップ 13 のデータを更新する。なお、フリップフロップ 13 の初期値は 0 であり、1 画素分の画像データに対して最初に加算器 9 で乗算済みデータに加算されるのは初期値 0 である。

【0047】

加算器 9 での加算は数式 (1) で表される $h_{UD}(Dn - Uk)$ と $x(k)$ の積の累積算である。なお、乗算器 8 と加算器 9 とをそれぞれ複数備えることで、演算の並列性が高まり処理速度を高速化することが可能である。

【0048】

セクタ 4 は、予め設定された選択範囲の画素数分の画像データに対して数式 (1) の累積算が終了するまでは加算器 3 に対してフリップフロップ 13 に格納されたデータを与えるが、予め設定された選択範囲の画素数分の画像データに対する累積算が終了した後はフリップフロップ 13 に格納されたデータをシフト回路 14 に与え、フリップフロップ 13 が 0 に初期化される。なお、セクタ 4 の選択動作は制御装置 7 により制御される。

30

【0049】

シフト回路 14 は、シフトレジスタなどで構成され、セクタ 4 より出力された累積演算結果のデータをシフト演算することで、出力画像データのビット数を入力画像データのビット数と一致させクリップ回路 15 に与える。なお、シフト回路 14 は、出力画像データのビット数は入力画像データのビット数と一致させる場合や演算精度を高くするために出力画像データのビット数が入力画像より大きい場合や演算精度を低くするために出力画像データのビット数が入力画像より小さい場合がある。

40

【0050】

クリップ回路 15 は、シフト回路 14 より出力され、ビット数が調整された演算結果のデータを、予め設定される最小値と最大値の範囲にデータを切り取り、出力画像データ $y(n)$ を生成する。出力画像データ $y(n)$ は DMAW6 から DMA バス 19 を介して、主記憶装置 30 に直接にかき込まれる。

【0051】

< 出力画像データの作成方法 >

このようにして得られた出力画像データ $y(n)$ は、予め設定された画素数分の入力画

50

像データに基づいて作成された1画素分の画像データであり、同じ処理を異なる選択範囲の予め設定された画素数分の画素に対して行い、次の出力画像データ $y(n)$ を得る。

【0052】

これを、1画像の全ての画素に対して繰り返し実行することで、数式(1)に基づいて拡大あるいは縮小された画像が得られる。

【0053】

図7に示す拡大縮小処理装置100においては、伝達関数からインパルス応答 $h_{UD}(n)$ を算出し、補間係数を係数演算回路20で算出する構成を採っている。

【0054】

ここで、図6に示される内挿点 X を出力 $y(0)$ とし、その座標を初期値 b_0 とし、出力 $y(j)$ の座標を b_j とすると、座標 b_j と座標 b_0 の関係は以下の数式(6)の第1式のように表され、また、 t_i 、 t_{i+1} 、 t_{i+2} および t_{i+3} は、第2式~第5式のように表される。

【0055】

【数6】

$$\left\{ \begin{array}{l} b_j = b_0 + Pitch \times j \\ t_i = t_{i+1} + 1.0 \\ t_{i+1} = b_{j+1} - INT(b_j) \quad \dots (6) \\ t_{i+2} = 1.0 - t_{i+1} \\ t_{i+3} = t_{i+2} + 1.0 \end{array} \right.$$

【0056】

なお、上記数式(6)の第3式の $INT(b_j)$ は、座標 b_j から小数点以下を切り捨てる演算を表し、また、座標 b_j の値は前の座標 b_{j-1} の整数部より桁上がりした値で規定される。

【0057】

また、上記数式(6)は、以下の数式(7)のように変形することができる。

【0058】

【数7】

$$\left\{ \begin{array}{l} b_j = b_{j-1} + Pitch \\ t_i = t_{i+1} + 1.0 \\ t_{i+1} = b_j - INT(b_j) \quad \dots (7) \\ t_{i+2} = 1.0 - t_{i+1} \\ t_{i+3} = 2.0 - t_{i+1} \end{array} \right.$$

【0059】

ここで、図 8 を用いて、 U/D 倍する場合の $Pitch$ と初期値 b_0 の関係を説明する。図 8 においては、入力画像の画素データが $x(0)$ 、 $x(1)$ 、 \dots 、 $x(D-2)$ および $x(D-1)$ の D 個、出力画像の画素データが $y(0)$ 、 $y(1)$ 、 $y(2)$ 、 \dots 、 $y(U/2-1)$ 、 $y(U/2)$ 、 \dots 、 $y(U-3)$ 、 $y(U-2)$ 、 $y(U-1)$ の U 個あるものとして示している。

