(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6448410号
(P6448410)

(45)発行日　平成31年1月9日(2019.1.9)　(24)登録日　平成30年12月14日(2018.12.14)

(51)Int.Cl.			F I		
H04N	1/60	(2006.01)	H04N	1/60	160
G06T	1/00	(2006.01)	G06T	1/00	510
B41J	5/30	(2006.01)	B41J	5/30	C

請求項の数 17　(全 17 頁)

(21)出願番号　特願2015-37292(P2015-37292)
(22)出願日　平成27年2月26日(2015.2.26)
(65)公開番号　特開2016-163070(P2016-163070A)
(43)公開日　平成28年9月5日(2016.9.5)
審査請求日　平成30年2月7日(2018.2.7)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】データ変換装置およびその制御方法、プログラム

(57)【特許請求の範囲】
【請求項１】
各軸に２^{n}＋１個の格子点を有するｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換するデータ変換装置であって、
前記ルックアップテーブルは２^{m}個のメモリを有し、各座標の格子点データがその座標の各軸の座標値の奇数、偶数の組み合わせで特定されるメモリに格納されており、
前記入力データに基づいて、各軸の座標値がｎ＋１ビットで表された、前記ルックアップテーブルの格子点座標を、補間処理に用いるためｍ＋１個生成する生成手段と、
前記格子点座標の各軸の座標値の最下位ビットを連結して得られるｍビットのデータに基づいて、前記２^{m}個のメモリからアクセス先のメモリを決定する決定手段と、
前記格子点座標を、（２$^{n-1}$＋１）^{m}個以下の連続するアドレスのいずれか一つに変換して得られたアドレスで、前記アクセス先のメモリにアクセスするアクセス手段と、を備えることを特徴とするデータ変換装置。
【請求項２】
前記アクセス手段は、前記格子点座標の各軸のｎ＋１ビットの座標値から前記最下位ビットを除いた各軸がｎビットの座標値で構成される座標を、前記アクセス先のメモリに格納される格子点データの個数の連続するアドレスのいずれか一つに変換することを特徴とする請求項１に記載のデータ変換装置。
【請求項３】
前記２^{m}個のメモリの各々に格納される前記格子点データの個数は、前記格子点座標を

構成する座標値のうちの奇数の座標値の数をｐ（ｐは０以上ｍ以下の整数）とした場合に、（２$^{ｎ－１}$＋１）^{ｐ}×（２$^{ｎ－１}$）$^{ｍ－ｐ}$であることを特徴とする請求項２に記載のデータ変換装置。

【請求項４】

前記生成手段は、前記入力データを構成する要素値の各々の上位ｎビットに基づいてｍ次元の座標から１つの格子点座標を生成し、前記入力データを構成する要素値の各々から上位ｎビットを除いたデータに基づいて、補間処理に用いるための複数の格子点座標を生成することを特徴とする請求項１乃至３のいずれか１項に記載のデータ変換装置。

【請求項５】

前記ルックアップテーブルは３次元（ｍ＝３）であり、且つ、各軸に１７点（ｎ＝４）の格子点を持ち、

前記生成手段は、前記入力データを構成する要素値の各々の上位ｎビットに基づいて３次元の空間の１つの格子点座標を生成し、前記入力データを構成する要素値の各々から上位ｎビットを除いたデータに基づいて、前記１つの格子点座標を含む四面体補間処理に用いるためにさらに３つの格子点座標を生成することを特徴とする請求項４に記載のデータ変換装置。

【請求項６】

前記アクセス手段により前記ルックアップテーブルをアクセスして読み出された、前記生成手段で生成された格子点座標の格子点データを用いて、前記補間処理を実行する補間手段をさらに備えることを特徴とする請求項４または５に記載のデータ変換装置。

【請求項７】

前記補間手段は、前記入力データの各要素値の各々から前記上位ｎビットを除いた残りのビットにより表される値を用いて前記補間処理を行うことを特徴とする請求項６に記載のデータ変換装置。

【請求項８】

前記２^{ｍ}個のメモリは、（２$^{ｎ－１}$＋１）^{ｐ}×（２$^{ｎ－１}$）$^{ｍ－ｐ}$のｍ＋１種類の容量のいずれかの容量のメモリで構成されることを特徴とする請求項１乃至７のいずれか１項に記載のデータ変換装置。

【請求項９】

前記アクセス手段が生成するアドレスのビット幅は、ｍ×（ｎ－１）＋１またはｍ×（ｎ－１）であることを特徴とする請求項１乃至８のいずれか１項に記載のデータ変換装置。

【請求項１０】

前記アクセス手段は、最上位ビットの方向へビットをシフトするシフタと加算器を含み、前記格子点の座標の各軸の座標値の前記最下位ビットを除いたｎビットの座標値で構成される座標を入力して前記アドレスを出力する計算回路を、前記２^{ｍ}個のメモリの各々に対して有することを特徴とする請求項１乃至９のいずれか１項に記載のデータ変換装置。

【請求項１１】

前記ルックアップテーブルは３次元であり（ｍ＝３）、

前記アクセス手段は、最上位ビットの方向へビットをシフトするシフタと加算器を含み、前記格子点座標の各軸の座標値の前記最下位ビットを除いたｎビットの座標値で構成される座標を入力して前記アドレスを出力する複数の計算回路を有し、

前記座標の第１の軸の座標値を除く第２、第３の軸の座標値の奇数、偶数が一致する座標に対応したメモリについて共通の計算回路を用いることを特徴とする請求項１乃至９のいずれか１項に記載のデータ変換装置。

【請求項１２】

前記アクセス手段は、各軸に２^{ｎ}個の格子点を有するｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換することを指示する選択信号に応じて、前記格子点座標の各軸の座標値の前記最下位ビットを除いたｎビットで構成される座標を、（２$^{ｎ－１}$）^{ｍ}個の連続するアドレスのいずれかに変換して得られたアドレスで、前

記アクセス先のメモリにアクセスすることを特徴とする請求項１に記載のデータ変換装置。

【請求項１３】

前記生成手段は、前記選択信号に応じて、前記入力データを構成する要素値が取り得る最大値を2^n-1で等分した値で、前記入力データのそれぞれの要素値を除算して得られた商に基づいてｍ次元の格子点座標を生成し、前記除算により得られた余りに基づいて補間処理に用いるためのｍ＋１個の格子点座標を生成することを特徴とする請求項１２に記載のデータ変換装置。

【請求項１４】

ｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換するデータ変換装置であって、前記ルックアップテーブルでは、ｍ次元の座標により特定される格子点データが、座標を形成する各軸の座標値の奇数、偶数の組み合わせによって特定される、複数のメモリのうちの一つに格納されており、

前記入力データに基づいて、前記ルックアップテーブルの各軸の座標値を含む格子点座標を生成する生成手段と、

前記格子点座標の各軸の座標値に基づいて前記複数のメモリのうちのアクセス先のメモリを決定する決定手段と、

前記複数のメモリのそれぞれへ格納される格子点データの個数のうち最大の個数の連続したアドレスのうちの一つへ前記格子点座標を変換して得られたアドレスで、前記アクセス先のメモリをアクセスするアクセス手段と、を備えることを特徴とするデータ変換装置。

【請求項１５】

各軸に2^n+1個の格子点を有するｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換するデータ変換装置の制御方法であって、

前記ルックアップテーブルは2^m個のメモリを有し、各座標の格子点データがその座標の各軸の座標値の奇数、偶数の組み合わせで特定されるメモリに格納されており、

前記入力データに基づいて、各軸の座標値がｎ＋１ビットで表された、前記ルックアップテーブルの格子点座標を、補間処理に用いるためｍ＋１個生成する生成工程と、

前記格子点座標の各軸の座標値の最下位ビットを連結して得られるｍビットのデータに基づいて、前記2^m個のメモリからアクセス先のメモリを決定する決定工程と、

前記格子点座標を、$(2^{n-1}+1)^m$個以下の連続するアドレスのいずれか一つに変換して得られたアドレスで、前記アクセス先のメモリにアクセスするアクセス工程と、を有することを特徴とするデータ変換装置の制御方法。

