

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-227231

(P2004-227231A)

(43) 公開日 平成16年8月12日(2004.8.12)

(51) Int. Cl.⁷

G06T 1/00
H04N 1/387

F I

G06T 1/00 500B
G06T 1/00 510
H04N 1/387

テーマコード(参考)

5B057
5C076

審査請求 未請求 請求項の数 18 O L (全 31 頁)

(21) 出願番号 特願2003-13500 (P2003-13500)
(22) 出願日 平成15年1月22日(2003.1.22)

(特許庁注: 以下のものは登録商標)
Windows
Macintosh

(71) 出願人 000004226
日本電信電話株式会社
東京都千代田区大手町二丁目3番1号
(74) 代理人 100083552
弁理士 秋田 収喜
(74) 代理人 100103746
弁理士 近野 恵一
(72) 発明者 能登 肇
東京都千代田区大手町二丁目3番1号 日
本電信電話株式会社内
(72) 発明者 橋本 秋彦
東京都千代田区大手町二丁目3番1号 日
本電信電話株式会社内

最終頁に続く

(54) 【発明の名称】 画像データへの立体情報埋め込み方法、装置、プログラム、および記録媒体

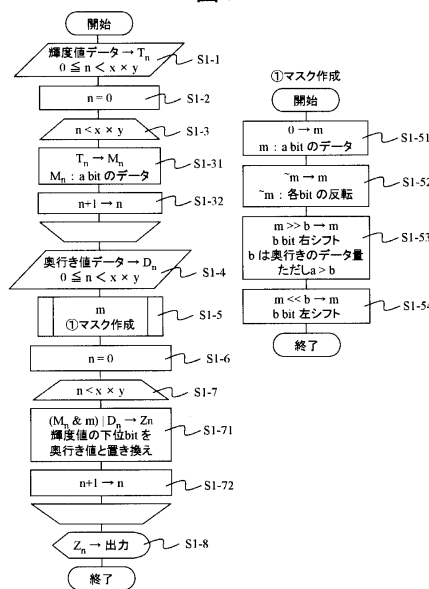
(57) 【要約】

【課題】従来の立体情報を扱うデータ形式では、一般的にデータ量が肥大してしまい通信に向いていなかった。またデータ量が肥大しない場合も専用のビューアーが必要であった。

【解決手段】画像データの画素ごとに輝度値を割り振る第1ステップと、画素ごとに割り振られた輝度値の下位bitを、その画素に対応する奥行きデータに置き換える第2ステップと、により画像データへ立体情報埋め込む。これにより、従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく奥行き値を埋め込むことが可能となる。

【選択図】 図1

図1



【特許請求の範囲】

【請求項 1】

モノクロ画像データに奥行きデータを埋め込む方法であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る第 1 ステップと、
前記割り振られた輝度値の下位 b i t を、当該画素に対応する前記奥行きデータに置き換える第 2 ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

【請求項 2】

モノクロ画像データに奥行きデータを埋め込む方法であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る第 1 ステップと、
前記モノクロ画像データと前記奥行きデータとから各画素ごとに第 2 の奥行きデータを作成する第 2 ステップと、
前記割り振られた輝度値の下位 b i t を、当該画素に対応する前記第 2 の奥行きデータに置き換える第 3 ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

10

【請求項 3】

カラー画像データに奥行きデータを埋め込む方法であって、
前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る第 1 ステップと、
前記割り振られた赤・緑・青の各輝度値ごとに個別に定められた b i t 長の下位 b i t を、
当該画素に対応する前記奥行きデータを分割したデータに置き換える第 2 ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

20

【請求項 4】

カラー画像データに奥行きデータを埋め込む方法であって、
前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る第 1 ステップと、
前記カラー画像データと前記奥行きデータとから各画素ごとに第 2 の奥行きデータを作成する第 2 ステップと、
前記割り振られた赤・緑・青の各輝度値ごとに個別に定められた b i t 長の下位 b i t を、
当該画素に対応する前記第 2 の奥行きデータを分割したデータに置き換える第 3 ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

30

【請求項 5】

モノクロ画像データに視差情報を埋め込む方法であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る第 1 ステップと、
前記割り振られた輝度値の下位 b i t の一部を、当該画素に対応する視差の有無を表わすデータに置き換える第 2 ステップと、
視差のある画素に対して前記輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、当該画素に対応する視差量を表わすデータに置き換える第 3 ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

40

【請求項 6】

請求項 5 において、
前記モノクロ画像の対になる視差画像上にのみ表示されるオクルージョン画素を抽出する第 4 ステップと、
前記抽出された順に前記オクルージョン画素の輝度値データの上位 b i t から所定の b i t 数分のみを画素ごとに複数のデータに分割する第 5 ステップと、
視差のない画素に対して前記輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、前記分割されたデータに順次置き換えていく第 6 ステップと、
をさらに含むことを特徴とする画像データへの立体情報埋め込み方法。

【請求項 7】

50

カラー画像データに視差情報を埋め込む方法であって、
前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る第1ステップと、
前記割り振られた赤・緑・青の各輝度値ごとに個別に定められたbit長の下位bitの一部を、当該画素に対応する視差の有無を表わすデータに置き換える第2ステップと、
視差のある画素に対して前記赤・緑・青の各輝度値の下位bitのうち前記視差の有無を表わすデータに使用されなかったbitを、当該画素に対応する視差量を表わすデータに置き換える第3ステップと、
を含むことを特徴とする画像データへの立体情報埋め込み方法。

【請求項8】

請求項7において、
前記カラー画像の対になる視差画像上にのみ表示されるオクルージョン画素を抽出する第4ステップと、
前記抽出された順に前記オクルージョン画素の赤・緑・青の各輝度値データの上位bitから所定のbit数分のみを画素ごとに複数のデータに分割する第5ステップと、
視差のない画素に対して前記赤・緑・青の各輝度値の下位bitのうち前記視差の有無を表わすデータに使用されなかったbitを、前記分割されたデータに順次置き換えていく第6ステップと、
をさらに含むことを特徴とする画像データへの立体情報埋め込み方法。

10

【請求項9】

モノクロ画像データに奥行きデータを埋め込む装置であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る手段と、
前記割り振られた輝度値の下位bitを、当該画素に対応する前記奥行きデータに置き換える下位bit置き換え手段と、
を具備することを特徴とする画像データへの立体情報埋め込み装置。

20

【請求項10】

モノクロ画像データに奥行きデータを埋め込む装置であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る手段と、
前記モノクロ画像データと前記奥行きデータとから各画素ごとに第2の奥行きデータを作成する埋め込みデータ作成手段と、
前記割り振られた輝度値の下位bitを、当該画素に対応する前記第2の奥行きデータに置き換える下位bit置き換え手段と、
を具備することを特徴とする画像データへの立体情報埋め込み装置。

30

【請求項11】

カラー画像データに奥行きデータを埋め込む装置であって、
前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る手段と、前記割り振られた赤・緑・青の各輝度値ごとに個別に定められたbit長の下位bitを、当該画素に対応する前記奥行きデータを分割したデータに置き換える下位bit置き換え手段と、
を具備することを特徴とする画像データへの立体情報埋め込み装置。

【請求項12】

カラー画像データに奥行きデータを埋め込む装置であって、
前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る手段と、
前記カラー画像データと前記奥行きデータとから各画素ごとに第2の奥行きデータを作成する埋め込みデータ作成手段と、
前記割り振られた赤・緑・青の各輝度値ごとに個別に定められたbit長の下位bitを、当該画素に対応する前記第2の奥行きデータを分割したデータに置き換える下位bit置き換え手段と、
を具備することを特徴とする画像データへの立体情報埋め込み装置。

40

【請求項13】

モノクロ画像データに視差情報を埋め込む装置であって、
前記モノクロ画像データの画素ごとに輝度値を割り振る手段と、

50

前記割り振られた輝度値の下位 b i t の一部を、当該画素に対応する視差の有無を表わすデータに置き換えるとともに、

視差のある画素に対して前記輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、当該画素に対応する視差量を表わすデータに置き換える下位 b i t 置き換え手段と、

を具備することを特徴とする画像データへの立体情報埋め込み装置。

【請求項 14】

請求項 13 において、

前記モノクロ画像の対となる視差画像上にもみ表示されるオクルージョン画素を抽出するオクルージョン検出手段と、

10

前記抽出された順に前記オクルージョン画素の輝度値データの上位 b i t から所定の b i t 数分のみを画素ごとに複数のデータに分割する輝度値データ分割手段と、

視差のない画素に対して前記輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、前記分割されたデータに順次置き換えていく第 2 の下位 b i t 置き換え手段と、

をさらに具備することを特徴とする画像データへの立体情報埋め込み装置。

【請求項 15】

カラー画像データに視差情報を埋め込む装置であって、

前記カラー画像データの画素ごとに赤・緑・青の各輝度値を割り振る手段と、

前記割り振られた赤・緑・青の各輝度値ごとに個別に定められた b i t 長の下位 b i t の一部を、当該画素に対応する視差の有無を表わすデータに置き換えるとともに、

20

視差のある画素に対して前記赤・緑・青の各輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、当該画素に対応する視差量を表わすデータに置き換える下位 b i t 置き換え手段と、

を具備することを特徴とする画像データへの立体情報埋め込み装置。

【請求項 16】

請求項 15 において、

前記カラー画像の対となる視差画像上にもみ表示されるオクルージョン画素を抽出するオクルージョン検出手段と、

前記抽出された順に前記オクルージョン画素の赤・緑・青の各輝度値データの上位 b i t から所定の b i t 数分のみを画素ごとに複数のデータに分割する輝度値データ分割手段と

30

、
視差のない画素に対して前記赤・緑・青の各輝度値の下位 b i t のうち前記視差の有無を表わすデータに使用されなかった b i t を、前記分割されたデータに順次置き換えていく第 2 の下位 b i t 置き換え手段と、

をさらに具備することを特徴とする画像データへの立体情報埋め込み装置。

【請求項 17】

請求項 1 ないし 8 のいずれかに記載の画像データへの立体情報埋め込み方法におけるステップをコンピュータに実行させるためのプログラムとした

ことを特徴とする画像データへの立体情報埋め込みプログラム。

40

【請求項 18】

請求項 1 ないし 8 のいずれかに記載の画像データへの立体情報埋め込み方法におけるステップをコンピュータに実行させるためのプログラムとし、該プログラムを記録した

ことを特徴とする画像データへの立体情報埋め込みプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、2次元の画像データに立体情報を埋め込む方法、装置、プログラム、および記録媒体に関する。

50

【0002】

【従来の技術】

物体の立体形状を記述する最も一般的な従来法としてポリゴンを用いたものがある。ポリゴンとは立体形状の表現に用いる多角形で、物体の立体形状を多角形の面の集合とするサーフェスモデルなどで使用される。ポリゴンを用いて立体形状を用いているフォーマットの例として、CADのデータ交換に用いられるIGESやWebでの立体形状表示用のVRMLなどがある。しかしポリゴンを用いた立体形状の記述は詳細なモデルを表現しようとするデータ量が膨れ上がるという問題点がある。また表示には専用のビューアが必要となり、一般的に普及している2次元画像を表示しているビューアでは見ることが出来ない。