【0060】

入力画像の画素データ間の距離を 1.0 とし、出力画像の画素データ間の距離を $Pitch$ とすると、図 8 より以下の数式 (8) を得ることができる。

【0061】

【数 8】

$$(U-1) \times Pitch + 2b_0 = D-1 \quad \dots (8)$$

【0062】

ここで、 $Pitch = D/U$ として数式 (8) に代入すると、初期値 b_0 は以下の数式 (9) で表すことができる。

【0063】

【数 9】

$$b_0 = \frac{D-U}{2U} \quad \dots (9)$$

【0064】

なお、図 8 では、入力される画像と出力される画像の中心を同じにする場合を示しており、この場合、 $U=3$ 、 $D=2$ の場合 ($3/2$ 倍) では、初期値は $b_0 = -1/6$ となる。

【0065】

図 8 のように入力される画像と出力される画像の中心を同じにすると、出力画像の画素データ $y(0)$ および $y(U-1)$ が、入力画像の画素データ $x(0) \sim x(D-1)$ の外側に位置することになる。出力画像の画素データ $y(0)$ および $y(U-1)$ が、それぞれ入力画像の画素データ $x(0)$ および $x(D-1)$ と同じ位置になるようにしたい場合は、初期値 $b_0 = 0$ とすることで図 9 のようにすることができる。この場合、 $Pitch$ は以下の数式 (10) で表すことができる。

【0066】

【数 10】

$$Pitch = \frac{D-1}{U-1} \quad \dots (10)$$

【0067】

また、図 6 に示したように、4 点の画素に対して 1 つの内挿点を設定する場合を例に採ると、出力画像の画素データは、入力画像の画素データの選択範囲を所定方向にずらすことで得られる。以下、その動作を図 10 ~ 図 13 を用いて説明する。

【0068】

図 10 においては、入力画像の画素データ $x(-1)$ 、 $x(0)$ 、 $x(1)$ および $x(2)$ に基づいて内挿点となる出力画像データ $y(0)$ を作成する (補間する) 動作を模式的に表しており、画素データ $x(-1)$ 、 $x(0)$ 、 $x(1)$ および $x(2)$ に、それぞれ係数演算回路 20 で算出した補間係数を乗算することで出力画像の画素データ $y(0)$

10

20

30

40

50

を得る。

【 0 0 6 9 】

なお、画素データ $x(-1)$ は、画像で言うと左端の画素のない領域の画素データであり、画像を折り返して使用する対称拡張法により画素データが生成されている。

【 0 0 7 0 】

また、図 1 1 においては、入力画像の画素データ $x(-1)$ 、 $x(0)$ 、 $x(1)$ および $x(2)$ に基づいて内挿点となる出力画像の画素データ $y(1)$ を補間する動作を模式的に表しており、画素データ $x(-1)$ 、 $x(0)$ 、 $x(1)$ および $x(2)$ に、それぞれ係数演算回路 2 0 で算出した補間係数を乗算することで出力画像の画素データ $y(1)$ を得る。

10

【 0 0 7 1 】

また、図 1 2 においては、画素データの選択範囲を右に 1 画素分ずらし（スキップし）、入力画像の画素データ $x(0)$ 、 $x(1)$ 、 $x(2)$ および $x(3)$ に基づいて内挿点となる出力画像の画素データ $y(2)$ を補間する動作を模式的に表しており、画素データ $x(0)$ 、 $x(1)$ 、 $x(2)$ および $x(3)$ に、それぞれ係数演算回路 2 0 で算出した補間係数を乗算することで出力画像の画素データ $y(2)$ を得る。

【 0 0 7 2 】

また、図 1 3 においては、画素データの選択範囲を右に 1 画素分スキップし、入力画像の画素データ $x(1)$ 、 $x(2)$ 、 $x(3)$ および $x(4)$ に基づいて内挿点となる出力画像の画素データ $y(3)$ を補間する動作を模式的に表しており、画素データ $x(1)$ 、
 $x(2)$ 、 $x(3)$ および $x(4)$ に、それぞれ係数演算回路 2 0 で算出した補間係数を乗算することで出力画像の画素データ $y(3)$ を得る。