【請求項１６】

ｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換するデータ変換装置であって、前記ルックアップテーブルでは、ｍ次元の座標により特定される格子点データが、座標を形成する各軸の座標値の奇数、偶数の組み合わせによって特定される、複数のメモリのうちの一つに格納されており、

前記入力データに基づいて、前記ルックアップテーブルの各軸の座標値を含む格子点座標を生成する生成工程と、

前記格子点座標の各軸の座標値に基づいて前記複数のメモリのうちのアクセス先のメモリを決定する決定工程と、

前記複数のメモリのそれぞれへ格納される格子点データの個数のうち最大の個数の連続したアドレスのうちの一つへ前記格子点座標を変換して得られたアドレスで、前記アクセス先のメモリをアクセスするアクセス工程と、を有することを特徴とするデータ変換装置の制御方法。

【請求項１７】

請求項１５または１６に記載されたデータ変換装置の制御方法の各工程をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】
【０００１】
本発明は、ルックアップテーブルを用いたデータ変換装置およびその制御方法に関する。
【背景技術】
【０００２】
近年、画像の入力機器としてスキャナやデジタルカメラ、デジタルビデオカメラなどが普及している。一方、出力機器としてはインクジェットや電子写真などの方式を用いた各種カラープリンタが普及している。これらのカラー入出力機器はそれぞれ固有の色空間を有している。例えばスキャナで得たカラー画像をそのまま別のカラープリンタに転送して印刷した場合、印刷されたカラー画像の色がスキャナによって読みとられたオリジナルのカラー画像の色と一致することはほとんどない。このようなカラー画像等に対するデバイス間での色再現性の問題を解決するためには、入力デバイスの色空間を出力デバイスの色空間に変換する処理（色空間変換と呼ぶ）が必要となる。このような色空間変換は入出力機器の色再現性を高めるために入出力機器に搭載されている。
【０００３】
この色空間変換とは、具体的には入力ガンマ補正、輝度濃度変換、マスキング、黒生成、出力ガンマ補正等の一連の画像処理全体、もしくはその中の一部の処理を指す。一般的には、入力デバイスの３色（例えばレッド、ブルー、グリーンの３色。ＲＧＢとも略せる）のデジタル画像信号を同時に参照して、出力デバイスの３色（例えばシアン、マゼンタ、イエローの３色。ＣＭＹとも略せる）、あるいは４色（例えばシアン、マゼンタ、イエロー、ブラックの４色。ＣＭＹＫとも略せる）のデジタル画像信号に変換することが多い。
【０００４】
色空間変換の実現方法として、予め変換結果をルックアップテーブル（以下、ＬＵＴと呼ぶ）としてメモリに記憶しておき、入力された画像信号に対してそのＬＵＴを参照して変換結果を出力する方法がある。このＬＵＴを用いた色空間変換方法では、ＬＵＴのメモリを削減するため補間演算を併用するのが一般的である。即ち、被変換色の代表点に対応する変換色のデータをメモリに記憶させ、メモリ内に記憶された数点の変換色データを元に補間演算を行い、新たな変換色データを算出することが行われている。ＲＧＢなどの軸上の代表点の取り方は細かいほど変換精度は高くなるが、必要なメモリの容量が増える。逆に代表点の取り方が粗いと変換精度は低くなるが、必要なメモリの容量は減る。軸上の代表点の数をグリッド数または格子点数と呼び、８から３３までのグリッド数（格子点数）がよく使用されている。
【０００５】
格子点上の代表点を参照する個数は補間方法によって異なる。例えば、立方体補間の場合は８点、四面体補間の場合は４点の代表点を参照して補間演算を行う。代表点の変換色データはメモリに格納されているので、複数の代表点の変換色データを参照するためにはメモリへの複数回のアクセスが発生するため、逐次的なアクセスを行っていては処理性能を上げることが出来ない。そこで、メモリを複数に分割し複数回のメモリアクセスを同時に行うことで、処理性能を向上させる方法が提案されている。
【０００６】
特許文献１では立方体補間を高速に実行するため、３２格子点の変換色データを８個のメモリに分けて記憶し、並列に読み出すことで高速化を行っている。また、特許文献２には、２のべき乗では表せられない格子点数の変換色データをメモリに分割して記憶する方法が示されている。一例として３３格子点の変換色データを８個のメモリに分割し、そのアドレスを１７ビットで生成する方法について記載している。特許文献１，２のいずれにおいても、各軸の座標値の偶数／奇数が同一となる代表点が同一のメモリに格納される。たとえば、被変換色がＲＧＢの３色の場合、ＲＧＢの各軸の座標値の偶数、奇数の組み合わせである（偶数、偶数、偶数）、（偶数、偶数、奇数）、・・・、（奇数、奇数、奇数

）で、それぞれ別の８個のメモリに分割して格納することで高速処理を可能にしている。
【先行技術文献】
【特許文献】
【０００７】
【特許文献１】米国特許第４８３７７２２号明細書
【特許文献２】特開２００１－３６７５５号公報
【発明の概要】
【発明が解決しようとする課題】
【０００８】
特許文献１では３２格子点での処理を行うために合計３２７６８（＝３２＊３２＊３２）個のデータを格納できるメモリが必要である。また、特許文献２では３３格子点での処理を行うために３５９３７（＝３３＊３３＊３３）個のデータを格納できるメモリが必要である。互いに必要なメモリ容量が異なるため、複数のグリッド数に対応するためには、最も大きい容量に合わせてメモリ分割を行う必要がある。また、一軸当たりの格子点数が奇数の場合に、上述のように８個のメモリにデータを分けると、各メモリに格納するデータ容量は各軸の座標値の偶数、奇数の組合せで異なってくる。偶数に属する格子点の数と奇数に属する格子点の数が異なるためである。従って、ＬＵＴを設定するために代表点の変換色データ（以下、格子点データと記す）をメモリに格納する際、メモリ毎に格子点データ数を意識して格納する必要があった。あるいは、このような制御をなくすために、最も容量の大きなメモリにあわせて不要なデータを追加して格納する必要があった。
【０００９】
一方、格子点データの参照時のＬＵＴのアクセスの際は、各軸の格子点座標のＬＳＢを連結したアドレスがＲＡＭを選択するアドレス（以下、バンクアドレスと記す）になり、残りの上位ビットを連結したものがＲＡＭへの実アドレスになる。上記３３グリッドの場合、各軸には３３個の格子点があるため、座標は各軸６ビットで表現され、ＬＳＢはバンクアドレスとして使用されるので、実アドレスは（６－１）＋（６－１）＋（６－１）＝１５ビットとなる。つまり、アドレス空間としては2^{15}＝３２７６８の容量を持つことになる。しかしながら、各軸の奇数番目の格子点のみあるいは偶数番目の格子点のみが参照されるので、各軸について参照される格子点の数は最大で１７個（３３／２＋１）となる。すなわち、各メモリが実際に必要な容量は最大１７×１７×１７＝４９１３個の格子点データを格納できる容量であり、そのため、メモリアドレス上、飛び飛びのアドレスに格子点データが格納されることになってしまう。
【００１０】
本発明は上記課題に鑑みてなされたものであり、ＬＵＴを複数のメモリに分割して格納する際の、メモリアドレス生成に関して好適な構成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】
【００１１】
上記の目的を達成するための本発明の一態様によるデータ変換装置は以下の構成を備える。すなわち、
各軸に2^{n}＋１個の格子点を有するｍ次元のルックアップテーブルを参照して、ｍ個の要素値を有する入力データを変換するデータ変換装置であって、
前記ルックアップテーブルは2^{m}個のメモリを有し、各座標の格子点データがその座標の各軸の座標値の奇数、偶数の組み合わせで特定されるメモリに格納されており、
前記入力データに基づいて、各軸の座標値がｎ＋１ビットで表された、前記ルックアップテーブルの格子点座標を、補間処理に用いるためｍ＋１個生成する生成手段と、
前記格子点座標の各軸の座標値の最下位ビットを連結して得られるｍビットのデータに基づいて、前記2^{m}個のメモリからアクセス先のメモリを決定する決定手段と、
前記格子点座標を、（2^{n-1}＋１）^{m}個以下の連続するアドレスのいずれか一つに変換して得られたアドレスで、前記アクセス先のメモリにアクセスするアクセス手段と、を備える。