10

【0003】

上記データ量肥大を解決した従来法としてXVL技術がある。XVL技術とは大まかな格子点と、その間を内挿する滑らかな自由曲面で立体形状を表現する技術で、ポリゴンを用いたものに比べて大幅にデータ量を減らすことが可能になる。しかし、この場合も表示には専用のビューアを用いる必要があり、2次元画像との互換性を持たない(非特許文献1参照)。

【0004】

また立体視用に特化した従来法として、単純に2枚のステレオ画像を用意するものがある。この方法は2枚の画像のうち片方だけを用いれば2次元画像として用いることはできるが、データを単一のファイルとして管理することができない・データ量が2倍に増えてしまうといった問題がある(非特許文献2参照)。

20

【0005】

本技術の類似技術として、原画像を任意サイズの複数ブロックに分割し、各ブロックごとの高さ情報をコード化したものを原画像の特定周波数成分に電子透かしとして埋め込む技術が開示されている。この技術では、埋め込み時に人間が知覚できない周波数成分を選択し変更量を随時決定する工夫により、原画像の劣化を防ぐことが可能となる。

【0006】

しかし、デコードを行うためには高さ情報の埋め込み位置と変更量を鍵情報として別途取得し同時に配布しなければならない問題がある。また、鍵情報をあらかじめ決めておき専用ビューアに持たせることもあるが、その場合は原画像が劣化する問題がある。さらに、コードは各領域ごとに割り振られるため、基本的には各領域ごとにしか高さを割り振ることができない問題点がある。なお、1つのコードが領域内の各画素ごとに付与した高さ情報を表すこととすれば、原理的には全画素の高さ情報を割り振ることができ、専用ビューアにコードと高さの対応を持たせることが困難となる(特許文献1参照)。

30

【0007】

【特許文献1】

特開平10-293861号公報

【非特許文献1】

ラティス・テクノロジー株式会社、「XVLとは? - 標準化に向けて」、[online]、ラティス・テクノロジー株式会社、[平成14年12月2日検索]、インターネット <URL: <http://www.xvl3d.com/ja/whatsxvl/sta.htm>>

40

【非特許文献2】

井下哲夫、「エッジ情報を利用した画像圧縮」、[online]、大阪大学大学院 基礎工学研究科、[平成14年12月2日検索]、インターネット <URL: <http://www.sys.es.osaka-u.ac.jp/Paper/B1998/inoshita.html>>

【0008】

【発明が解決しようとする課題】

近年、PCや携帯端末を用いて画像データの通信を行い付属の画像ビューアを用いて閲

50

覧することが一般的になっている。現状では主に扱われる画像データは輝度情報のみであるが、画像の立体情報のやり取りが同時にできれば画像の活用の幅が広がり利用方法が増大する。

【0009】

しかしながら従来の立体情報を扱うデータ形式では、一般的にデータ量が肥大してしまい通信に向いていなかった。またデータ量が肥大しない場合も専用のビューアーが必要であった。以上の問題により、立体情報を含む画像データは、主にオフラインで専用のビューアーを持った端末同士でのみ交換されていた。

【0010】

【課題を解決するための手段】

本発明は上記の課題を鑑みて発明されたものであり、従来の画像データと互換性を持ち、専用ビューアーで閲覧した場合のみ立体情報が反映される画像データへの立体情報埋め込み方法、装置、プログラム、および記録媒体を提供する。

【0011】

具体的には従来の画像データが持つ各画素の輝度値の下位bitを、その画素に対応する立体情報と置き換える。下位bitを立体情報と置き換えても従来のビューアーでは置き換えられたことが認識できないので、下位bitが改変された従来の画像データとして読み込まれる。一方、専用ビューアーでは下位bitが立体情報として扱えるので表示に立体形状の反映が可能になる。なお専用ビューアーでの立体情報埋め込みの有無に対する判断は、画像データのヘッダーを用いるか、もしくは画像データの特定位置の下位bitに立体情報以外の固有の情報を埋め込むことにより解決する。

【0012】

埋め込む立体情報は復元すべきデータによって決定する。復元すべきデータが立体形状の場合は、画素ごとにその画素における奥行き情報を埋め込む。また復元すべきデータがステレオ画像の場合は、画素ごとにその画素における視差情報を埋め込む。

【0013】

本発明では画像データの下位bitに立体情報を埋め込むため真の輝度値を改変することになる。しかし人間が分別できる彩度は赤や青でおよそ40段階(5~6bit)なので、一般的に用いられているRGB24bit画像では、赤成分と青成分の下位2~3bit程度は改変しても多くの場合知覚されることはない。また視感度が高い緑成分も下位2bit程度までは改変の影響は小さい。

【0014】

【発明の実施の形態】

以下、実施例に基づいて、この発明をさらに詳しく説明する。

【0015】

[実施例1]

図1は本発明の第1の実施例(請求項1に記載の発明に対応する実施例)のフローを示す図である。以下、図1を用いて詳細な動作の説明を行う。

【0016】

まずモノクロ画像(横方向x画素、縦方向y画素)の輝度値データ T_n を読み込む(ステップS1-1)。

【0017】

次に読み込んだモノクロ画像の輝度値をa bitの画素ごとのデータ M_n として割り振る(ステップS1-2、S1-3)。本実施例では一般的な256階調の輝度値を持つモノクロ画像として、a bit = 8 bitとする。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0018】

次に奥行き値データ D_n の読み込みを行う(ステップS1-4)。奥行き値データは輝度値データと同じ画像サイズ(横方向x画素、縦方向y画素)を持つb bit(ただしa > b)のデータとし、本実施例では2 bitのbit長と定義する。

10

20

30

40

50

【0019】

続いて輝度値データの下位 b bit を奥行き値データ D_n と置き換えるためのマスクを作成する (ステップ $S1-5$)。マスクは a bit 長の長さを持ち、下位 b bit が 0、残りの上位 b bit が 1 で表される。具体的には a bit 長のデータに 0 を代入したものを反転し (ステップ $S1-51$ 、 $S1-52$)、それを b bit だけ右シフトした後、 b bit だけ左シフトすることにより作成する (ステップ $S1-53$ 、 $S1-54$)。ここで、 $m \gg b$ は m を b bit 右シフトする演算、 $m \ll b$ は m を b bit 左シフトする演算を表す。

【0020】

ここで作成したマスクを用いて、各々の画素に対応する輝度値の下位 b bit を奥行き値と置き換える (ステップ $S1-7$)。置き換えた値は埋め込みデータ Z_n として出力される (ステップ $S1-8$)。

10

【0021】

なお下位 b bit への奥行き値の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。

【0022】

図 2 は実際の埋め込み例を示す。2-1 は 8 bit の輝度値データ、2-2 は 2 bit の奥行き値データ (ただし図では 2 bit で表すことができる 4 階調を強調表示してある)、2-3 は 2-1 に 2-2 を埋め込んだデータを従来のビューアーで表示した例である。

【0023】

図 3 は埋め込みデータをビューアーで表示する場合を示す。埋め込み画像 3-1 を従来型の画像ビューアー 3-21 で表示した場合は、従来と同様の画像出力 3-22 を得ることができる。一方、埋め込み画像 3-1 を専用のビューアー 3-31 で表示した場合は、従来と同様の画像出力と奥行き値データ出力の 2 つ 3-32 を同時に得ることができる。専用のビューアー 3-31 で表示する場合、奥行き値埋め込みの有無に対する判断はヘッダー情報から判断する。具体的には、本実施例で用いる画像フォーマットを png 形式とし、ヘッダーの付随的ブロック部にあるテキスト・データに奥行き値の有無を判別する情報を埋め込む。付随的ブロックは通常エンコードはそれらを書く必要がなく、デコードはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコードでは解釈されない。またヘッダーに奥行き値の有無を埋め込むのではなく、画面の上下左右の周囲の 1 ラインずつの輝度値データのいずれかの下位 b bit を奥行き値の変わりに奥行き値の有無を判断する情報と置き換えても良い。なお、本発明の画像フォーマットは上記実施例に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダーの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

20

30

【0024】

以上の手順により、モノクロ画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく、奥行き値を埋め込むことが可能となる。

【0025】

[実施例 2]

図 4 は本発明の第 2 の実施例 (請求項 1 に記載の発明に対応する実施例) のフローを示す図である。以下、図 4 を用いて詳細な動作の説明を行う。

40

【0026】

まずモノクロ画像 (横方向 x 画素、縦方向 y 画素) の輝度値データ T_n を読み込む (ステップ $S4-1$)。次に読み込んだモノクロ画像の輝度値を a bit の画素ごとのデータ M_n として割り振る (ステップ $S4-2$ 、 $S4-3$)。本実施例では一般的な 256 階調の輝度値を持つモノクロ画像として、 a bit = 8 bit とする。このように一度割り振りをを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0027】

50

次に奥行き値データ D_n の読み込みを行う (ステップ S 4 - 4)。奥行き値データは輝度値データと同じ画像サイズ (横方向 x 画素、縦方向 y 画素) を持つ任意の b bit のデータとする。本実施例では b bit = a bit とし、埋め込み可能な b bit 長を c bit とする。

【0028】

このとき、埋め込み可能なデータ長 (c) と奥行き値データ D_n の b bit 長 (b) が異なるため変換を行う (ステップ S 4 - 5)。具体的には、 D_n が表現可能な最大変位量を埋め込み可能なデータ長 c が表現可能な最大変位量で除算を行い c の最小変位量に対する D_n の変位量 l を求め (ステップ S 4 - 5 1、S 4 - 5 2)、 D_n の各画素に対応する奥行き値を l で除算することにより変換した奥行き値データ D'_n を得る (ステップ S 4 - 5 4)。

【0029】

続いて輝度値データの下位 b bit を変換した奥行き値データ D'_n と置き換えるためのマスクを作成する (ステップ S 4 - 6)。マスクは a bit 長の長さを持ち、下位 c bit が 0、残りの上位 b bit が 1 で表される。具体的には a bit 長のデータに 0 を代入したものを反転し (ステップ S 4 - 6 1、S 4 - 6 2)、それを c bit だけ右シフトした後、 c bit だけ左シフトすることにより作成する (ステップ S 4 - 6 3、S 4 - 6 4)。ここで作成したマスクを用いて、各々の画素に対応する輝度値の下位 b bit を奥行き値と置き換える (ステップ S 4 - 8)。置き換えた値は埋め込みデータ Z_n として出力される (ステップ S 4 - 9)。

【0030】

なお下位 b bit への奥行き値の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。

【0031】

図 5 は本実施例でのエンコード・デコードの例を示す。5 - 1 は入力データを示し、5 - 1 1 は 8 bit の輝度値データ、5 - 1 2 は 8 bit の奥行きデータである。5 - 2 は埋め込み可能なデータ長を 2 bit としたときに演算される埋め込み用の奥行き値データである。5 - 3 は実施例 1 に示される方法などを用いて 5 - 1 1 に 5 - 2 を埋め込んだデータである。奥行き値を埋め込んだデータ 5 - 3 を専用ビューア 5 - 4 を用いてデコードすると、輝度値データ 5 - 5 1、奥行き値データ 5 - 5 2 の両方を含む出力 5 - 5 を得ることができる。ここで輝度値データ 5 - 5 1 は 5 - 3、奥行き値データ 5 - 5 2 は 5 - 2 と等価である。デコードされる奥行き値データ 5 - 5 2 は、入力した奥行き値データ 5 - 1 2 を 8 bit から 2 bit に変換したため離散的なものに変換される。