20

【 0 0 7 3 】

なお、出力画像の $y(4)$ の計算に際しては画素データの選択範囲はスキップせずに、補間係数を変えることで対応する。

【 0 0 7 4 】

このように、補間対象となる入力画像の画素データの選択範囲を、画像の左端から所定画素数ずつ右にスキップさせたり、あるいはスキップさせずに、係数演算回路 2 0 で算出した補間係数を乗算することで出力画像データを得る。

【 0 0 7 5 】

このような入力画像の画素データの選択範囲のスキップ数については、以下に示す数式 (1 1) により決定することができる。

30

【 0 0 7 6 】

【 数 1 1 】

$$\text{Skip} = \lfloor b_{j+1} \rfloor - \lfloor b_j \rfloor \dots (11)$$

【 0 0 7 7 】

上記数式 (1 1) はフロア関数を表しており、隣り合う画素の座標をそれぞれフロア関数で表し、その差によりスキップ数が決定されることを表している。フロア関数は数値を整数化する関数であり、例えば「 1 . 5 」をフロア関数で処理すると「 1 」、「 2 . 5 」ならば「 2 」となる。

40

【 0 0 7 8 】

以上説明した方法により、予め設定された画素数分の入力画像データに基づいて出力画像データを得ることができる。

【 0 0 7 9 】

< 補間係数の算出方法 >

補間係数の算出には種々の方法が考えられるが、例えば、キュービックコンボリューションを用いた内挿法を説明する数式 (4) を、整数の乗算と加減算とデータのシフトのみで実現できるように下記の数式 (1 2) のように変形することができる。

50

【 0 0 8 0 】

【 数 1 2 】

$$\left\{ \begin{array}{l} h(t_i)' = \text{round}(h(t_i) * 2^n) \\ h(t_{i+1})' = \text{round}(h(t_{i+1}) * 2^n) \\ h(t_{i+2})' = \text{round}(h(t_{i+2}) * 2^n) \\ h(t_{i+3})' = 2^n - (h(t_i)' + h(t_{i+1})' + h(t_{i+2})') \end{array} \right. \quad \dots (12)$$

10

【 0 0 8 1 】

数式(12)においては、補間関数 $h(t_i)$ に 2^n (n は係数の精度を表し、正の整数) を乗算して round 演算した値を $h(t_i)'$ とし、補間関数 $h(t_{i+1})$ に 2^n を乗算して round 演算した値を $h(t_{i+1})'$ とし、補間関数 $h(t_{i+2})$ に 2^n を乗算して round 演算した値を $h(t_{i+2})'$ とし、 2^n から $h(t_i)'$ 、 $h(t_{i+1})'$ および $h(t_{i+2})'$ を差し引いた値を $h(t_{i+3})'$ としている。このため、上記数式(12)を図7に示した係数演算回路20に実装する場合、距離 t_{i+3} に対応する補間係数については減算により算出することができる。

20

【 0 0 8 2 】

図14は、係数演算回路20の構成を示すブロック図である。図14に示すように、係数演算回路20は、制御装置7に接続された座標演算回路201、座標演算回路201に接続された係数演算回路202、203および204、係数演算回路202~204の出力を受ける係数演算回路205を有している。なお、係数演算回路202~204の出力は、それぞれセクタ3にも与えられる。

【 0 0 8 3 】

係数演算回路205は、係数演算回路202~204の出力を加算する加算器2051と、精度設定部2050で設定された 2^n から係数演算回路202~204の出力の加算値を減算する減算器2052とを有し、減算器2052の出力がセクタ3に与えられる構成となっている。なお、精度設定部2050では、補間係数の精度に合わせて n の値を設定する部位であり、補間係数の精度が10ビットであれば $n = 10$ となり、 $2^{10} = 1024$ となる。

30

【 0 0 8 4 】

係数演算回路20は、制御装置7からの制御信号により演算を開始し、座標演算回路201では、数式(7)に基づいて内挿点の座標 b_j を算出する。

【 0 0 8 5 】

ここで、数式(12)を係数演算回路20に実装した場合の座標演算回路201の構成について図15を用いて説明する。図15に示すように、座標演算回路201は、制御装置7に接続される加算器2011、加算器2011の出力を受けるフリップフロップ2012、フリップフロップ2012の出力を分岐して出力するセクタ2013を有している。セクタ2013は、フリップフロップ2012の出力を小数演算回路2014、整数演算回路2015および加算器2011に与え、小数演算回路2014では、演算結果の小数部分を切り捨てる演算を行い、その結果を t_{i+1} として出力する。なお、この距離 t_{i+1} は、他の距離を算出するために用いられるので、便宜的に基準距離と呼称する場合もある。