【発明の効果】
【００１２】
本発明によれば、ＬＵＴを複数のメモリに分割して格納する際のメモリアドレス生成において、無駄なアドレスの生成が低減される。
【図面の簡単な説明】
【００１３】
【図１Ａ】第１実施形態による
【図１Ｂ】第１実施形態におけるデータ変換部の構成例を示すブロック図。
【図２】座標生成部の構成を示すブロック図。
【図３】単位立方体を６つの四面体に分割する例を示す図。
【図４】下位データの大小比較に応じた順序データと係数を表す図。
【図５】第１実施形態におけるアドレス生成部の回路構成例を示すブロック図。
【図６】第１実施形態におけるアドレス生成部の回路を４つに統合した図。
【図７】第２実施形態におけるデータ変換部の構成例を示すブロック図。
【図８】第２実施形態におけるアドレス生成部の回路構成例を示すブロック図。
【図９】分割したメモリと対応するワード数とアドレスのビット幅を示す図。
【発明を実施するための形態】
【００１４】
以下、添付の図面を参照して、本発明を好適な実施形態に基づいて詳細に説明する。なお、以下の実施形態において示す構成は一例にすぎず、本発明は図示された構成に限定されるものではない。
【００１５】
＜第１実施形態＞
図１Ａは、第１実施形態による画像形成装置１０の概略の構成例を示すブロック図である。格納部１２に格納されている画像データは、ＲＧＢの３つの要素値を有する入力データである。画像入力部１３は、格納部１２から格納されている画像データを読み出し、画像処理部１４へ供給する。画像処理部１４は、ＲＧＢ空間の画像データに周知の変換処理を施し、ＣＭＹＫ空間の画像データへ変換する。画像処理部１４により得られたＣＭＹＫ空間の画像データは印刷部１５へ供給され、印刷される。画像処理部１４は、ルックアップテーブルを参照して、３つの要素値を有する入力データを４つの要素値からなるデータへ変換するデータ変換部１００を有する。なお、画像処理部１４は、ＲＯＭ２２に格納されたプログラムを実行して、画像形成装置１０の全体的な制御を実現するＣＰＵ２１を有している。以下に説明するデータ変換部１００の構成や機能の一部またはすべては、ＣＰＵ２１がＲＯＭ２２に格納されたプログラムを実行することにより実現されてもよいし、専用のハードウエアで実現されてもよい。
【００１６】
図１Ｂは、第１実施形態によるデータ変換処理を実行するデータ変換部１００の構成例を示すブロック図である。データ変換部１００では、ｍ個の軸の各々に$2^{n}+1$個の格子点を有するｍ次元のルックアップテーブル（以下、ＬＵＴ）を参照して、ｍ個の要素値を有する入力データを変換する制御を行う。本実施形態のデータ変換部１００は、ＬＵＴの参照によりＲＧＢデータをＣＭＹＫデータへ変換するための３次元（ｍ＝３）のＬＵＴを用いた例を説明する。より詳しくは、データ変換部１００が、各色１０ビットのＲＧＢの画素データ（ｍ＝３）を入力し、１７格子点（ｎ＝４）の３次元のＬＵＴを用いた四面体補間処理による色変換を行い、各色１０ビットのＣＭＹＫの画素データとして出力する例を示す。まず、入力されたＲＧＢの各データは、基準点決定部１０１へ入力され、そこでＬＵＴの各軸の格子点を特定する上位ｎビット（本例では上位４ビット）と、それ以外の下位ビットに分けられる。下位ビットは領域判定部１０２へ入力される。領域判定部１０２は、補間に用いる四面体の領域判定を行い、四面体を特定するのに用いられる順序データ（order[0]～order[2]）と、補間処理のための補間係数（係数０～３）を出力する。他方、上位ｎビットは座標生成部１０３へ入力される。座標生成部１０３は、基準点決定部

１０１からの上位ビットと、領域判定部１０２からの順序データとを参照して、四面体補間処理に用いる四面体の頂点を特定する格子点の座標を生成する。

【００１７】

座標生成部１０３で生成された格子点座標はアドレス生成部１０４へ入力される。アドレス生成部１０４は、入力された格子点座標から、アクセス先のメモリを指定するバンクデータと、そのアクセス先のメモリへアクセスするためのアドレスを生成し、出力する。生成されたアドレスはＬＵＴ部１０５へ入力され、バンクデータはセレクタ部１０６へ入力される。ＬＵＴ部１０５では、入力されたアドレスにしたがってＬＵＴから読み出された値が出力される。ＬＵＴ部１０５は2^m個、本例では$2^3=8$個のメモリを有し、生成されたアドレスに従って各メモリから読み出された格子点データ（各色１０ビットのＣＭＹＫデータ）を出力する。セレクタ部１０６はＬＵＴ部１０５から出力された格子点データとアドレス生成部１０４からのバンクデータにより、補間処理に必要な４つのデータを選択し、選択データとして出力する。補間部１０７では選択データと係数データとの積和演算による補間処理を行い、新たなＣＭＹＫの値を算出して出力する。以上が本実施形態のデータ変換部１００による、四面体補間処理を含むデータ変換処理の大まかな流れである。次に、各構成要素の詳細な説明をしていく。

【００１８】

基準点決定部１０１は、入力されたＲＧＢの画素データ（ＲＧＢ各１０ビット）を、メモリから補間演算に必要な値（ＬＵＴの格子点の値）をリードするためのアドレス生成に必要な上位データと、補間演算時の係数として使用される下位データに分割する。上位データと下位データへ分ける方法は、公知の方法を用いれば良い。たとえば、一軸あたりの格子点数が2^n+1個の場合、入力データの各軸を2^n個の区間に分け、そのうちの一つを特定するための上位ｎビットが上位データとして出力される。一例を挙げると、１軸当たり１７格子点（ｎ＝４）の場合、入力データの各軸を１６区間に分けた時に入力データが属する区間を示す部分、すなわち上位４ビットが上位データとして出力される。これは各色の入力データ（１０ビット）のうちの上位４ビットを抽出して出力することで実現され得る。一方、下位データは入力データの１０ビットのうちの下位６ビットとなるが、それらはＲＧＢ成分で正規化されてから出力される。

【００１９】

領域判定部１０２は、入力された下位データの示す座標が、選択された単位立方体を６つの四面体領域に分割したうちのどの四面体に属するかを決定する。単位立方体とは、上述の上位４ビットにより特定される格子点（基準点）を（ｘ，ｙ，ｚ）とした場合に、（ｘ，ｙ，ｚ）～（ｘ＋１，ｙ＋１，ｚ＋１）の８個の格子点により特定される立方体である。図３（ａ）～（ｆ）に、単位立方体を分割する６つの四面体領域を示す。一番左下の出発点を基準点３０１と呼び、この点を基準として入力された下位データのうち値の大きい軸の方向を選んで進み、最終的に対角点３０４に到達する経路によって四面体が一つだけ決まる。基準点３０１からＲＧＢいずれかの軸で距離が１だけ進んだ点を第１中間点３０２、さらに別の軸で距離が１つ進んだ点を第２中間点３０３、最後の点を対角点３０４と呼ぶ。基準点３０１と対角点３０４は６つの四面体で共通である。