【0032】

以上の手順により、モノクロ画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、奥行き値を埋め込むことが可能となる。

【0033】

[実施例 3]

図 6 は本発明の第 3 の実施例 (請求項 2 に記載の発明に対応する実施例) のフローを示す図である。以下、図 6 を用いて詳細な動作の説明を行う。

【0034】

まずモノクロ画像 (横方向 x 画素、縦方向 y 画素) の輝度値データ T_n を読み込む (ステップ S 6 - 1)。次に読み込んだモノクロ画像の輝度値を a bit の画素ごとのデータ M_n として割り振る (ステップ S 6 - 2、S 6 - 3)。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0035】

次に奥行き値データ D_n の読み込みを行う (ステップ S 6 - 4)。奥行き値データは輝度値データと同じ画像サイズ (横方向 x 画素、縦方向 y 画素) を持つ任意の b bit のデータとする。

【0036】

10

20

30

40

50

続いて埋め込み用の奥行き値データ D'_n を、 M_n 、および D_n から演算し（ステップ S6-5）、画素ごとに埋め込む bit 長を算出する。

【0037】

ところでコンピュータにより画面に表示される画像は通常補正が行われて出力される。この値は指定を行わなくても OS によって決められた値があり、Windows では 2.2、Macintosh では 1.8 が用いられている。値が 1 より大きいとき画像は明るくなり、小さければ暗くなる。つまり 1 より大きければ補正式に従って輝度値は入力より出力の方が大きくなる。

【0038】

Windows と Macintosh の平均値を取って $\gamma = 2.0$ を選択したとき、最大輝度値における 94% の値を入力すれば補正式に従って 97% の値で出力される。輝度値データが 8 bit であるとき、下位 3 bit は約 3% (100% - 97%) の変位に相当し、下位 4 bit は約 6% (100% - 94%) の変位に相当する。すなわち補正による出力を考えると、入力がある程度大きければ 4 bit の改変を行っても出力としては 3 bit の改変にしか相当しない。

【0039】

埋め込みデータの作成は次の手順で行う。まず始めに輝度値データ M_n の値が低輝度であるときの埋め込みデータの bit 長および高輝度であるときの埋め込みデータの bit 長を呼び出す（ステップ S6-51）。

【0040】

このとき、埋め込みデータの bit 長と奥行き値データ D_n の bit 長が異なるため変換を行う。具体的には D_n が表現可能な最大変位量を、低輝度時および高輝度時の埋め込みデータが表現可能な最大変位量で除算を行い、低輝度時および高輝度時の最小変位量に対する変位量を求め（ステップ S6-52）、 D_n の各画素に対応する奥行き値を求めた変位量で除算することにより変換した奥行き値データ D'_n を得る（ステップ S6-542、S6-543）。なお専用ビューアーにてデコードを行うときは上位 4 bit の値が全て 1 のとき高輝度側と判断する。

【0041】

続いて輝度値データの下位 bit を変換した埋め込み用奥行き値データ D'_n と置き換えるためのマスクを作成する（ステップ S6-6）。マスクは a bit 長の長さを持ち、低輝度値側の場合は下位 c bit の値が 0・残りの上位 bit が 1、高輝度値側の場合は下位 d bit が 0・残りの上位 bit が 1 で表される。ここで $d > c$ とする。具体的には a bit 長のデータに 0 を代入したものを反転し（ステップ S6-61、S6-62）、それを c bit (または d bit) だけ右シフト、左シフトの順で操作することにより低輝度用マスク M_c と高輝度用マスク M_d を作成する（ステップ S6-63、S6-64）。

【0042】

ここで作成したマスクを用いて、各々の画素に対応する輝度値の下位 bit を奥行き値と置き換える（ステップ S6-8）。具体的には輝度値データ M_n が低輝度のときはマスク M_c 、高輝度のときはマスク M_d を用いて、輝度値データ M_n の下位 bit を埋め込み用奥行き値データ D'_n と置き換える。

【0043】

置き換えた値は埋め込みデータ Z_n として出力される（ステップ S6-9）。なお下位 bit への奥行き値の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。また今回は輝度値データから再現される画像内において輝度が高い画素により多くの情報量を埋め込む演算を行ったが、画像内の各画素に埋め込みデータが目立たないように多くのデータ埋め込む演算であればよい。例として電子透かしを埋め込む場合と同様に、画像内のエッジ部分などの高周波成分となる画素により多くの情報量を埋め込む演算などがある。

【0044】

10

20

30

40

50

以上の手順によりモノクロ画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、より多くの奥行き値を埋め込むことが可能となる。

【0045】

[実施例4]

図7は本発明の第4の実施例(請求項3に記載の発明に対応する実施例)のフローを示す図である。以下、図7を用いて詳細な動作の説明を行う。

【0046】

まずカラー画像(横方向 x 画素、縦方向 y 画素)の輝度値データ T_n を読み込む(ステップS7-1)。

【0047】

次に読み込んだカラー画像の輝度値をRGB各成分に付いて $a \times 3 \text{ bit}$ の画素ごとのデータ M_{RGBn} として割り振る(ステップS7-2、S7-3)。 M_{RGBn} は M_{Rn} 、 M_{Gn} 、 M_{Bn} より構成され、それぞれが $a \text{ bit}$ 長のカラー画像のR成分、G成分、B成分にあたるデータである。本実施例では、一般的なRGB成分それぞれ 8 bit の輝度値を持つカラー画像として、 $a \text{ bit} = 8 \text{ bit}$ とする。このように一度割り振りをを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0048】

次に奥行き値データ D_n の読み込みを行う(ステップS7-4)。奥行き値データは輝度値データと同じ画像サイズ(横方向 x 画素、縦方向 y 画素)を持つ $b \text{ bit}$ (ただし $a \times 3 > b$)のデータとし、本実施例では 4 bit の $b \text{ bit}$ 長と定義する。

【0049】

続いて輝度値データの下位 $b \text{ bit}$ を奥行き値データ D_n と置き換えるためのマスクを作成する(ステップS7-5)。マスク m_{RGB} は m_R 、 m_G 、 m_B より構成され、それぞれ $a \text{ bit}$ 長のR成分、G成分、B成分用のマスクである。RGB成分各要素に埋め込めるデータ長をそれぞれ b_R 、 b_G 、 b_B としデータ長の合計は $b \text{ bit}$ と等価(図8)としたとき、 m_R 、 m_G 、 m_B はそれぞれ下位 b_R 、 b_G 、 $b_B \text{ bit}$ が0、残りの上位 $b \text{ bit}$ が1で表される。また b_R 、 b_G 、 b_B の $b \text{ bit}$ 長の関係は $b_G > b_R > b_B$ かつ、 b_G が上位 $b \text{ bit}$ 側・ b_B が下位 $b \text{ bit}$ 側となるようにする(b_R は中間)。これは人間の視感度の強さが緑、赤、青の順で弱くなることを利用し人間が知覚しにくい成分に、より多くのデータを埋め込むためである。本実施例では $b_G = 0 \text{ bit}$ 、 $b_R = 2 \text{ bit}$ 、 $b_B = 2 \text{ bit}$ とした。

【0050】

マスクの具体的作成方法は m_R 、 m_G 、 m_B に0を代入して反転し(ステップS7-51、S7-52)、それぞれ b_R 、 b_G 、 $b_B \text{ bit}$ ずつ右シフト、左シフトの順で操作することにより作成する(ステップS7-53、S7-54)。

【0051】

ここで作成したマスクを用いて、各々の画素に対応するRGB成分の輝度値の下位 $b \text{ bit}$ を奥行き値と置き換える(ステップS7-6、S7-7)。具体的には奥行き値データ D_n が持つ各画素 $b \text{ bit}$ のデータを b_R 、 b_G 、 $b_B \text{ bit}$ ずつ分割し、埋め込みデータ Z_{RGBn} の下位 $b \text{ bit}$ と置き換える(ステップS7-71、S7-72)。

【0052】

置き換えた値は埋め込みデータ Z_{RGBn} として出力される(ステップS7-8)。

【0053】

なお下位 $b \text{ bit}$ への奥行き値の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。

【0054】

図9は実際の埋め込み例を示す。9-1はRGB成分各 8 bit の輝度値データ、9-2は 4 bit の奥行き値データ(ただし図では 4 bit で表すことができる16階調を強調表示してある)、9-3は9-1に9-2を埋め込んだデータを従来のビューアーで表示した例である。

10

20

30

40

50

【0055】

図10は埋め込みデータをビューアーで表示する場合を示す。埋め込み画像10-1を従来型の画像ビューアー10-21で表示した場合は、従来と同様の画像出力10-22を得ることができる。一方、埋め込み画像10-1を専用のビューアー10-31で表示した場合は、従来と同様の画像出力と奥行き値データ出力の2つ10-32を同時に得ることができる。専用のビューアー10-31で表示する場合、奥行き値埋め込みの有無に対する判断はヘッダー情報から判断する。具体的には、本実施例で用いる画像フォーマットをpng形式とし、ヘッダーの付随的ブロック部にあるテキストデータに奥行き値の有無を判別する情報を埋め込む。付随的ブロックは通常エンコードはそれらを書く必要がなく、デコードはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコードでは解釈されない。またヘッダーに奥行き値の有無を埋め込むのではなく、画面の上下左右の周囲の1ラインずつの輝度値データのいずれかの下位bitを奥行き値の変わりに奥行き値の有無を判断する情報と置き換えても良い。なお、本発明の画像フォーマットは上記実施例に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダーの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

10

【0056】

以上の手順により、カラー画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく、奥行き値を埋め込むことが可能となる。

【0057】

20

[実施例5]

図11は本発明の第5の実施例(請求項3に記載の発明に対応する実施例)のフローを示す図である。以下、図11を用いて詳細な動作の説明を行う。

【0058】

まずカラー画像(横方向x画素、縦方向y画素)の輝度値データ T_n を読み込む(ステップS11-1)。

【0059】

次に読み込んだカラー画像の輝度値をRGB各成分に付いて $a \times 3 \text{ bit}$ の画素ごとのデータ M_{RGBn} として割り振る(ステップS11-2、S11-3)。 M_{RGBn} は M_{Rn} 、 M_{Gn} 、 M_{Bn} より構成され、それぞれが $a \text{ bit}$ 長のカラー画像のR成分、G成分、B成分にあたるデータである。本実施例では、一般的なRGB成分それぞれ 8 bit の輝度値を持つカラー画像として、 $a \text{ bit} = 8 \text{ bit}$ とする。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

30

【0060】

次に奥行き値データ D_n の読み込みを行う(ステップS11-4)。奥行き値データは輝度値データと同じ画像サイズ(横方向x画素、縦方向y画素)を持つ $b \text{ bit}$ のデータとする。本実施例では $b \text{ bit} = a \text{ bit}$ とし、埋め込み可能な $b \text{ bit}$ 長を $c \text{ bit}$ とする。