40

【 0 0 8 6 】

また、小数演算回路2014の出力は加算器2018に与えられ、加算器2018において数値1.0との加算を行って、その結果を t_i として出力する。

50

【 0 0 8 7 】

また、小数演算回路 2 0 1 4 の出力は減算器 2 0 1 9 に与えられ、減算器 2 0 1 9 において数値 1 . 0 との減算を行って、その結果を t_{i+2} として出力する。

【 0 0 8 8 】

座標演算回路 2 0 1 では、制御装置 7 から Pitch が D / U として与えられ、これに先に得られた内挿点の座標を加算することで新たな内挿点の座標を得るが、最初の座標の算出に際しては、初期値 b_0 がフリップフロップ 2 0 1 2 に与えられ、D / U と加算されて座標 b_j となる。

【 0 0 8 9 】

その後の内挿点の座標の算出に際しては、フリップフロップ 2 0 1 2 に格納された現在の座標 b_j と D / U との加算を加算器 2 0 1 1 で行って次の座標 b_{j+1} を得る。

10

【 0 0 9 0 】

また、整数演算回路 2 0 1 5 では、フリップフロップ 2 0 1 2 の出力をフロア関数を用いて整数化する演算を行い、その結果を、加算器 2 0 1 1 の出力をフロア関数を用いて整数化する整数演算回路 2 0 1 6 での演算結果から減算器 2 0 1 7 を用いて減算することで、スキップ数を算出し、制御装置 7 に与える。これは、数式 (1 1) を用いて説明した演算に該当する。

【 0 0 9 1 】

内挿点の座標 b_j が得られると、 t_i 、 t_{i+1} および t_{i+2} が求まるので、係数演算回路 2 0 2 ~ 2 0 4 では数式 (4) から補間関数 $h(t_i)$ 、 $h(t_{i+1})$ および $h(t_{i+2})$ を求め、数式 (1 2) に基づいて、それぞれ補間係数 $h(t_i)'$ 、 $h(t_{i+1})'$ および $h(t_{i+2})'$ を算出する。また、係数演算回路 2 0 5 では、 2^n から補間係数 $h(t_i)'$ 、 $h(t_{i+1})'$ および $h(t_{i+2})'$ を差し引いた値を補間係数 $h(t_{i+3})'$ として算出する。

20

【 0 0 9 2 】

これらの補間係数はセレクタ 3 を介して乗算器 8 に与えられ、入力画像の画素データに乗算される。

【 0 0 9 3 】

なお、上述したように、座標演算回路 2 0 1 では、数式 (1 1) を用いて、入力画像の画素データの選択範囲をスキップさせるスキップ数も算出して制御装置 7 に与え、制御装置 7 では、当該スキップ数に基づいて、次の演算対象となる入力画像の画素データを決定する。

30

【 0 0 9 4 】

ここで、係数演算回路 2 0 2 ~ 2 0 4 における補間係数の算出について図 1 6 および図 1 7 を用いて説明する。

【 0 0 9 5 】

図 1 6 は、補間係数 $h(t_i)'$ および $h(t_{i+3})'$ を算出する係数演算系統図であり、併せて係数演算回路 2 0 2 の構成を示す図である。

【 0 0 9 6 】

図 1 6 において、「 t 」は座標演算回路 2 0 1 から出力された距離 t_i または t_{i+3} の値であり、簡単化のため「 t 」としている。なお、先に説明したように数式 (1 2) を実装する場合は、距離 t_{i+3} の補間係数については座標演算回路 2 0 5 で算出するので、係数演算回路 2 0 2 では補間係数 $h(t_i)'$ を算出することになる。

40

【 0 0 9 7 】

図 1 6 に示すように、係数演算回路 2 0 2 は、座標演算回路 2 0 1 から出力された t の値どうしを、乗算器 M L 1 において乗算することで t^2 を算出し、乗算器 M L 1 が出力する t^2 の値と座標演算回路 2 0 1 から出力された t の値とを乗算器 M L 2 において乗算することで t^3 を算出する。

【 0 0 9 8 】

その後、 t 、 t^2 および t^3 に対して、それぞれ乗算器 M L 3、M L 4 および M L 5 に

50

において、定数 $8a$ 、 $-5a$ および a を乗算し、それぞれの演算結果と $-4a$ とを加算器 $AD1$ で加算することで関数 $h(t)$ を得る。そして、当該関数 $h(t)$ の演算結果を $round$ 演算回路 $RD1$ で、例えば四捨五入することで補間係数 $h(t)'$ を得る。なお、この場合、補間係数 $h(t)'$ は補間係数 $h(t_i)'$ である。