【００２０】

領域判定部１０２は、まず、入力されたＲＧＢの各々の下位データの大きさを比較して、大きさの順序を示す順序データ0rder[0]～[2]を決定する。0rder[0]は最も大きな値を持つ色を表し、0rder[2]は最も小さな値を持つ色を表す。例えば、Ｒ成分の下位データをｒ、Ｇ成分の下位データをｇ、Ｂ成分の下位データをｂとし、ｒ＞ｂ＞ｇとすると0rder[0]＝0、0rder[1]＝2、0rder[2]＝1となる（０～２はそれぞれｒ、ｇ、ｂを示す）。下位データの大小比較のためにはｒ－ｇの値と、ｇ－ｂの値と、ｂ－ｒの値の３つの値の正負を調べれば良く、その結果に応じて得られる0rder[0]～[2]を表にしたものを図４に示す。また、領域判定部１０２は同時に、補間のための係数０から係数３も算出する。係数０～３の算出方法は、図４に示したとおりである。なお、図４の係数０の算出部分でＷと書いているのは、正規化の１にあたるものであり、ｒ＋ｇ＋ｂ＝Ｗを満たす。

【００２１】
座標生成部１０３は、各色の上位４ビットのデータにしたがって、補間処理に必要な基準点、第１中間点、第２中間点、対角点の４点の座標を求める。図２に座標生成部１０３の内部構成を示す。まず、入力された上位ｎビット（４ビット）のデータ（上位データ）はビット拡張部２０１で５ビットのデータへ拡張される。これは、対角点の座標値が最大で（2^n+1，2^n+1，2^n+1）、すなわち（１７，１７，１７）となるためである。３つのビット拡張部２０１により得られた５ビットのデータは、ビット連結器２０２、１軸加算器２０３、２軸加算器２０４、３軸加算器２０５に同時に入力される。また、１軸加算器２０３には領域判定部１０２から0rder[0]が、２軸加算器２０４には領域判定部１０２から0rder[0]と0rder[1]が入力される。

【００２２】
ビット連結器２０２は、３つのビット拡張部２０１から入力された５ビットのデータを連結し、基準点の座標（１５ビット）として出力する。１軸加算器２０３は、0rder[0]が示す色と対応した軸の座標値を基準点から１つ増加させて第１中間点の座標（１５ビット）を得る。例えば、0rder[0]が０であればＲ軸の座標値（ビット拡張部２０１－１からの５ビットデータ）を１つ増加させる。２軸加算器２０４は、0rder[0]と0rder[1]の値に対応する２つの軸の座標値を基準点から１つずつ増加させて第２中間点の座標を得る。例えば、0rder[0]が０で、0rder[1]が２であれば、Ｒ軸とＢ軸の座標値を１だけ増加させて第２中間点の座標とする。３軸加算器２０５は、入力された全ての上位データにおいて、全ての軸の値を１ずつ加算して対角点の座標を得る。これにより補間に必要な４点の座標を求めることが出来る。また、図２が示すように、ビット連結器２０２、１軸加算器２０３、２軸加算器２０４、３軸加算器２０５は並列に動作することが可能であり、４つの格子点座標値を同時に得ることができる。上記のように１軸当たり１７格子点の場合、各軸について０～１６の値を取り得るので、１軸あたり５ビットが必要となり、格子点座標としては３軸分の１５ビットの値が出力される。より一般化すると、１軸当たりの格子点数が2^n+1の場合、格子点座標を表すために１軸当たりｎ＋１ビットが必要となる。

【００２３】
アドレス生成部１０４は、座標生成部１０３から入力された４つの格子点座標を元に、ＬＵＴ部１０５のメモリからデータを読み出すためのアドレスを生成する。まず、本実施形態の補間処理で使用される具体的なメモリ構成について説明する。１７格子点を有する３次元のＬＵＴを用いて格子点の補間処理を行う場合、全てのデータを格納するのに必要なメモリ容量は１７×１７×１７＝４９１３個の格子点データに対応した容量となる。このメモリをＭＥＭ０からＭＥＭ７までの８個のメモリに分割し、それぞれのメモリへ格納される格子点データを、各軸の座標値の偶数、奇数の並びで分類して格納する。ＭＥＭ０へは格子点座標が（偶数、偶数、偶数）に対応するデータを格納する。１７格子点の場合、偶数座標は０から１６の間に、０，２，４，６，８，１０，１２，１４，１６の９点が存在し、奇数座標は１，３，５，７，９，１１，１３，１５の８点が存在する。したがって、ＭＥＭ０へは９×９×９＝７２９個のデータが格納される。

【００２４】
また、ＭＥＭ１は格子点座標が（偶数、偶数、奇数）に対応するデータを格納し、必要なメモリの容量は９×９×８＝６４８個のデータに対応する容量となる。ＭＥＭ２は格子点座標が（偶数、奇数、偶数）に対するデータを格納し、必要なメモリの容量はＭＥＭ１と同じである。ＭＥＭ３は格子点座標が（偶数、奇数、奇数）に対応するデータを格納し、必要なメモリの容量は９×８×８＝５７６個のデータに相当する容量となる。以下も同様にメモリを用意していき、ＭＥＭ７では格子点座標（奇数、奇数、奇数）に対応するデータを格納し、必要なメモリの容量は８×８×８＝５１２個のデータに対応する容量となる。分割したメモリ毎にワード数をまとめたものを図９に示した。また、アドレスの指定に必要なビット幅はＭＥＭ０からＭＥＭ６までが１０ビットで、ＭＥＭ７だけは９ビットとすることが出来る。

【００２５】

より一般化すると、格子点数が2^n+1の３次元のＬＵＴを用いた補間処理に必要なデータ量は$(2^n+1)^3$である。このデータを８個のメモリへ座標の偶奇で分割すると、
［ａ］　$(2^{n-1}+1)\times(2^{n-1}+1)\times(2^{n-1}+1)$個の格子点データを格納するメモリが１つ、
［ｂ］　$(2^{n-1}+1)\times(2^{n-1}+1)\times(2^{n-1})$個の格子点データを格納するメモリが３つ、
［ｃ］　$(2^{n-1}+1)\times(2^{n-1})\times(2^{n-1})$個の格子点データを格納するメモリが３つ、
［ｄ］　$(2^{n-1})\times(2^{n-1})\times(2^{n-1})$個の格子点データを格納するメモリが１つ、
という構成になる。［ａ］～［ｃ］のタイプの７つのメモリのアクセスに必要なアドレスのビット幅は３（ｎ－１）＋１ビットで、［ｄ］のタイプの最も小さいメモリのアクセスに必要なアドレスのビット幅は３（ｎ－１）ビットとすることが出来る。ｍ次元のＬＵＴの場合、アドレスのビット幅はｍ×（ｎ－１）＋１またはｍ×（ｎ－１）となる。

【００２６】

アドレス生成部１０４は、このように分割されたメモリに対するアドレスとして、まず格子点に対応するデータをどのメモリから読み出すかを示すバンクアドレスを生成する。バンクアドレスは、ｍ次元の各軸の座標値の最下位ビット（ＬＳＢ）を連結してｍビットのバンクデータを生成する。本実施形態では、（Ｒ［０］，Ｇ［０］，Ｂ［０］）で表される３ビットの値の連結によりバンクアドレスが得られる。そして、得られたバンクアドレスに応じて下記の８つの計算式で各メモリに対するアドレスを算出する。なお、下式において、［］内は抽出ビット位置を示し、［４…１］は１ビット目から４ビット目の４ビットを抽出したビット列であることを示す。

【００２７】

ＭＥＭ０：
Ｒ［４…１］×９×９＋Ｇ［４…１］×９＋Ｂ［４…１］　・・・式（１－１）
ＭＥＭ１：
Ｒ［４…１］×９×８＋Ｇ［４…１］×８＋Ｂ［３…１］　・・・式（１－２）
ＭＥＭ２：
Ｒ［４…１］×８×９＋Ｇ［３…１］×９＋Ｂ［４…１］　・・・式（１－３）
ＭＥＭ３：
Ｒ［４…１］×８×８＋Ｇ［３…１］×８＋Ｂ［３…１］　・・・式（１－４）
ＭＥＭ４：
Ｒ［３…１］×９×９＋Ｇ［４…１］×９＋Ｂ［４…１］　・・・式（１－５）
ＭＥＭ５：
Ｒ［３…１］×９×８＋Ｇ［４…１］×８＋Ｂ［３…１］　・・・式（１－６）
ＭＥＭ６：
Ｒ［３…１］×８×９＋Ｇ［３…１］×９＋Ｂ［４…１］　・・・式（１－７）
ＭＥＭ７：
Ｒ［３…１］×８×８＋Ｇ［３…１］×８＋Ｂ［３…１］　・・・式（１－８）。