【0061】

40

このとき、埋め込み可能なデータ長(c)と奥行き値データ D_n の $b \text{ bit}$ 長(b)が異なるため変換を行う(ステップS11-5)。具体的には、 D_n が表現可能な最大変位量を埋め込み可能なデータ長 c が表現可能な最大変位量で除算を行い c の最小変位量に対する D_n の変位量 l を求め(ステップS11-51、S11-52)、 D_n の各画素に対応する奥行き値を l で除算することにより変換した奥行き値データ D'_n を得る(ステップS11-54)。

【0062】

続いて輝度値データの下位 $b \text{ bit}$ を変換した奥行き値データ D'_n と置き換えるためのマスクを作成する(ステップS11-6)。マスク m_{RGB} は m_R 、 m_G 、 m_B より構成され、それぞれ $a \text{ bit}$ 長のR成分、G成分、B成分用のマスクである。RGB成分各要

50

素に埋め込めるデータ長をそれぞれ c_R 、 c_G 、 c_B とし、データ長の合計は c bit と等価 (図 12) としたとき、 m_R 、 m_G 、 m_B はそれぞれ下位 c_R 、 c_G 、 c_B bit が 0、残りの上位 bit が 1 で表される。また c_R 、 c_G 、 c_B の bit 長の関係は $c_G > c_R > c_B$ かつ、 c_G が上位 bit 側・ c_B が下位 bit 側となるようにする (c_R は中間)。これは人間の視感度の強さが緑、赤、青の順で弱くなることを利用し人間が知覚しにくい成分に、より多くのデータを埋め込むためである。本実施例では $c_G = 0$ bit、 $c_R = 2$ bit、 $c_B = 2$ bit とした。

【0063】

マスクの具体的作成方法は m_R 、 m_G 、 m_B それぞれに 0 を代入して反転し (ステップ S11-61、S11-62)、それぞれ c_R 、 c_G 、 c_B bit ずつ右シフト、左シフトの順で操作することにより作成する (ステップ S11-63、S11-64)。

【0064】

ここで作成したマスクを用いて、各々の画素に対応する RGB 成分の輝度値の下位 bit を奥行き値と置き換える (ステップ S11-7、S11-8)。具体的には奥行き値データ D_n が持つ各画素 c bit のデータを c_R 、 c_G 、 c_B bit ずつ分割し、埋め込みデータ Z_{RGBn} の下位 bit と置き換える (ステップ S11-81、S11-82)。

【0065】

置き換えた値は埋め込みデータ Z_{RGBn} として出力される (ステップ S11-9)。なお下位 bit への奥行き値の埋め込み方法は上記アルゴリズムに限定されることはなく、同様の結果を得られれば良い。

【0066】

図 13 は本実施例でのエンコード・デコードの例を示す。13-1 は入力データを示し、13-11 は RGB 成分各 8 bit の輝度値データ、13-12 は 8 bit の奥行き値データである。13-2 は埋め込み可能なデータ長を 4 bit としたときに演算される埋め込み用の奥行き値データである。13-3 は 13-11 に 13-2 を埋め込んだデータである。奥行き値を埋め込んだデータ 13-3 を専用ビューアー 13-4 を用いてデコードすると、輝度値データ 13-51、奥行き値データ 13-52 の両方を含む出力 13-5 を得ることができる。ここで輝度値データ 13-51 は 13-3、奥行き値データ 13-52 は 13-2 と等価である。デコードされる奥行き値データ 13-52 は、入力した奥行き値データ 13-12 を 8 bit から 4 bit に変換したため離散的なものに変換される。

【0067】

以上の手順により、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、奥行き値を埋め込むことが可能となる。

【0068】

[実施例 6]

図 14 は本発明の第 6 の実施例 (請求項 4 に記載の発明に対応する実施例) のフローを示す図である。以下、図 14 を用いて詳細な動作の説明を行う。

【0069】

まずカラー画像 (横方向 x 画素、縦方向 y 画素) の輝度値データ、 T_n を読み込む (ステップ S19-1)。

【0070】

次に読み込んだカラー画像の輝度値を RGB 各成分に付いて $a \times 3$ bit の画素ごとのデータ M_{RGBn} として割り振る (ステップ S14-2、S14-3)。 M_{RGBn} は M_{Rn} 、 M_{Gn} 、 M_{Bn} より構成され、それぞれが a bit 長のカラー画像の R 成分、G 成分、B 成分にあたるデータである。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0071】

次に奥行き値データ D_n の読み込みを行う (ステップ S14-4)。奥行き値データは輝

度値データと同じ画像サイズ（横方向 x 画素、縦方向 y 画素）を持つ b bit（ただし $a \times 3 > b$ ）のデータとする。

【0072】

続いて埋め込み用の奥行き値データ D'_n を、 M_n および D_n から演算し（ステップ S14-5）、画素ごとに埋め込む bit 長を算出する。

【0073】

ところでコンピュータにより画面に表示される画像は通常補正が行われて出力される。この値は指定を行わなくても OS によって決められた値があり、Windows では 2.2、Macintosh では 1.8 が用いられている。値が 1 より大きいとき画像は明るくなり、小さければ暗くなる。つまり 1 より大きければ補正式に従って輝度値は入力より出力の方が大きくなる。

10

【0074】

Windows と Macintosh の平均値を取って $= 2.0$ を選択したとき、最大輝度値における 94% の値を入力すれば補正式に従って 97% の値で出力される。輝度値データが 8 bit であるとき、下位 3 bit は約 3%（100% - 97%）の変位に相当し、下位 4 bit は約 6%（100% - 94%）の変位に相当する。すなわち補正による出力を考えると、入力がある程度大きければ 4 bit の改変を行っても出力としては 3 bit の改変にしか相当しない。

【0075】

埋め込みデータの作成は次の手順で行う。まず始めに輝度値データ M_{RGBn} の各成分の値が低輝度であるときの埋め込みデータの bit 長および高輝度であるときの埋め込みデータの bit 長を呼び出す（ステップ S14-51）。

20

【0076】

このとき、埋め込みデータの bit 長と奥行き値データ D_n の bit 長が異なるため変換を行う。具体的には全ての画素に対して次のような変換作業を行う（ステップ S14-52、S14-53）。画素ごとに M_{RGBn} の各成分に埋め込めるデータの bit 長を求め、その bit 長の合計を対応する画素の埋め込みデータ長 e とする（ステップ S14-53 ~ S14-5351、5352）。 D_n が表現可能な最大変位量を、埋め込みデータ長 e が表現可能な最大変位量で画素ごとに除算することによりスケール変換を行い、埋め込み用奥行き値データ D'_n を得る（ステップ S14-536、S14-537）。なお専用ビューアーにてデコードを行うときは画素ごとの各成分における上位 4 bit の値が全て 1 のとき高輝度側と判断する。

30

【0077】

続いて輝度値データの下位 bit を変換した埋め込み用奥行き値データ D'_n と置き換えるためのマスクを作成する（ステップ S14-6）。マスクは a bit 長の長さを持ち、低輝度値側の場合は下位 c bit の値が 0・残りの上位 bit が 1、高輝度値側の場合は下位 d bit が 0・残りの上位 bit が 1 で表される。ここで $d > c$ とする。具体的には a bit 長のデータに 0 を代入したものを反転し（ステップ S14-61、S14-62）それを c bit（または d bit）だけ右シフト、左シフトの順で操作することにより低輝度用マスク M_c と高輝度用マスク M_d を作成する（ステップ S14-63、S14-64）。

40

【0078】

ここで作成したマスクを用いて、各々の画素に対応する輝度値の下位 bit を奥行き値と置き換える（ステップ S14-8）。具体的には輝度値データ M_{RGBn} において各成分ごとに低輝度・高輝度の判定を行い、低輝度の成分にはマスク M_c 、高輝度の成分にはマスク M_d を用いて、輝度値データ M_{RGBn} の下位 bit を埋め込み用奥行き値データ D'_n を分割したものと置き換える。分割の手順は下位 bit より各成分に埋め込み可能な bit 長ずつ切り取っていく。

【0079】

置き換えた値は埋め込みデータ Z_{RGBn} として出力される（ステップ S14-9）。な

50

お下位 bit への奥行き値の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。また今回は輝度値データから再現される画像内において輝度が高い画素により多くの情報量を埋め込む演算を行ったが、画像内の各画素に埋め込みデータが目立たないようより多くのデータ埋め込む演算であればよい。例として画像内のエッジ部分などの高周波成分となる画素により多くの情報量を埋め込む演算などがある。

【0080】

以上の手順により、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、より多くの奥行き値を埋め込むことが可能となる。

【0081】

10

[実施例7]

図15は本発明の第7の実施例(請求項5に記載の発明に対応する実施例)のフローを示す図である。以下、図15を用いて詳細な動作の説明を行う。

【0082】

まずモノクロ画像(横方向 x 画素、縦方向 y 画素)の輝度値データ T_n を読み込む(ステップ $S15-1$)。

【0083】

次に読み込んだモノクロ画像の輝度値を a bit の画素ごとのデータ M_n として割り振る(ステップ $S15-2$ 、 $S15-3$)。本実施例では一般的な256階調の輝度値を持つモノクロ画像として、 a $bit = 8$ bit とする。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

20

【0084】

次に視差データ P_n の読み込みを行う(ステップ $S15-4$)。視差データは輝度値データの各画素に対応する視差量を持つ p bit のデータである。本実施例では p $bit = 1$ bit とする。また埋め込み可能な bit 長を p' bit とし、 p' bit は p bit と等価または $p+1$ bit の bit 長を持つ。

【0085】

ここで視差データ P_n (bit 長 p) を埋め込み用データ P'_n (bit 長 p') の変換を行う(ステップ $S15-5$)。具体的には、 P_n を P'_n にコピーし、さらに、視差の有無を判定可能とするため、視差がある画素は P'_n の最上位 bit を立てる(ステップ $S15-53$)。

30

【0086】

続いて輝度値データの下位 bit を埋め込み用データ P'_n と置き換えるためのマスクを作成する(ステップ $S15-6$)。マスクは a bit 長の長さを持ち、下位 p' bit が0、残りの上位 bit が1で表される。具体的には a bit 長のデータに0を代入したものを反転し(ステップ $S15-61$ 、 $S15-62$)、それを p' bit だけ右シフトした後、 p' bit だけ左シフトすることにより作成する(ステップ $S15-63$ 、 $S15-64$)。

【0087】

ここで作成したマスクを用いて、輝度値データ M_n の各々の画素 n に対応する輝度値の下位 bit を埋め込み用データ P'_n と置き換える(ステップ $S15-8$)。

40

【0088】

置き換えた値は埋め込み済みデータ Z_n として出力される(ステップ $S15-9$)。なお下位 bit への視差の埋め込み方法は上記アルゴリズムに限定されることはなく、同様の結果を得られれば良い。なお視差データの埋め込み手順は上記に限定されるものではなく、同様の結果を得られれば良い。

【0089】

図16は本実施例でのエンコード・デコードの例を示す図である。入力データ16-1は輝度値データ($a = 8$ bit)16-11、視差データ($p = 1$ bit)16-12から構成される。視差データ16-12は図15に示した方法により輝度値データ16-11