【0099】

図17は、補間係数 $h(t_{i+1})'$ および $h(t_{i+2})'$ を算出する係数演算系統図であり、併せて係数演算回路203および204の構成を示す図である。

【0100】

図17において、「 t 」は座標演算回路201から出力された t_{i+1} または t_{i+2} の値であり、簡単化のため「 t 」としている。

10

【0101】

図17に示すように、係数演算回路203(204も同じ)は、座標演算回路201から出力された t の値どうしを、乗算器 $ML11$ において乗算することで t^2 を算出し、乗算器 $ML11$ が出力する t^2 の値と座標演算回路201から出力された t の値とを乗算器 $ML12$ において乗算することで t^3 を算出する。

【0102】

その後、 t^2 および t^3 に対して、それぞれ乗算器 $ML13$ および $ML14$ において、定数 $a+2$ および $-(a+3)$ を乗算し、それぞれの演算結果と1とを加算器 $AD11$ で加算することで関数 $h(t)$ を得る。そして、当該関数 $h(t)$ の演算結果を $round$ 演算回路 $RD11$ で、例えば四捨五入することで補間係数 $h(t)'$ を得る。なお、この場合、補間係数 $h(t)'$ は補間係数 $h(t_{i+1})'$ または $h(t_{i+2})'$ である。

20

【0103】

<変形例>

なお、上記の説明では、数式(12)に示されるように、 $h(t_i)'$ 、 $h(t_{i+1})'$ および $h(t_{i+2})'$ を個々に求めた後、 2^n から $h(t_i)'$ 、 $h(t_{i+1})'$ および $h(t_{i+2})'$ を差し引いた値を $h(t_{i+3})'$ としていたが、これに限定されるものではない。

【0104】

例えば、以下に示す数式(13)のように、 $h(t_{i+1})'$ 、 $h(t_{i+2})'$ 、 $h(t_{i+3})'$ を個々に求めた後、 2^n から $h(t_{i+1})'$ 、 $h(t_{i+2})'$ 、 $h(t_{i+3})'$ を差し引くことで $h(t_i)'$ を求めても良い。

30

【0105】

【数13】

$$\begin{cases} h(t_i)' = 2^n - (h(t_{i+1})' + h(t_{i+2})' + h(t_{i+3})') \\ h(t_{i+1})' = \text{round}(h(t_{i+1}) * 2^n) \\ h(t_{i+2})' = \text{round}(h(t_{i+2}) * 2^n) \\ h(t_{i+3})' = \text{round}(h(t_{i+3}) * 2^n) \end{cases} \quad \dots (13)$$

40

【0106】

ここで、数式(13)を係数演算回路20に実装した場合の座標演算回路201の構成について図18を用いて説明する。図18に示すように、座標演算回路201は、制御装置7に接続される加算器2011、加算器2011の出力を受けるフリップフロップ2012、フリップフロップ2012の出力を分岐して出力するセクタ2013を有してい

50

る。セレクタ 2013 は、フリップフロップ 2012 の出力を小数演算回路 2014、整数演算回路 2015 および加算器 2011 に与え、小数演算回路 2014 では、演算結果の小数部分を切り捨てる演算を行い、その結果を t_{i+1} として出力する。

【0107】

また、小数演算回路 2014 の出力は減算器 2020 に与えられ、減算器 2020 において数値 1.0 との減算を行って、その結果を t_{i+2} として出力する。

【0108】

また、小数演算回路 2014 の出力は減算器 2021 に与えられ、減算器 2021 において数値 2.0 との減算を行って、その結果を t_{i+3} として出力する。

【0109】

この場合、係数演算回路 202 ~ 204 のそれぞれに、距離 t_{i+1} 、 t_{i+2} 、 t_{i+3} を与えることになるので、係数演算回路 202 および 203 では図 17 に示した係数演算を行う構成を採り、係数演算回路 204 では図 16 に示した係数演算を行う構成を採ることとなる。

【0110】

なお、その他、図 15 に示した座標演算回路 201 と同一の構成については同一の符号を付し、重複する説明は省略する。

【0111】

また、以下に示す数式 (14) のように、 $h(t_i)'$ 、 $h(t_{i+2})'$ 、 $h(t_{i+3})'$ を個々に求めた後、 2^n から $h(t_i)'$ 、 $h(t_{i+2})'$ 、 $h(t_{i+3})'$ を差し引くことで $h(t_{i+1})'$ を求めても良い。