【００２８】

上式で示されているように、入力データに４ビットのデータと３ビットのデータがある。格子点座標が偶数の場合取り得る値は０から１６の偶数であり、そこからＬＳＢを除いた値は０から８までの値となり、それらは４ビットで表現され得る。一方、奇数の場合は、取り得る値は１から１５の奇数であり、ＬＳＢを除いた値は０から７までの値となるから３ビットあれば足りるためである。また、この計算式により、ＭＥＭ７のアドレスだけはビット幅が９ビットとなり、他は１０ビットとなることも分かる。

【００２９】

これらのアドレス計算式（１－１）から（１－８）をブロック図にしたものを図５に示す。図５は、最上位ビットの方向（左方向）へビットをシフトするシフタと加算器とを含

み、格子点の座標の各軸の座標値の最下位ビットを除いたｎビットまたはｎ－１ビットの座標値で構成される座標を入力してメモリに対応したアドレスを出力する計算回路を示す。図５でシフタ５０１，５０２，５０３，５０４は、入力された値を左方向へシフトをさせる要素部品であり、シフタの右上に書かれた量のシフトを行う。加算器５０５，５０６，５０７，５０８，５０９，５１０，５１１は、入力された複数の値を足し合わせてその結果を出力する要素部品である。

【００３０】

例えば、シフタ５０１－１は、１軸当たりの格子点数が2^n+1個の場合、２ｎ－２ビットの左シフトを行う。シフト量の部分にｎビットと表現されているが、１７グリッドの場合$17=2^4+1$なのでｎ＝４である。したがって、図５（ａ）の構成では、シフタ５０１－１がＲ［４…１］を６回左にシフトし、とシフタ５０２－１Ｒ［４…１］を４回左にシフトする。そして、加算器５０５－１がそれらのシフト結果とＲ［４…１］を加算するため、加算器５０５－１からはＲ［４…１］×９×９の結果が得られる。また、シフタ５０４－１は、Ｇ［４…１］をｎ－１＝３回左にシフトし、加算器５０７－１がそのシフト結果とＧ［４…１］を加算するため、加算器５０５－１からはＧ［４…１］×９の結果が得られる。そして、加算器５０８－１が加算器５０５－１の出力、加算器５０７－１の出力、およびＢ［４…１］を加算することにより、式（１－１）の演算結果が得られる。同様に、図５（ｂ）～図５（ｈ）のそれぞれの構成により、式（１－２）～式（１－８）の演算結果が得られる。

【００３１】

なお、格子点の座標の第１の軸（Ｒ軸）の座標値を除く第２、第３の軸（Ｇ，Ｂ軸）の座標値の奇数、偶数が一致する座標について共通の計算式を用いることができる。したがって、式（１－１）～式（１－８）は以下の４つのアドレス計算式にまとめられる。すなわち、

ＭＥＭ０またはＭＥＭ４：

Ｒ［４…１］×９×９＋Ｇ［４…１］×９＋Ｂ［４…１］　・・・式（２－１）

ＭＥＭ１またはＭＥＭ５：

Ｒ［４…１］×９×８＋Ｇ［４…１］×８＋Ｂ［４…１］　・・・式（２－２）

ＭＥＭ２またはＭＥＭ６：

Ｒ［４…１］×８×９＋Ｇ［４…１］×９＋Ｂ［４…１］　・・・式（２－３）

ＭＥＭ３またはＭＥＭ７：

Ｒ［４…１］×８×８＋Ｇ［４…１］×８＋Ｂ［４…１］　・・・式（２－４）。

【００３２】

図６（ａ）～（ｄ）に式（２－１）から（２－４）の演算結果を得るための４つのアドレス生成部のブロック図を示す。図６でシフタ６０１，６０２，６０３，６０４は、入力された値を左方向へシフトをさせる要素部品であり、シフタの右上に書かれた量のシフトを行う。加算器６０５，６０６，６０７，６０８は、入力された複数の値を足し合わせてその結果を出力する要素部品である。色変換回路のスループットを１／２以下で動作させる場合は、図６（ａ）～（ｄ）の構成を用いることで回路規模の削減が可能である。一方、スループットが１などの大きな値で動作させる場合は、例えば、ＭＥＭ０とＭＥＭ４を同時に計算する可能性があるため、図６に示したそれぞれの計算回路を２つずつ、あるいは図５に示したような構成が必要となる。なお、図６（ｄ）のビット抽出器６０９は、加算器６０８から出力される１０ビットの加算結果のうちの下位９ビットを抽出して出力する。

【００３３】

以上のようなアドレス生成を、座標生成部１０３から入力される、基準点から対角点までの４点について行い、４つのバンクアドレスとアクセスアドレスを得る。ここで、４つのバンクアドレスとアクセスアドレスをＢａｎｋ［０］～［３］、Ａｄｄｒｅｓｓ［０］～［３］と表す。つまり、アドレス生成部１０４は、基準点に対応する格子点データを得るためにＢａｎｋ［０］で示されるメモリのＡｄｄｒｅｓｓ［０］へアクセスする。同様

に、アドレス生成部１０４は、第１中間点のデータの更新点データを得るために、Ｂａｎｋ［１］で示されるメモリのＡｄｄｒｅｓｓ［１］へアクセスする。また、アドレス生成部１０４は、第２中間点の格子点データを得るためにＢａｎｋ［２］で示されるメモリのＡｄｄｒｅｓｓ［２］へアクセスする。さらに、アドレス生成部１０４は、対角点の格子点データを得るために、Ｂａｎｋ［３］で示されるメモリのＡｄｄｒｅｓｓ［３］へアクセスする。例えば、Ｂａｎｋ［０］が３、Ａｄｄｒｅｓｓ［０］が１である場合、ＬＵＴにおける基準点の座標に対応する格子点データは、ＭＥＭ３のアドレス１に格納されており、そこからデータの読み出しが行われる。なお、Ｂａｎｋ［０］～［３］に含まれなかった残りの４つのメモリはリードしても使用されないデータであるため、それらのメモリへはアドレスは何の値を出力しても良いし、０を出力することにしても良い。

【００３４】

ＬＵＴ部１０５は、上述したＭＥＭ０からＭＥＭ７までの８個のメモリを内部に有し、アドレス生成部１０４で生成されたアドレスＡｄｄｒｅｓｓ［０］～［３］を入力として対応するメモリを参照し、格子点データを取得し出力する。セレクタ部１０６は、入力された８個のデータの中から、アドレス生成部１０４より入力されるＢａｎｋ［０］～［３］で示されるメモリから読み出されたデータを選択し、選択データとして出力する。ここで、ＬＵＴ部１０５の各メモリから供給される格子点データはＣＭＹＫ各色１０ビットの計４０ビットのデータである。セレクタ部１０６は、これらのデータのうちＢａｎｋ［０］～［３］で示される４つのメモリからの格子点データを選択し、それぞれからＣＭＹＫの色成分ごとのデータ（１０ビット）を抽出して、色成分に対応する補間部１０７へ出力する。以降で、Ｂａｎｋ［０］で示されるバンクから読み出されたデータのうちのシアン成分をＣ［０］、マゼンタの成分をＭ［０］、イエローの成分をＹ［０］、ブラックの成分をＢ［０］と記す。

【００３５】

補間部１０７は、領域判定部１０２より入力された係数データと、セレクタ部１０６より入力された選択データとを用いて下記のような積和演算を行い、結果を目的色データとして出力する。