50

に埋め込まれ、埋め込み済み画像 16 - 2 を得る。視差データ 16 - 12 上で黒色で示される部分が視差量 0、白色で示される部分が視差量 1 を示す。なお視差量の値が必ずしも実際の視差と一致する必要は無く、予め指定しておいたパラメータにより変換してもよい。本実施例では 1 視差量が画像上の 10 画素を示すものとして予めパラメータを決める。また視差データ 16 - 12 の視差量 0 を視差無しの状態として代用する場合は視差の有無判定用の `bit` は不要となるため、 P_n から P'_n の変換 (S15 - 5) を省略し、 $P_n = P'_n$ としても良い。埋め込みデータ 16 - 2 を専用ビューアー 16 - 3 でデコードすることにより、出力データ 16 - 4 を得る。出力データは右目用画像 16 - 41 と左目用画像 16 - 42 から構成される。右目用画像 16 - 41 は埋め込み済み画像 16 - 2 と等価であり、また埋め込まれた視差量 1 につき 10 画素右に移動させることにより左目用画像 16 - 42 を得ることができる。本実施例では予め決めたパラメータより右目用画像から左目用画像の変換を行ったが、その逆に左目用画像から右目用画像の変換を行うことも容易である。専用ビューアーでデコードする際の情報はヘッダー部に埋め込む。具体的には、本実施例で用いる画像フォーマットを `png` 形式とし、ヘッダーの付随的ブロック部にあるテキストデータにパラメータ情報を埋め込む。付随的ブロックは通常エンコードはそれらを書く必要がなく、デコーダはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコーダでは解釈されない。またヘッダーにパラメータを埋め込むのではなく、視差量と同様に画像に埋め込んで良い。具体的には、画面の上下左右の周囲の 1 ラインずつの輝度値データのいずれかの下位 `bit` をパラメータ記述部と判断する方法などがある。

10

20

【0090】

なお本発明の画像フォーマットは上記に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダーの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

【0091】

以上の手順により、モノクロ画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

【0092】

[実施例 8]

図 17 は本発明の第 8 の実施例 (請求項 6 に記載の発明に対応する実施例) のフローを示す図である。以下、図 17 を用いて詳細な動作の説明を行う。

30

【0093】

まずモノクロ画像 (横方向 x 画素、縦方向 y 画素) の輝度値データ T_n を読み込む (ステップ S17 - 1)。

【0094】

次に読み込んだモノクロ画像の輝度値を a `bit` の画素ごとのデータ M_n として割り振る (ステップ S17 - 2、S17 - 3)。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0095】

次に視差データ P_n の読み込みを行う (ステップ S17 - 4)。視差データは輝度値データの各画素に対応する視差量を持つ p `bit` のデータである。本実施例では p `bit` = 1 `bit` とする。また埋め込み可能な `bit` 長を p' `bit` とし、 p' `bit` は p `bit` と等価または $p + 1$ `bit` の `bit` 長を持つ。

40

【0096】

ここで視差データ P_n (`bit` 長 p) から埋め込み用データ P'_n (`bit` 長 p') への変換を行う (ステップ S17 - 5)。具体的には、 P_n を P'_n にコピーし、さらに視差がある画素は P'_n の最上位 `bit` を立てる (ステップ S17 - 53)。

【0097】

続いて輝度値データの下位 `bit` を埋め込み用データ P'_n 、と置き換えるためのマスクを作成する (ステップ S17 - 6)。マスクは a `bit` 長の長さを持ち、下位 p' `bit`

50

bit が 0、残りの上位 bit が 1 で表される。具体的には a bit 長のデータに 0 を代入したものを反転し (ステップ S 17 - 61、S 17 - 62)、それを p' bit だけ右シフトした後、p' bit だけ左シフトすることにより作成する (ステップ S 17 - 63、S 17 - 64)。

【0098】

ここで作成したマスクを用いて、輝度値データ M_n の各々の画素 n に対応する輝度値の下位 bit を埋め込み用データ P'_n と置き換える (ステップ S 17 - 8)。

【0099】

次にオクルージョン部の検出を行う。検出方法を記述する前に、図 18 を用いてオクルージョン部について説明する。18 - 1 は輝度値データの例、18 - 2 は視差データの例を示している。視差データ 18 - 2 上で黒色で示される部分が視差量 0、白色で示される部分が視差量 1 を示す。なお視差量の値が必ずしも実際の視差と一致する必要は無く、デコード時には予め指定しておいたパラメータにより変換してもよい。本実施例では 1 視差量が画像上の 10 画素を示し、右方向に移動するものとして予めパラメータを決める。なおこのパラメータは必要な視差画像にあわせて任意に設定すればよい。18 - 3 はパラメータに従ってデコードされた元のデータ 18 - 1 に対する視差画像を示している。18 - 3 において白抜きの部分 18 - 31 は元のデータ 18 - 1 に変換元の画素が無かった部分 (画像の移動後にデータが空白となる部分)、つまり元のデータ 18 - 1 に対するオクルージョン部である。また 18 - 4 は元のデータ 18 - 1 において、視差画像 18 - 3 への変換時に移動した画素を示している。18 - 4 において白抜きの部分 18 - 41 は視差画像 18 - 3 に変換先の画素が無かった部分 (画像の移動後にかくれる部分)、つまり視差画像 18 - 3 に対するオクルージョン部である。ステップ S 17 - 9 で検出するオクルージョン部は 18 - 31 と 18 - 41 によって説明した 2 種類である。具体的に検出を行う場合、単純に移動元の有無および移動先の有無を調べればよい。移動元がない画素の領域は 18 - 31 部、移動先が無い画素の領域は 18 - 41 部に相当する。

【0100】

次に予め用意された背景画像からオクルージョン部の輝度値データを読み込む (ステップ S 17 - 10)。ただしここで輝度値データを読み込むオクルージョン部は 18 - 31 部に相当する領域だけである。

【0101】

この読み込んだ輝度値データをステップ S 17 - 9 で検出した 18 - 41 部および視差が無い画素 (ステップ S 17 - 53 で P'_n の最上位 bit を 0 とした領域) の下位 bit すなわち P'_n のうち判定用 bit を除いた部分に埋め込む (ステップ S 17 - 10)。埋め込み順は左から右へとライン単位で検索を行い検出された 18 - 31 部の画素ごとに視差画像に変換後の輝度値を分割し、同様に左から右へとライン単位で検索を行い検出された 18 - 41 部および視差が無い画素の下位 bit にシーケンシャルに埋め込んでいく。このようにシーケンシャルに埋め込みを行うことにより、画素の位置情報を埋め込まなくても、同様のシーケンシャルな検索によりデコード時に正しい画素位置に輝度値データを復元できる。

【0102】

以上の手順により下位 bit を置き換えたものを埋め込みデータ Z_n として出力する (ステップ S 17 - 12)。

【0103】

以上により抜けのない視差画像を再生可能な立体画像データを得ることができる。

【0104】

なお下位 bit への視差の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。

【0105】

上記、画像をデコードする際の情報はヘッダ部に埋め込む。具体的には、本実施例で用いる画像フォーマットを png 形式とし、ヘッダの付随的ブロック部にあるテキスト・

10

20

30

40

50

データにパラメータ情報を埋め込む。付随的ブロックは通常エンコーダはそれらを書く必要がなく、デコーダはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコーダでは解釈されずにそのまま表示される。またヘッダーにパラメータを埋め込むのではなく、視差量と同様に画像に埋め込んでも良い。具体的には、画面の上下左右の周囲の1ラインずつの輝度値データのいずれかの下位bitをパラメータ記述部と判断する方法などがある。なお、本発明の画像フォーマットは上記に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダーの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

【0106】

以上の手順により、モノクロ画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

[実施例9]

図19は本発明の第9の実施例(請求項7に記載の発明に対応する実施例)のフローを示す図である。以下、図19を用いて詳細な動作の説明を行う。

【0107】

まずカラー画像(横方向x画素、縦方向y画素)の輝度値データ T_n を読み込む(ステップS19-1)。

【0108】

次に読み込んだカラー画像の輝度値をRGB各成分に付いて $a \times 3$ bitの画素ごとのデータ M_{RGBn} として割り振る(ステップS19-2、S19-3)。 M_{RGBn} は M_{Rn} 、 M_{Gn} 、 M_{Bn} より構成され、それぞれが a bit長のカラー画像のR成分、G成分、B成分にあたるデータである。本実施例では、一般的なRGB成分それぞれ8 bitの輝度値を持つカラー画像として、 a bit = 8 bitとする。このように一度割り振りをを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0109】

次に視差データ P_n の読み込みを行う(ステップS19-4)。視差データは輝度値データの各画素に対応する視差量を持つ p bitのデータである。本実施例では p bit = 1 bitとする。また埋め込み可能なbit長を p' bitとし、 p' bitは p bitと等価または $p+1$ bitのbit長を持つ。本実施例では p' bit = 2 bit = $p+1$ bitとする。

【0110】

ここで視差データ P_n (bit長 p)を埋め込み用データ P'_n (bit長 p')の変換を行う(ステップS19-5)。具体的には、 P_n を P'_n にコピーし、さらに視差がある画素は P'_n の最上位bitを立てる(ステップS19-53)。

【0111】

続いて輝度値データの下位bitを奥行き値データ P_n と置き換えるためのマスクを作成する(ステップS19-6)。マスク m_{RGB} は m_R 、 m_G 、 m_B より構成され、それぞれ a bit長のR成分、G成分、B成分用のマスクである。RGB成分各要素に埋め込めるデータ長をそれぞれ p'_R 、 p'_G 、 p'_B としデータ長の合計は p' bitと等価(図20)としたとき、 m_R 、 m_G 、 m_B はそれぞれ下位 p'_R 、 p'_G 、 p'_B bitが0、残りの上位bitが1で表される。また p'_R 、 p'_G 、 p'_B のbit長の関係は $p'_G > p'_R > p'_B$ かつ、 p'_G が上位bit側・ p'_B が下位bit側となるようにする(p'_R は中間)。これは人間の視感度の強さが緑、赤、青の順で弱くなることを利用し人間が知覚しにくい成分に、より多くのデータを埋め込むためである。本実施例では $p'_G = 0$ bit、 $p'_R = 0$ bit、 $p'_B = 2$ bitとした。

【0112】

ここで作成したマスクを用いて、輝度値データ M_{RGBn} の各々の画素 n に対応する輝度値の下位bitを埋め込み用データ P'_n と置き換える(ステップS19-8)。

10

20

30

40

50

【0113】

置き換えた値は埋め込みデータ Z_{RGBn} として出力される（ステップ S19-9）。なお下位 bit への視差の埋め込み方法は上記アルゴリズムに限定されることはなく、同様の結果を得られれば良い。なお視差データの埋め込み手順は上記に限定されるものではなく、同様の結果を得られれば良い。