【0112】

【数 14】

$$\begin{cases} h(t_i)' = \text{round}(h(t_i) * 2^n) \\ h(t_{i+1})' = 2^n - (h(t_i)' + h(t_{i+2})' + h(t_{i+3})') \\ h(t_{i+2})' = \text{round}(h(t_{i+2}) * 2^n) \\ h(t_{i+3})' = \text{round}(h(t_{i+3}) * 2^n) \end{cases} \quad \dots (14)$$

【0113】

ここで、数式 (14) を係数演算回路 20 に実装した場合の座標演算回路 201 の構成について図 19 を用いて説明する。図 19 に示すように、座標演算回路 201 は、制御装置 7 に接続される加算器 2011、加算器 2011 の出力を受けるフリップフロップ 2012、フリップフロップ 2012 の出力を分岐して出力するセレクタ 2013 を有している。セレクタ 2013 は、フリップフロップ 2012 の出力を小数演算回路 2014、整数演算回路 2015 および加算器 2011 に与え、小数演算回路 2014 では、演算結果の小数部分を切り捨てる演算を行い、その結果を t_{i+1} として出力する。

【0114】

また、小数演算回路 2014 の出力は加算器 2022 に与えられ、加算器 2022 において数値 1.0 との加算を行って、その結果を t_i として出力する。

【0115】

また、小数演算回路 2014 の出力は減算器 2023 に与えられ、減算器 2023 において数値 1.0 との減算を行って、その結果を t_{i+2} として出力する。

【0116】

また、また、小数演算回路 2014 の出力は減算器 2024 に与えられ、減算器 202

10

20

30

40

50

4において数値2.0との減算を行って、その結果を t_{i+3} として出力する。

【0117】

この場合、係数演算回路202~204のそれぞれに、距離 t_i 、 t_{i+2} 、 t_{i+3} を与えることになるので、係数演算回路202および204では図16に示した係数演算を行う構成を採り、係数演算回路203では図17に示した係数演算を行う構成を採ることとなる。

【0118】

なお、その他、図15に示した座標演算回路201と同一の構成については同一の符号を付し、重複する説明は省略する。

【0119】

また、以下に示す数式(15)のように、 $h(t_i)'$ 、 $h(t_{i+1})'$ 、 $h(t_{i+3})'$ を個々に求めた後、 2^n から $h(t_i)'$ 、 $h(t_{i+1})'$ 、 $h(t_{i+3})'$ を差し引くことで $h(t_{i+2})'$ を求めても良い。

【0120】

【数15】

$$\begin{cases} h(t_i)' = \text{round}(h(t_i) * 2^n) \\ h(t_{i+1})' = \text{round}(h(t_{i+1}) * 2^n) \\ h(t_{i+2})' = 2^n - (h(t_i)' + h(t_{i+1})' + h(t_{i+3})') \\ h(t_{i+3})' = \text{round}(h(t_{i+3}) * 2^n) \end{cases} \dots (15)$$

【0121】

ここで、数式(15)を係数演算回路20に実装した場合の座標演算回路201の構成について図20を用いて説明する。図20に示すように、座標演算回路201は、制御装置7に接続される加算器2011、加算器2011の出力を受けるフリップフロップ2012、フリップフロップ2012の出力を分岐して出力するセレクタ2013を有している。セレクタ2013は、フリップフロップ2012の出力を小数演算回路2014、整数演算回路2015および加算器2011に与え、小数演算回路2014では、演算結果の小数部分を切り捨てる演算を行い、その結果を t_{i+1} として出力する。

【0122】

また、小数演算回路2014の出力は加算器2025に与えられ、加算器2025において数値1.0との加算を行って、その結果を t_i として出力する。

【0123】

また、小数演算回路2014の出力は減算器2026に与えられ、減算器2026において数値2.0との減算を行って、その結果を t_{i+3} として出力する。

【0124】

この場合、係数演算回路202~204のそれぞれに、距離 t_i 、 t_{i+1} 、 t_{i+3} を与えることになるので、係数演算回路202および204では図16に示した係数演算を行う構成を採り、係数演算回路203では図17に示した係数演算を行う構成を採ることとなる。