出力データのＣ成分＝係数０×Ｃ［０］＋係数１×Ｃ［１］＋係数２×Ｃ［２］＋係数３×Ｃ［３］

出力データのＭ成分＝係数０×Ｍ［０］＋係数１×Ｍ［１］＋係数２×Ｍ［２］＋係数３×Ｍ［３］

出力データのＹ成分＝係数０×Ｙ［０］＋係数１×Ｙ［１］＋係数２×Ｙ［２］＋係数３×Ｙ［３］

出力データのＫ成分＝係数０×Ｋ［０］＋係数１×Ｋ［１］＋係数２×Ｋ［２］＋係数３×Ｋ［３］。

【００３６】

以上説明したように、第１実施形態によれば、各軸の格子点数が$2^{n}+1$個のｍ次元のＬＵＴから得られたデータを補間処理して目的とするデータを得るデータ変換部１００において、ｍ＋１種類のアドレス空間の2^{m}個のメモリが用いられる。ここで、ｍ＋１種類のメモリの容量は、座標を構成する座標値のうち奇数の座標値の数をｐ（ｐは０以上ｍ以下の整数）とした場合に、$(2^{n-1}+1)^{p}\times(2^{n-1})^{m-p}$となる。上述した実施形態では、ｍ＝３（３次元ＬＵＴ）であるため、以下の４種類のアドレス空間の８個のメモリが用いられる。

（ａ）　$(2^{n-1}+1)\times(2^{n-1}+1)\times(2^{n-1}+1)$　…１つ

（ｂ）　$(2^{n-1}+1)\times(2^{n-1}+1)\times(2^{n-1})$　…３つ

（ｃ）　$(2^{n-1}+1)\times(2^{n-1})\times(2^{n-1})$　…３つ

（ｄ）　$(2^{n-1})\times(2^{n-1})\times(2^{n-1})$　…１つ

【００３７】

そのため、（ａ）～（ｃ）のアドレス空間のメモリについてはアドレスのビット幅が３（ｎ－１）＋１となり、（ｄ）のアドレス空間のメモリについてはアドレスのビット幅が

３（ｎ－１）となる。一般に、一軸あたり２^{n}＋１個の格子点を有する３次元ＬＵＴでは、各格子点の座標値を表すために、各軸についてｎ＋１ビットのデータが必要である。ｎ＋１ビットの座標値の最下位ビットをバンク切替に用いる場合でも、ｎビットのアドレス幅、すなわち、２^{n}×２^{n}×２^{n}個のアドレスの空間を持つメモリが８個必要になる。たとえば、上記実施形態のようにｎ＝４とした場合、４０９６個のアドレスの空間となる。これに対して、本実施形態によれば、７２９個のアドレス空間のメモリが１つ、６４８個のアドレス空間のメモリが３つ、５７６個のアドレス空間のメモリが３つ、５１２個のアドレス空間のメモリが１つあれば済む。このように、上述した（ａ）～（ｄ）の４種類のアドレス空間のメモリを用いることができるので、メモリ容量を削減することができ、アドレスのビット幅の削減も可能となる。

【００３８】

なお、上記のようなメモリ容量を実現するために、式（１－１）～（１－８）では、座標生成部から入力した座標を、上述した（ａ）～（ｄ）で示されるアドレス数（格子点データの数）までの連続するアドレスのいずれかへ変換してアクセスアドレスとしている。たとえば、式（１－１）では、（ａ）のメモリ（ｎ＝４）へのアクセスに関して、７２９個のアドレスのいずれか、すなわち０～７２８の連続するアドレスのいずれかに対応するように入力された座標を変換している。これを一般化すると、本実施形態のアドレス生成部１０４は、各軸の座標値がｎビットで表される格子点の座標を、メモリに格納されている格子点データの数に対応した連続する値（アドレス）のいずれかに変換するものである。これにより、メモリには全ての格子点データを連続するアドレスに格納することができ、必要なメモリ容量を低減することができる。なお、２^{m}個のメモリの各々に格納されている格子点データの数は、奇数の座標値の数をｐ（ｐは０以上ｍ以下の整数）とした場合に、（２$^{n-1}$＋１）^{p}×（２$^{n-1}$）$^{m-p}$となる。

【００３９】

なお、上記では、ＬＵＴの格子点データを格納する複数のメモリのそれぞれに格納される格子点データの個数の連続したアドレスの一つへ格子点座標を対応付けている。複数のメモリのそれぞれに格納される格子点データの個数は、座標値の偶数、奇数の組み合わせにより変化し、全て同じものとはならず、アドレスを生成するための計算式も異なってくる。これに対して、ＬＵＴの格子点データを格納する複数のメモリのうち格納される格子点データの個数の最大数（上述のように各軸に２^{n}＋１個の格子点が配置される３次元ＬＵＴの場合は、（２$^{n-1}$＋１）^{3}となる）を全てのメモリに適用してもよい。このようにすれば、複数のメモリの全てにおいて、（２$^{n-1}$＋１）^{3}個のアドレスの一つへ格子点座標を変換する構成を用いることができ、アドレスの算出に同じ計算式を用いることができる。上述のように、３次元のＬＵＴで各軸の格子点数が１７個の場合、各メモリに確保される格子点データのアドレスは７２９個となり、各メモリについて、格子点座標は７２９個の連続するアドレスの一つへ変換されることになる。すなわち、本実施形態では、各軸の格子点数が２^{n}＋１であるｍ次元のＬＵＴの場合、（２$^{n-1}$＋１）^{m}個以下の連続するアドレスの一つへ格子点座標が変換されることで、メモリへのアクセスアドレスのビット幅、各メモリのデータ容量の低減を実現している。

【００４０】

また、上記ではＬＵＴの各軸の格子点数が同一の場合を説明したが、これに限られるものでない。各軸の格子点数が異なっていても、複数のメモリのうちのアクセス先のメモリへ格納される格子点データの個数は一意に決まるので、その個数分の連続したアドレスのうちの一つへ格子点座標を変換すればよい。あるいは、複数のメモリのそれぞれへ格納される格子点データの個数のうち最大の個数の連続したアドレスのうちの一つへ格子点座標を変換するよう構成を採用できることも明らかである。

【００４１】

＜第２実施形態＞

第１実施形態では、１７グリッドでの処理時にメモリを４種類の大きさで８個に分割することで、アドレス生成式が４つにまとめられ、ビット幅の短いアドレスを生成できるこ

とを示した。第２実施形態では、同じメモリ構成を持ちながら、アドレス生成部１０４を変えることで１６グリッドの処理と１７グリッドの処理を両方行うことのできる例を示す。図７に第２実施形態に係るブロック図を示す。図１と異なる点は１６グリッド処理か１７グリッド処理かを指示する選択信号としての１ビットの信号“17GridMode”が入力として加えられる。アドレス生成部１００４は、“17GridMode”が１に設定されていれば１７グリッドで処理し、０であれば１６グリッドで処理する。17GridModeは基準点決定部１００１とアドレス生成部１００４に入力され、それらの動作が第１実施形態と異なる。従って、第２実施形態では基準点決定部１００１とアドレス生成部１００４について詳細に説明する。

【００４２】

基準点決定部１００１は、17GridModeの設定値を受けてその設定値に従って、上位データと下位データを算出して出力する。１７グリッド時の処理では第１実施形態と同様の処理が行われる。１６グリッド時の処理には公知の方法を用いることができる。１６グリッド時の処理の一例を挙げると、入力最大値を１５で等分した値で入力データを除算して得られた商を上位データとして座標生成部１０３へ出力し、該除算による余りの値をＲＧＢ各成分で正規化した値を下位データとして領域判定部１０２へ出力する。より一般化して表現すると、2^n個のグリッドの場合、入力最大値を2^n-1で等分した値で入力データを除算して得られた商を上位データとし、余りの値をＲＧＢ各成分で正規化した値を下位データとする。