【0114】

図21は本実施例でのエンコード・デコードの例を示す図である。入力データ21-1は輝度値データ ($a = 24 bit$) 21-11、視差データ ($p = 1 bit$) 21-12から構成される。視差データ21-12は図19に示した方法により輝度値データ21-11に埋め込まれ、埋め込み済み画像21-2を得る。視差データ21-12上で黒色で示される部分が視差量0、白色で示される部分が視差量1を示す。なお視差量の値が必ずしも実際の視差と一致する必要は無く、予め指定しておいたパラメータにより変換してもよい。本実施例では1視差量が画像上の10画素を示すものとして予めパラメータを決める。また輝度値データ21-12の視差量0を視差無しの状態として代用する場合は P_n から P'_n の変換 (S19-5) を省略し、 $P_n = P'_n$ としても良い。埋め込みデータ21-2を専用ビューアー21-3でデコードすることにより、出力データ21-4を得る。出力データは右目用画像21-41と左目用画像21-42から構成される。右目用画像21-41は埋め込み済み画像21-2と等価であり、また埋め込まれた視差量1につき10画素右に移動させることにより左目用画像21-42を得ることができる。本実施例では予め決めたパラメータより右目用画像から左目用画像の変換を行ったが、その逆に左目用画像から右目用画像の変換を行うことも容易である。専用ビューアーでデコードする際の情報はヘッダ部に埋め込む。具体的には、本実施例で用いる画像フォーマットを png 形式とし、ヘッダの付随的ブロック部にあるテキスト・データにパラメータ情報を埋め込む。付随的ブロックは通常エンコードはそれらを書く必要がなく、デコードはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコードでは解釈されない。またヘッダにパラメータを埋め込むのではなく、視差量と同様に画像に埋め込んで良い。具体的には、画面の上下左右の周囲の1ラインずつの輝度値データのいずれかの下位 bit をパラメータ記述部と判断する方法などがある。なお、本発明の画像フォーマットは上記に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

【0115】

以上の手順により、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

【0116】

[実施例10]

図22は本発明の第10の実施例（請求項8に記載の発明に対応する実施例）のフローを示す図である。以下、図22を用いて詳細な動作の説明を行う。

【0117】

まずカラー画像（横方向 x 画素、縦方向 y 画素）の輝度値データ T_n を読み込む（ステップ S22-1）。

【0118】

次に読み込んだカラー画像の輝度値を RGB 各成分に付いて $a \times 3 bit$ の画素ごとのデータ M_{RGBn} として割り振る（ステップ S22-2、S22-3）。 M_{RGBn} は M_R 、 M_G 、 M_B より構成され、それぞれが $a bit$ 長のカラー画像の R 成分、 G 成分、 B 成分にあたるデータである。このように一度割り振りを行うことにより、輝度値データにパレットが使用されている場合にも対応可能となる。

【0119】

次に視差データ P_n の読み込みを行う（ステップ S22-4）。視差データは輝度値データの各画素に対応する視差量を持つ $p bit$ のデータである。本実施例では $p bit$

= 1 bit とする。また埋め込み可能な bit 長を p' bit とし、 p' bit は p bit と等価または $p + 1$ bit の bit 長を持つ。

【0120】

ここで視差データ P_n (bit 長 p) から埋め込み用データ P'_n (bit 長 p') への変換を行う (ステップ S22-5)。具体的には、 P_n を P'_n にコピーし、さらに視差がある画素は P'_n の最上位 bit を立てる (ステップ S2-53)。

【0121】

続いて輝度値データの下位 bit を奥行き値データ P_n と置き換えるためのマスクを作成する (ステップ S22-5)。マスク m_{RGB} は m_R 、 m_G 、 m_B より構成され、それぞれ a bit 長の R 成分、G 成分、B 成分用のマスクである。RGB 成分各要素に埋め込めるデータ長をそれぞれ p'_R 、 p'_G 、 p'_B としデータ長の合計は p' bit と等価 (図 20) としたとき、 m_R 、 m_G 、 m_B はそれぞれ下位 p'_R 、 p'_G 、 p'_B bit が 0、残りの上位 bit が 1 で表される。また p'_R 、 p'_G 、 p'_B の bit 長の関係は $p'_G > p'_R > p'_B$ かつ、 p'_G が上位 bit 側・ p'_B が下位 bit 側となるようにする (p'_R は中間)。これは人間の視感度の強さが緑、赤、青の順で弱くなることを利用し人間が知覚しにくい成分に、より多くのデータを埋め込むためである。

【0122】

ここで作成したマスクを用いて、輝度値データ M_{RGBn} の各々の画素 n に対応する輝度値の下位 bit を埋め込み用データ P'_n と置き換える (ステップ S22-8)。

【0123】

次にオクルージョン部の検出を行う。検出方法を記述する前に、図 23 を用いてオクルージョン部について説明する。23-1 は輝度値データの例、23-2 は視差データの例を示している。視差データ 23-2 上で黒色で示される部分が視差量 0、白色で示される部分が視差量 1 を示す。なお視差量の値が必ずしも実際の視差と一致する必要は無く、デコード時には予め指定しておいたパラメータにより変換してもよい。本実施例では 1 視差量が画像上の 10 画素を示し、右方向に移動するものとして予めパラメータを決める。なおこのパラメータは必要な視差画像にあわせて任意に設定すればよい。23-3 はパラメータに従ってデコードされた元のデータ 23-1 に対する視差画像を示している。23-3 において白抜きの部分 23-31 は元のデータ 23-1 に変換元の画素が無かった部分 (画像の移動後にデータが空白となる部分)、つまり元のデータ 23-1 に対するオクルージョン部である。また 23-4 は元のデータ 23-1 において、視差画像 23-3 への変換時に移動した画素を示している。23-4 において白抜きの部分 23-41 は視差画像 23-3 に変換先の画素が無かった部分 (画像の移動後にかくれる部分)、つまり視差画像 23-3 に対するオクルージョン部である。ステップ S22-9 で検出するオクルージョン部は 23-31 と 23-41 によって説明した 2 種類である。具体的に検出を行う場合、単純に移動元の有無および移動先の有無を調べればよい。移動元がない画素の領域は 23-31 部、移動先が無い画素の領域は 23-41 部に相当する。

【0124】

次に予め用意された背景画像からオクルージョン部の輝度値データを読み込む (ステップ S22-10)。ただしここで輝度値データを読み込むオクルージョン部は 23-31 部に相当する領域だけである。

【0125】

この読み込んだ輝度値データをステップ 22-9 で検出した 23-41 部および視差が無い画素 (ステップ S22-53 で P'_n の最上位 bit を 0 とした領域) の下位 bit すなわち P'_n のうち判定用 bit を除いた部分に埋め込む (ステップ S22-11)。埋め込み順は左から右へとライン単位で検索を行い検出された 23-31 部の画素ごとに視差画像に変換後の輝度値を分割し、同様に左から右へとライン単位で検索を行い検出された 23-41 部および視差が無い画素の下位 bit にシーケンシャルに埋め込んでいく。このようにシーケンシャルに埋め込みを行うことにより、画素の位置情報を埋め込まなくても、同様のシーケンシャルな検索によりデコード時に正しい画素位置に輝度値データを

10

20

30

40

50

復元できる。

【0126】

以上の手順により下位 *bit* を置き換えたものを埋め込みデータ Z_{RGBn} として出力する (ステップ S22-12)。

【0127】

なお下位 *bit* への視差の埋め込み方法の手順は上記アルゴリズムに限定されることはなく、同様の結果を得られれば適時順序の入れ替えなどを行っても良い。

【0128】

上記、画像をデコードする際の情報はヘッダー部に埋め込む。具体的には、本実施例で用いる画像フォーマットを *png* 形式とし、ヘッダーの付随的ブロック部にあるテキスト・データにパラメータ情報を埋め込む。付随的ブロックは通常エンコーダはそれらを書く必要がなく、デコーダはそれを無視してよいという意味でオプションであるため、従来の画像ビューアーの持つデコーダでは解釈されずにそのまま表示される。またヘッダーにパラメータを埋め込むのではなく、視差量と同様に画像に埋め込んで良い。具体的には、画面の上下左右の周囲の1ラインずつの輝度値データのいずれかの下位 *bit* をパラメータ記述部と判断する方法などがある。

10

【0129】

なお本発明の画像フォーマットは上記に限定されるものではなく、データの可逆性を保証するものであれば良い。同様にヘッダーの、データを埋め込むブロックも従来の画像ビューアーのデコードの邪魔にならない位置であればいずれの位置でも良い。

20

【0130】

以上の手順により、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

【0131】

[実施例11]

本実施例は実施例2、3、5、6に対応する装置に関する。図24に本実施例の装置の構成図を示す。本実施例の立体情報埋め込み装置24-1は、輝度値データ入力部24-2、奥行き値データ入力部24-3、埋め込みデータ作成部24-4、マスク作成部24-5、下位 *bit* 置き換え部24-6、およびデータ出力部24-7を備える。実施例2、3に対応する立体情報埋め込み装置24-1はモノクロデータ処理を行う24-2~24-7を備え、実施例5、6に対応する立体情報埋め込み装置24-1はカラーデータ処理を行う24-2~24-7を備える。立体情報埋め込み装置24-1の輝度値データ入力部24-2には輝度値データ24-8が入力され、奥行き値データ入力部24-3には奥行き値データ24-9が入力され、データ出力部24-7から立体画像データ24-10が出力される。

30

【0132】

次に立体情報埋め込み装置24-1の動作を説明する。輝度値データ入力部24-2が画像の輝度値データ24-8を読み込み、奥行き値データ入力部24-3が奥行き値データ24-9を読み込む。奥行き値データ入力部24-3の出力を用いて埋め込みデータ作成部24-4が埋め込みデータを作成する。マスク作成部24-5が輝度値データの下位 *bit* を埋め込みデータと置き換えるためのマスクを作成する。輝度値データ入力部24-2、埋め込みデータ作成部24-4、およびマスク作成部24-5の出力を用いて下位 *bit* 置き換え部24-6が各々の画素に対応する輝度値の下位 *bit* を埋め込みデータと置き換え、置き換えた値をデータ出力部24-7が立体画像データ24-10として出力する。なお、詳細な動作方法は実施例2、3、5、6に記載したとおりである。

40

【0133】

以下、前記実施例の変形例について説明する。本変形例は実施例1、4に対応する装置に関するものである。本変形例の装置は図24の立体情報埋め込み装置24-1から埋め込みデータ作成部24-4を省略したものである。実施例1に対応する装置はモノクロデータ処理を行う輝度値データ入力部24-2、奥行き値データ入力部24-3、マスク作成

50

部 2 4 - 5、下位 b i t 置き換え部 2 4 - 6、およびデータ出力部 2 4 - 7 を備え、奥行き値データ入力部 2 4 - 3 の出力が下位 b i t 置き換え部 2 4 - 6 に入力される。実施例 4 に対応する装置はカラーデータ処理を行う輝度値データ入力部 2 4 - 2、奥行き値データ入力部 2 4 - 3、マスク作成部 2 4 - 5、下位 b i t 置き換え部 2 4 - 6、およびデータ出力部 2 4 - 7 を備え、奥行き値データ入力部 2 4 - 3 の出力が下位 b i t 置き換え部 2 4 - 6 に入力される。

【 0 1 3 4 】

次に本変形例の立体情報埋め込み装置の動作を説明する。輝度値データ入力部 2 4 - 2 が画像の輝度値データ 2 4 - 8 を読み込み、奥行き値データ入力部 2 4 - 3 が奥行き値データ 2 4 - 9 を読み込む。マスク作成部 2 4 - 5 が輝度値データの低位 b i t を埋め込みデータと置き換えるためのマスクを作成する。輝度値データ入力部 2 4 - 2、奥行き値データ入力部 2 4 - 3、およびマスク作成部 2 4 - 5 の出力を用いて低位 b i t 置き換え部 2 4 - 6 が各々の画素に対応する輝度値の低位 b i t を埋め込みデータと置き換え、置き換えた値をデータ出力部 2 4 - 7 が立体画像データ 2 4 - 1 0 として出力する。なお、詳細な動作方法は実施例 1、4 に記載したとおりである。