【0125】

なお、その他、図15に示した座標演算回路201と同一の構成については同一の符号を付し、重複する説明は省略する。

【0126】

以上の説明では、画像を拡大する場合を例に採って説明したが、画像を縮小する場合も

10

20

30

40

50

同様に本発明を適用できることは言うまでもない。

【0127】

また、以上の説明では、水平方向についての画像の拡大縮小についてのみ述べたが、垂直方向についても本発明を適用することで画像の拡大縮小を行うことができることは言うまでもない。その場合、水平方向と垂直方向とで別々に発明を適用すれば良い。

【0128】

<プロセッサでの実現>

以上の説明においては画像の拡大縮小処理をハードウェアを用いて実現する構成を示したが、CPU (Central Processing Unit) やDSP (Digital Signal Processor) などのプロセッサ上で動作するソフトウェアにより実現することも可能である。

10

【0129】

CPUは一般的に図21に示すような構成を有している。すなわち、命令をメモリから読み出す命令フェッチャ (Instruction fetcher) 101、命令フェッチャ101が読み出した命令を処理する命令デコーダ (Instruction decoder) 102、CPU内のメモリとして動作するレジスタ (Registers) 103、レジスタ103に格納されたデータに基づいて四則演算や論理演算などの算術的な処理を行うALU (Arithmetic and Logic Unit) 104および図示されない外部のメモリとの間でのデータの授受を行うメモリアンターフェース105を有している。

【0130】

このような構成を有するプロセッサ上で本発明に係る画像の拡大縮小処理を実現する場合には、入力画像データに対して画素を補間することでU/D倍 (U、Dは自然数) に変換して出力画像データとするために、ALU104を使用して、予め拡大縮小処理で使用する補間係数を算出する係数演算を行い、算出した補間係数をレジスタ103または図示されないメモリに保存しておく。

20

【0131】

そして、入力画像データを構成する入力画素のそれぞれに対応するように、レジスタ103または図示されないメモリに保存された補間係数を乗算するステップ、得られた乗算済みデータの加算を繰り返し、入力画素の所定個数分について乗算済みデータの総和を取るステップ、所定個数分の入力画素について乗算済みデータの総和が得られるタイミングで乗算済みデータの総和を出力するステップ、得られた乗算済みデータの総和をシフト演算することで、出力画像データのビット数を入力画像データのビット数に調整するステップをALU104で実行することで画像の拡大縮小処理を実現することができる。

30

【0132】

なお、本発明は、その発明の範囲内において、各実施の形態を自由に組み合わせたり、各実施の形態を適宜、変形、省略することが可能である。

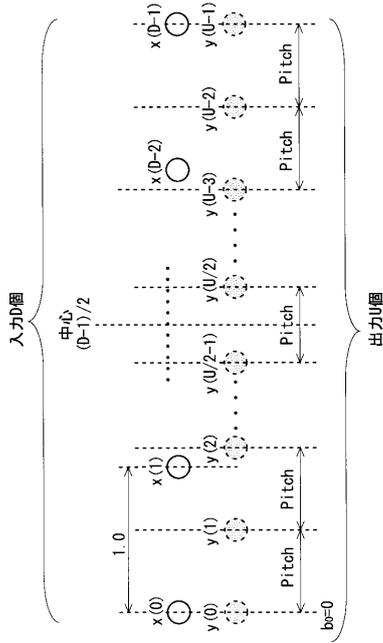
【符号の説明】

【0133】

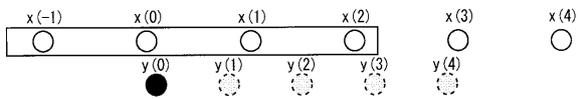
- 4 セレクタ
- 8 乗算器
- 9 加算器
- 14 シフト回路
- 20 係数演算回路
- 201 座標演算回路
- 202 ~ 205 係数演算回路