【００４３】

アドレス生成部１００４は、座標生成部１０３に入力された４つの格子点座標を元に、ＬＵＴ部１０５のメモリからデータを読み出すためのアドレスを生成する。バンクアドレスの算出方法は第１実施形態と同じで、格子点座標のＬＳＢをつなげて生成する。アドレス計算式は、１７グリッド処理の場合、第１実施形態と同じものとなる。１６グリッド処理の場合、アドレス計算式はＭＥＭ０からＭＥＭ７までのメモリについて１種類の式で表すことができ、以下の式で算出できる。

Ｒ［３…１］×８×８＋Ｇ［３…１］×８＋Ｂ［３…１］・・・式（３）

【００４４】

この式により生成されるアドレスは９ビットであり、最大値は５１１である。１６グリッド処理時のメモリのデータの格納方法は１７グリッド時の同じように、格子点座標の偶奇で分類して格納する。このようにすると０から１５までの整数の中で偶数の個数と奇数の個数はいずれも８個で一致するため、１７グリッドの時と異なり８個のメモリにはデータが５１２個ずつ均等に分割される。例えば、ＭＥＭ０の格納可能なワード数は７２９であるが、そのうちアドレスの若い方から順に５１２個分だけ格納されて補間処理に使用される。ＭＥＭ１からＭＥＭ６までの５１２を越える容量を持つメモリに関しても同様である。メモリのアドレスを表す信号線はＭＥＭ０からＭＥＭ６までは１０ビットで構成されているが、１６グリッドで使用する時は最上位ビットとして０を追加しておけば、アドレス線も共通に使用することが出来る。

【００４５】

１６グリッドでの処理時に用いる式（３）と１７グリッドでの処理時に用いる式（２－１）から式（２－４）までの両方を実現する回路として図８のような構成が挙げられる。図６と同じ動作をする素子については同じ記号を与えた。図８において、マスク８０１－１から８０１－６までで表されているものは、設定されている17GridModeの値によって出力を変える機能を持つ。設定されている17GridModeが１の時は入力データをそのまま出力し、０の時は入力データに関わらず０を出力する。この信号17GridModeを切り替えることで、図８の回路の動作を切り替えることが可能となる。すなわち、17GridModeが１の場合には図６の回路と同様に動作し、17GridModeが０の場合には、（ａ）～（ｄ）の全ての計算回路が上記式（３）の計算を実現する回路となる。

【００４６】

以上、説明したように、１６グリッドと１７グリッドを切り替える信号17GridModeを加

え、その信号に応じてアドレス生成方法を切り替える処理を加えることで、１６グリッドと１７グリッドの間で処理を容易に切り替えることが出来る。また、上記各実施形態によれば、奇数グリッド数の色変換処理においても無駄なアドレスを使用しないので、メモリの容量を最小化できる。また、アドレス生成時に所定ビットをマスクすることにより、偶数グリッド数の色変換処理にも対応可能となる。

【００４７】

＜その他の実施形態＞

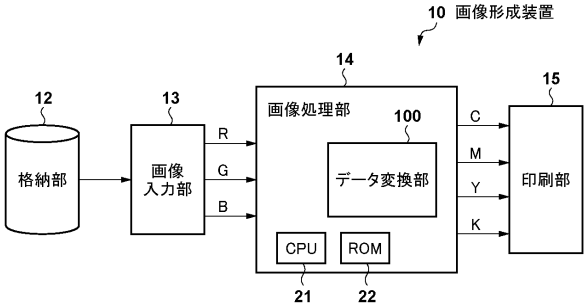
本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【符号の説明】

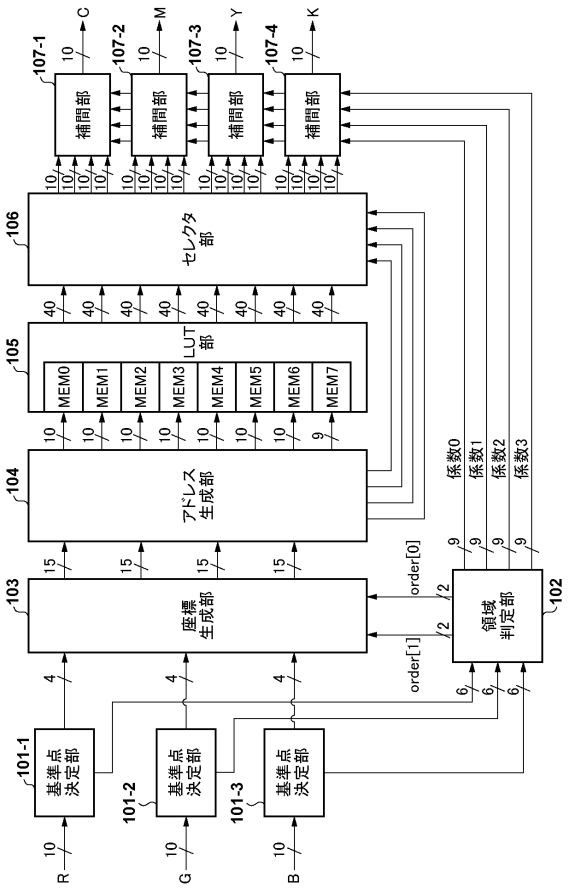
【００４８】

１００：データ変換部、点決定部、１０２：領域判定部、１０３：座標生成部、１０４：アドレス生成部、１０５：ＬＵＴ部、１０６：セレクタ部、１０７：補間部、２０１：ビット拡張部、２０２：ビット連結器、２０３：１軸加算器、２０３：２軸加算器、２０４：３軸加算器