10

【 0 1 3 5 】

[実施例 1 2]

本実施例は実施例 8、1 0 に対応する装置に関する。図 2 5 に本実施例の装置の構成図を示す。本実施例の立体情報埋め込み装置 2 5 - 1 は、輝度値データ入力部 2 5 - 2、視差データ入力部 2 5 - 3、埋め込みデータ作成部 2 5 - 4、マスク作成部 2 5 - 5、第 1 の低位 b i t 置き換え部 2 5 - 6、オクルージョン検出部 2 5 - 7、背景輝度値データ入力部 2 5 - 8、輝度値データ分割部 2 5 - 9、第 2 の低位 b i t 置き換え部 2 5 - 1 0、およびデータ出力部 2 5 - 1 1 を備える。実施例 8 に対応する立体情報埋め込み装置 2 5 - 1 はモノクロデータ処理を行う 2 5 - 2 ~ 2 5 - 1 1 を備え、実施例 1 0 に対応する立体情報埋め込み装置 2 5 - 1 はカラーデータ処理を行う 2 5 - 2 ~ 2 5 - 1 1 を備える。立体情報埋め込み装置 2 5 - 1 の輝度値データ入力部 2 5 - 2 には輝度値データ 2 5 - 1 2 が入力され、視差データ入力部 2 5 - 3 には視差データ 2 5 - 1 3 が入力され、背景輝度値データ入力部 2 5 - 8 には背景輝度値データ 2 5 - 1 4 が入力され、データ出力部 2 5 - 1 1 から立体画像データ 2 5 - 1 5 が出力される。

20

【 0 1 3 6 】

次に立体情報埋め込み装置 2 5 - 1 の動作を説明する。輝度値データ入力部 2 5 - 2 が画像の輝度値データ 2 5 - 1 2 を読み込み、視差データ入力部 2 5 - 3 が視差データ 2 5 - 1 3 を読み込む。視差データ入力部 2 5 - 3 の出力を用いて埋め込みデータ作成部 2 5 - 4 が埋め込みデータを作成する。マスク作成部 2 5 - 5 が輝度値データの低位 b i t を埋め込みデータと置き換えるためのマスクを作成する。輝度値データ入力部 2 5 - 2、埋め込みデータ作成部 2 5 - 4、およびマスク作成部 2 5 - 5 の出力を用いて第 1 の低位 b i t 置き換え部 2 5 - 6 が各々の画素に対応する輝度値の低位 b i t を埋め込みデータと置き換える。第 1 の低位 b i t 置き換え部 2 5 - 6 の出力からオクルージョン検出部 2 5 - 7 がオクルージョン部を検出する。一方、背景輝度値データ 2 5 - 1 4 から背景輝度値データ入力部 2 5 - 8 がオクルージョン部の輝度値データを読み込む。背景輝度値データ入力部から出力された輝度値データを輝度値データ分割部 2 5 - 9 が上位 b i t から指定された b i t 数だけ複数個に分割する。オクルージョン検出部 2 5 - 7、マスク作成部 2 5 - 5、および輝度値データ分割部 2 5 - 9 の出力を用いて第 2 の低位 b i t 置き換え部 2 5 - 1 0 が視差のない画像の低位 b i t に分割された輝度値を順次置き換えていき、置き換えたものをデータ出力部 2 5 - 1 1 が立体画像データ 2 5 - 1 5 として出力する。なお、詳細な動作方法は実施例 8、1 0 に記載したとおりである。

30

40

【 0 1 3 7 】

以下、前記実施例の変形例について説明する。本変形例は実施例 7、9 に対応する装置に関するものである。本変形例の装置は図 2 5 の立体情報埋め込み装置 2 5 - 1 からオクルージョン検出部 2 5 - 7、背景輝度値データ入力部 2 5 - 8、輝度値データ分割部 2 5 -

50

9、および第2の下位bit置き換え部25-10を省略したものである。実施例7に対応する装置はモノクロデータ処理を行う輝度値データ入力部25-2、視差データ入力部25-3、埋め込みデータ作成部25-4、マスク作成部25-5、第1の下位bit置き換え部25-6(以下、「下位bit置き換え部25-6」という)、およびデータ出力部25-11を備え、下位bit置き換え部25-6の出力がデータ出力部25-11に入力される。実施例9に対応する装置はカラーデータ処理を行う輝度値データ入力部25-2、視差データ入力部25-3、埋め込みデータ作成部25-4、マスク作成部25-5、下位bit置き換え部25-6、およびデータ出力部25-11を備え、下位bit置き換え部25-6の出力がデータ出力部25-11に入力される。

【0138】

10

次に本変形例の立体情報埋め込み装置25-1の動作を説明する。輝度値データ入力部25-2が画像の輝度値データ25-12を読み込み、視差データ入力部25-3が視差データ25-13を読み込む。視差データ入力部25-3の出力を用いて埋め込みデータ作成部25-4が埋め込みデータを作成する。マスク作成部25-5が輝度値データの低位bitを埋め込みデータと置き換えるためのマスクを作成する。輝度値データ入力部25-2、埋め込みデータ作成部25-4、およびマスク作成部25-5の出力を用いて下位bit置き換え部25-6が各々の画素に対応する輝度値の低位bitを埋め込みデータと置き換え、置き換えた値をデータ出力部25-11が立体画像データ25-15として出力する。なお、詳細な動作方法は実施例7、9に記載したとおりである。

【0139】

20

本発明の装置はコンピュータとプログラムによっても実現でき、プログラムを記録媒体に記録することも、ネットワークを通じて提供することも可能である。

【0140】

以上、本発明者によってなされた発明を、前記実施の形態に基づき具体的に説明したが、本発明は、前記実施の形態に限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは勿論である。

【0141】

【発明の効果】

本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば、下記の通りである。

30

【0142】

本発明の請求項1の発明によれば、モノクロ画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく奥行き値を埋め込むことが可能となる。

【0143】

請求項2の発明によれば、モノクロ画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく高bitの奥行き値を埋め込むことが可能となる。

【0144】

請求項3の発明によれば、カラー画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく奥行き値を埋め込むことが可能となる。

40

【0145】

請求項4の発明によれば、カラー画像において従来の画像データと互換性を保ちながら、データ量をヘッダーを除いて全く増やすことなく高bitの奥行き値を埋め込むことが可能となる。

【0146】

請求項5の発明によれば、モノクロ画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

【0147】

請求項6の発明によれば、モノクロ画像において従来の画像データと互換性を保ちながら

50

、データ量を増やすことなく、視差およびオクルージョン情報を埋め込むことが可能となる。

【0148】

請求項7の発明によれば、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差を埋め込むことが可能となる。

【0149】

請求項8の発明によれば、カラー画像において従来の画像データと互換性を保ちながら、データ量を増やすことなく、視差およびオクルージョン情報を埋め込むことが可能となる。

【図面の簡単な説明】

10

【図1】本発明の第1の実施例のフローを示す図である。

【図2】本発明の第1の実施例の埋め込み例を示す図（ディスプレイ上に表示された中間調画像の写真を含む、2bitの奥行きデータはディサ処理）である。

【図3】本発明の第1の実施例において、埋め込みデータをビューアーで表示する例を示す図（ディスプレイ上に表示された中間調画像の写真を含む、2bitの奥行きデータはディサ処理）である。

【図4】本発明の第2の実施例のフローを示す図である。

【図5】本発明の第2の実施例において、デコード・エンコードの例を示す図（ディスプレイ上に表示された中間調画像の写真を含む、2bitの奥行きデータはディサ処理）である。

20

【図6】本発明の第3の実施例のフローを示す図である。

【図7】本発明の第4の実施例のフローを示す図である。

【図8】RGB成分各要素に埋め込めるデータ長の説明図である。

【図9】本発明の第4の実施例の埋め込み例を示す図（ディスプレイ上に表示された中間調画像の写真を含む）である。

【図10】本発明の第4の実施例において、埋め込みデータをビューアーで表示する例を示す図（ディスプレイ上に表示された中間調画像の写真を含む）である。

【図11】本発明の第5の実施例のフローを示す図である。

【図12】RGB成分各要素に埋め込めるデータ長の説明図である。

【図13】本発明の第5の実施例において、デコード・エンコードの例を示す図（ディスプレイ上に表示された中間調画像の写真を含む）である。

30

【図14】本発明の第6の実施例のフローを示す図である。

【図15】本発明の第7の実施例のフローを示す図である。

【図16】本発明の第7の実施例において、デコード・エンコードの例を示す図（ディスプレイ上に表示された中間調画像の写真を含む）である。

【図17】本発明の第8の実施例のフローを示す図である。

【図18】オクルージョン部の説明図である（ディスプレイ上に表示された中間調画像の写真を含む）。

【図19】本発明の第9の実施例のフローを示す図である。

【図20】RGB成分各要素に埋め込めるデータ長の説明図である。

40

【図21】本発明の第9の実施例において、デコード・エンコードの例を示す図（ディスプレイ上に表示された中間調画像の写真を含む）である。

【図22】本発明の第10の実施例のフローを示す図である。

【図23】オクルージョン部の説明図（ディスプレイ上に表示された中間調画像の写真を含む）である。

【図24】本発明の第11の実施例の装置の構成図である。

【図25】本発明の第12の実施例の装置の構成図である。

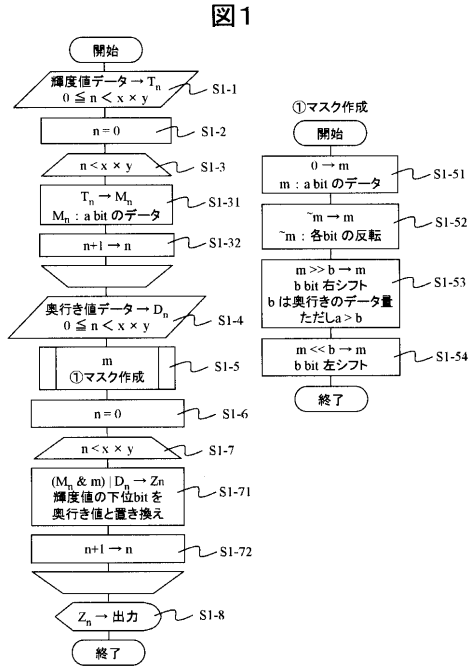
【符号の説明】

2 - 1 ... 8bitの輝度値データ、2 - 2 ... 2bitの奥行き値データ、2 - 3 ... 埋め込み済みデータの例、3 - 1 ... 埋め込みデータ、3 - 2 1 ... 従来型ビューアー、3 - 2 2 ...