40

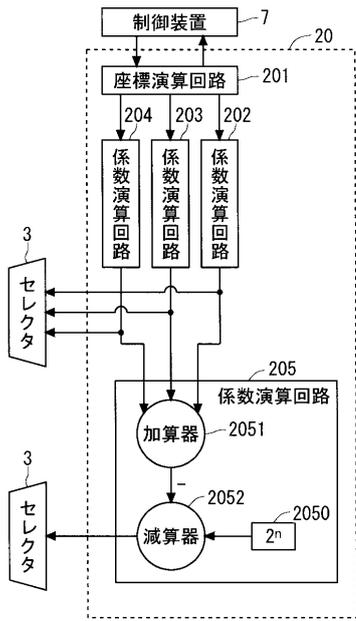
【図9】



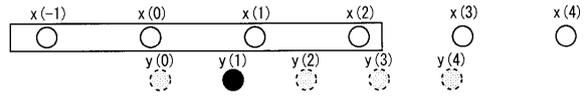
【図10】



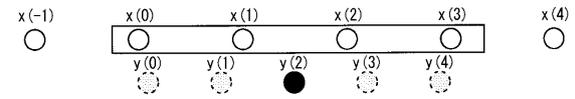
【図14】



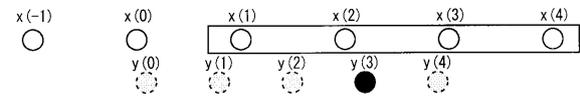
【図11】



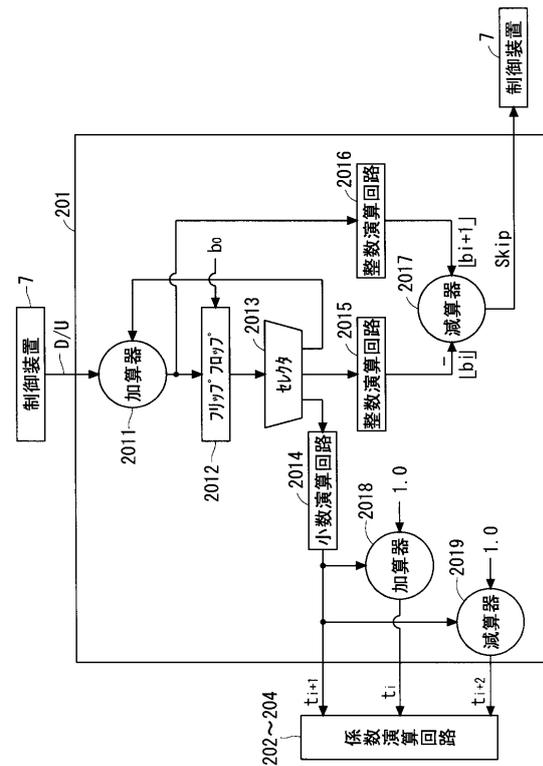
【図12】



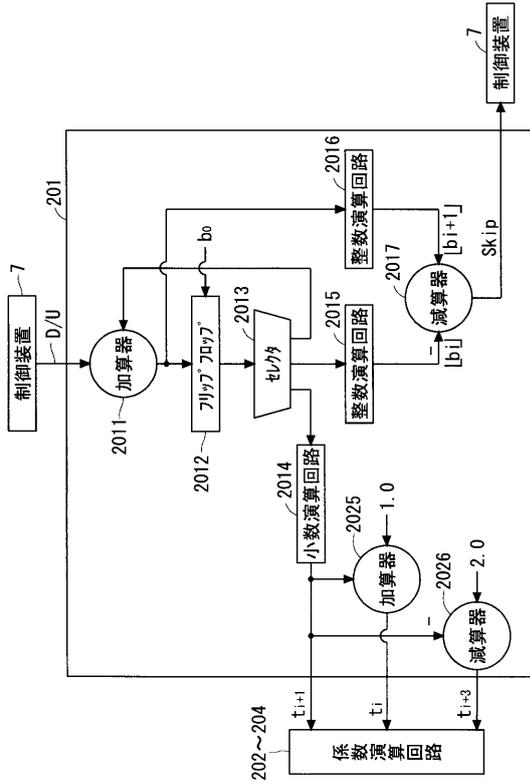
【図13】



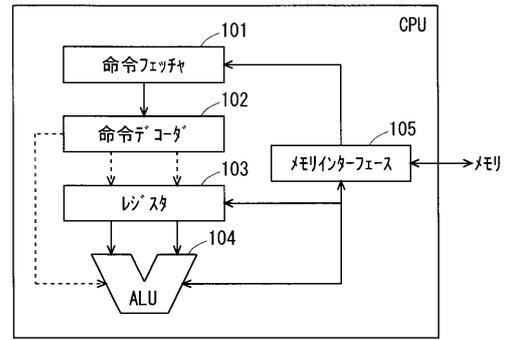
【図15】



【図 20】



【図 21】



フロントページの続き

- (56)参考文献 特開2011-9814(JP,A)
特開2015-12415(JP,A)
特開平11-308574(JP,A)
特開平10-133618(JP,A)

- (58)調査した分野(Int.Cl., DB名)
H04N 1/393
G06T 3/40