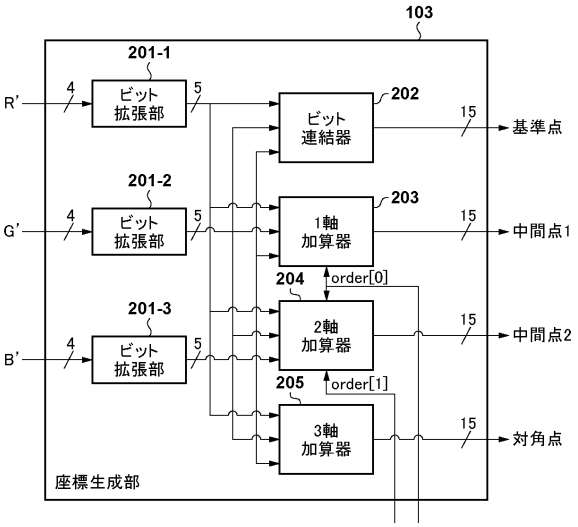
【図１Ａ】



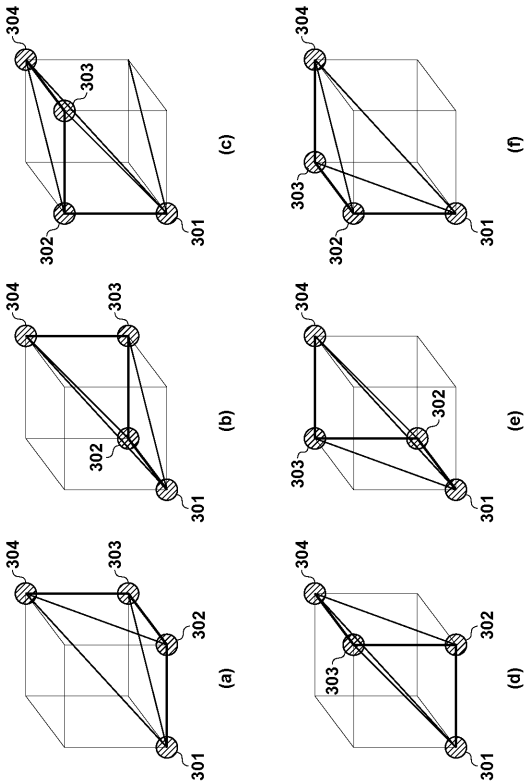
【図１Ｂ】



【図２】



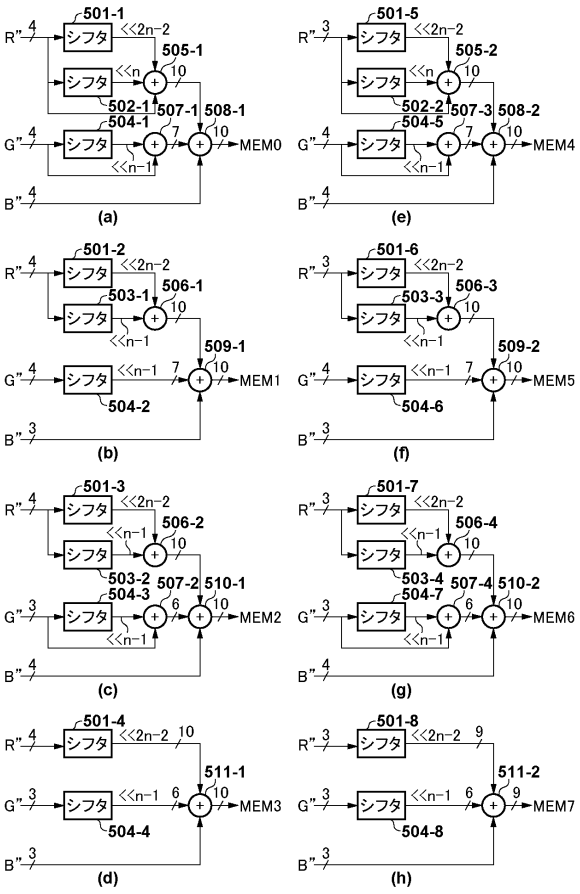
【図３】



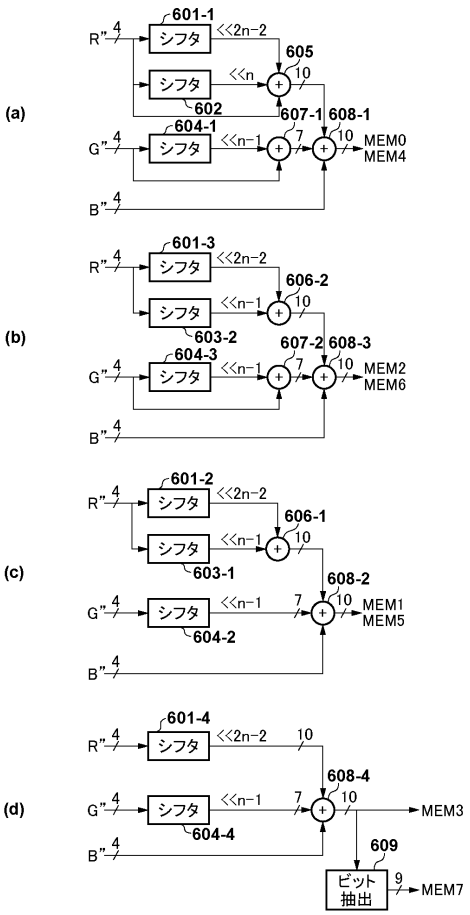
【図４】

r−g	g−b	b−r	order[0]	order[1]	order[2]	係数0	係数1	係数2	係数3
+	+	*	0	1	2	W−r	r−g	g−b	b
*	−	−	0	2	1	W−r	−(b−r)	−(g−b)	g
+	*	+	2	0	1	W−b	b−r	r−g	g
−	−	*	2	1	0	W−b	−(g−b)	−(r−g)	r
*	+	+	1	2	0	W−g	g−b	b−r	r
−	*	−	1	0	2	W−g	−(r−g)	−(b−r)	b

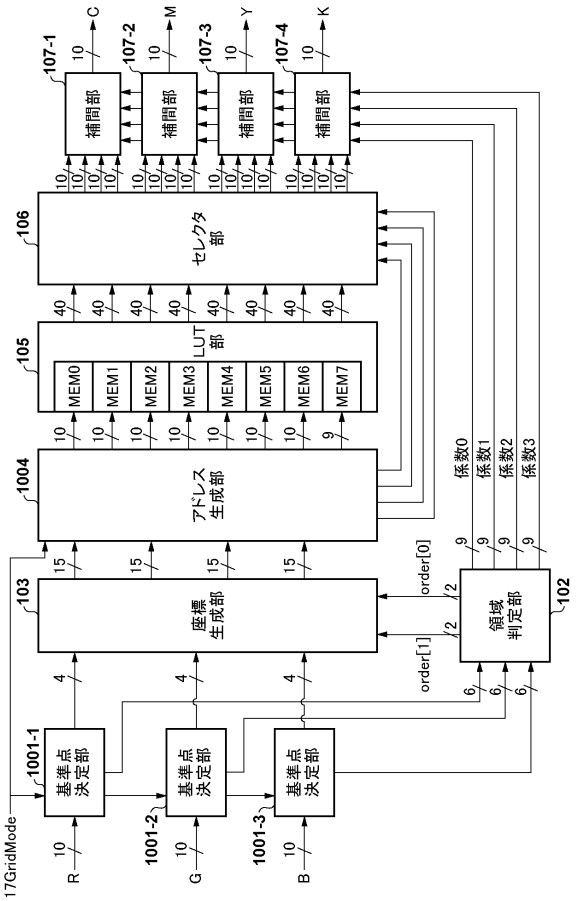
【図５】



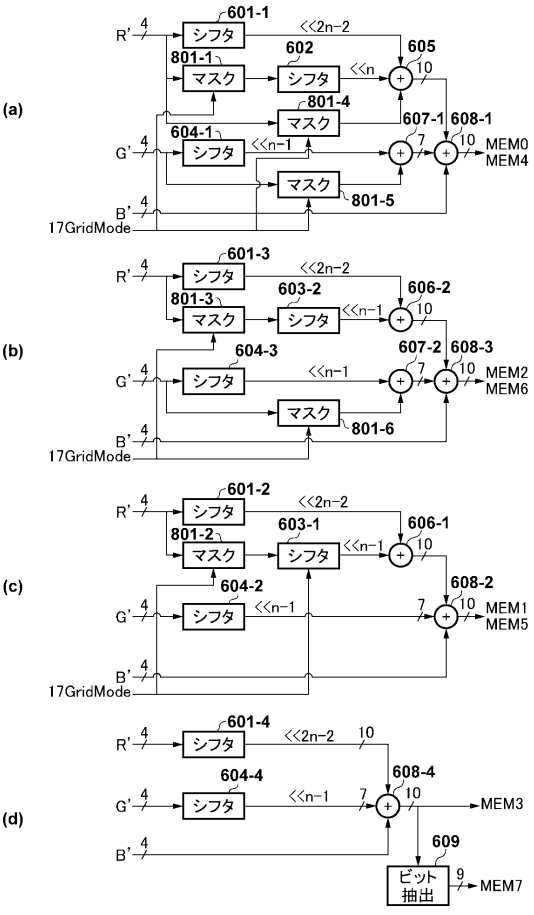
【図６】



【図７】



【図８】



【図９】

メモリ	座標のLSB			各軸の格子点数			word数	アドレスbit数
	R0	G0	B0	R	G	B		
MEM0	0	0	0	9	9	9	729	10
MEM1	0	0	1	9	9	8	648	10
MEM2	0	1	0	9	8	9	648	10
MEM3	0	1	1	9	8	8	576	10
MEM4	1	0	0	8	9	9	648	10
MEM5	1	0	1	8	9	8	576	10
MEM6	1	1	0	8	8	9	576	10
MEM7	1	1	1	8	8	8	512	9

フロントページの続き

(72)発明者　寶▲崎▼　健太
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内
(72)発明者　石川　尚
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

審査官　豊田　好一

(56)参考文献　特開２０１０－１１４５４２（ＪＰ，Ａ）
特開２００１－０３６７５５（ＪＰ，Ａ）
特開２０１０－１９３４７４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｎ　　１／４６－６２
Ｂ４１Ｊ　　５／３０
Ｇ０６Ｔ　　１／００