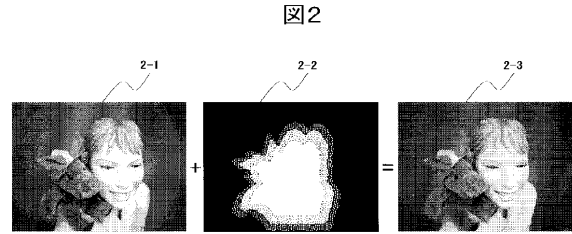
50

埋め込みデータを従来型ビューアーで表示した例、3-31...専用ビューアー、3-32...埋め込みデータを専用ビューアーで表示した例、5-1...入力データ、5-11...8bitの輝度値データ、5-12...8bitの奥行き値データ、5-2...変換した奥行き値データ、5-3...埋め込み済みデータ、5-4...専用ビューアー、5-5...出力データ、5-51...デコードされた輝度値データ、5-52...デコードされた奥行き値データ、9-1...RGB成分各8bitの輝度値データ、9-2...4bitの奥行き値データ、9-3...埋め込み済みデータの例、10-1...埋め込みデータ、10-21...従来型ビューアー、10-22...埋め込みデータを従来型ビューアーで表示した例、10-31専用ビューアー、10-32...埋め込みデータを専用ビューアーで表示した例、13-1...入力データ、13-11...RGB成分各8bitの輝度値データ、13-2...8bitの奥行き値データ、13-3...埋め込み済みデータの例、13-4...専用ビューアー、13-5...出力データ、13-51...デコードされた輝度値データ、13-52...デコードされた奥行き値データ、16-1...入力データ、16-11...輝度値データ、16-12...視差データ、16-2...埋め込み済みデータ、16-3...専用ビューアー、16-4...出力データ、16-41...右目用画像、16-42...左目用画像、18-1...輝度値データ、18-2...視差データ、18-3...視差画像、18-31...視差画像に対する元データのオクルージョン部、18-4...対応点のある画像領域、18-41...元データに対する視差画像のオクルージョン部、21-1...入力データ、21-11...輝度値データ、21-12...視差データ、21-2...埋め込み済みデータ、21-3...専用ビューアー、21-4...出力データ、21-41...右目用画像、21-42...左目用画像、23-1...輝度値データ、23-2...視差データ、23-3...視差画像、23-31...視差画像に対する元データのオクルージョン部、23-4...対応点のある画像領域、23-41...元データに対する視差画像のオクルージョン部、24-1...立体情報埋め込み装置、24-2...輝度値データ入力部、24-3...奥行き値データ入力部、24-4...埋め込みデータ作成部、24-5...マスク作成部、24-6...下位bit置き換え部、24-7...データ出力部、24-8...輝度値データ、24-9...奥行き値データ、24-10...立体画像データ、25-1...立体情報埋め込み装置、25-2...輝度値データ入力部、25-3...視差データ入力部、25-4...埋め込みデータ作成部、25-5...マスク作成部、25-6...第1の下位bit置き換え部、25-7...オクルージョン検出部、25-8...背景輝度値データ、25-9...輝度値データ分割部、25-10...第2の下位bit置き換え部、25-11...データ出力部、25-12...輝度値データ、25-13...視差データ、25-14...背景輝度値データ、25-15...立体画像データ

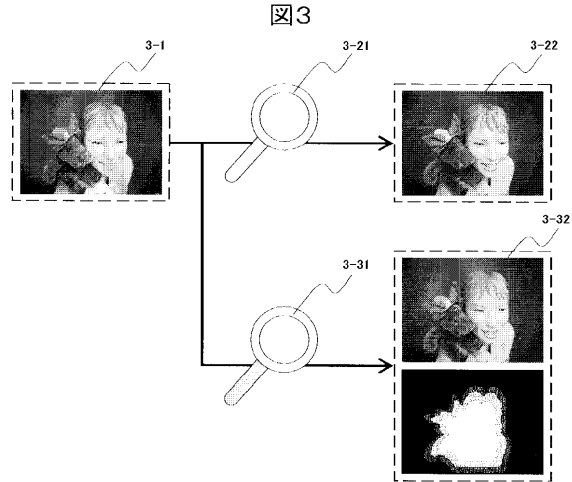
【 図 1 】



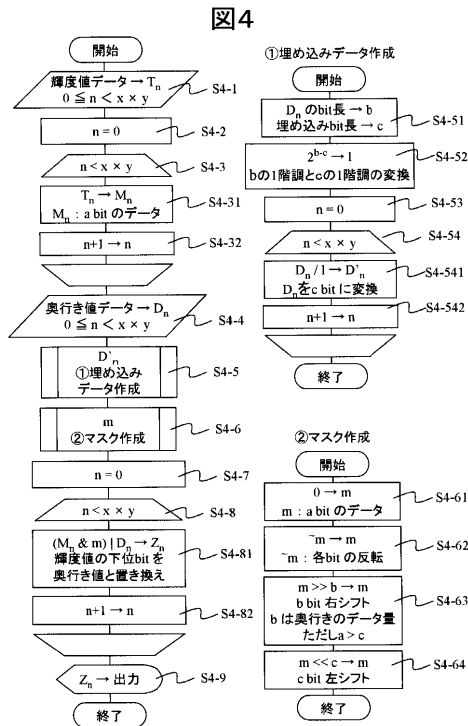
【 図 2 】



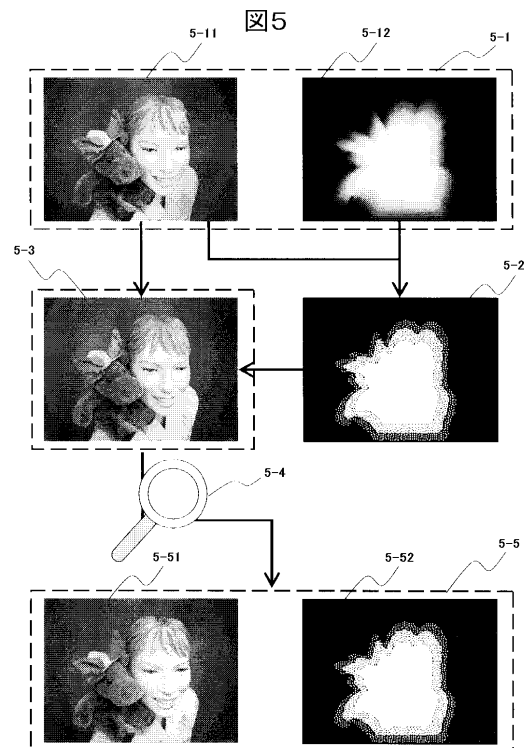
【 図 3 】



【 図 4 】

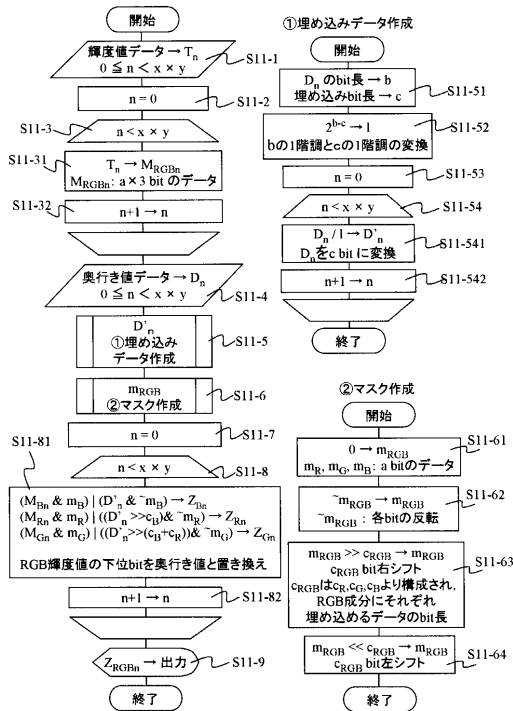


【 図 5 】



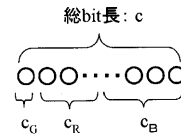
【 図 1 1 】

図11



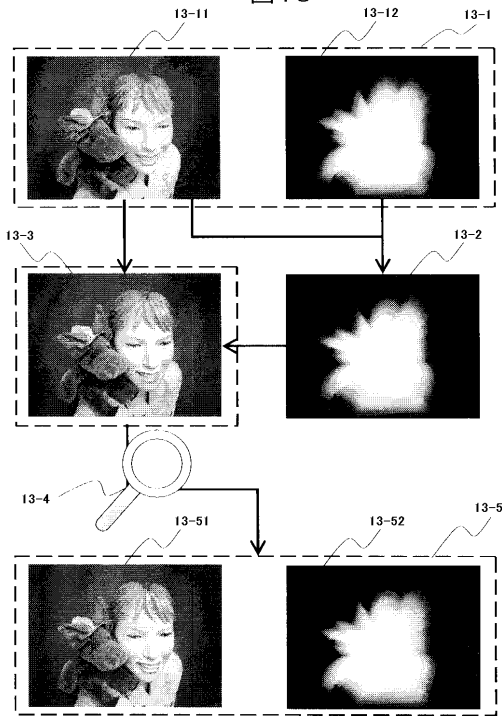
【 図 1 2 】

図12



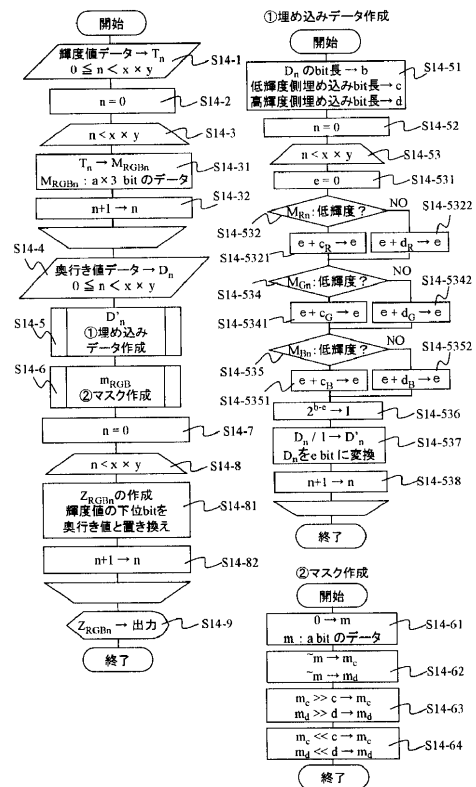
【 図 1 3 】

図13

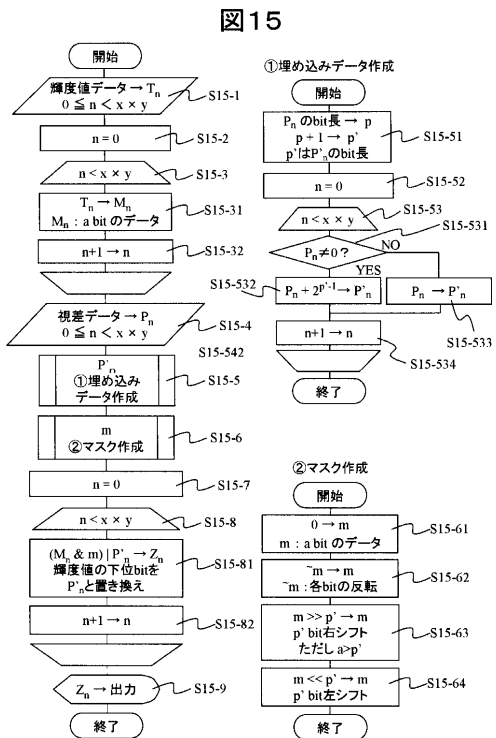


【 図 1 4 】

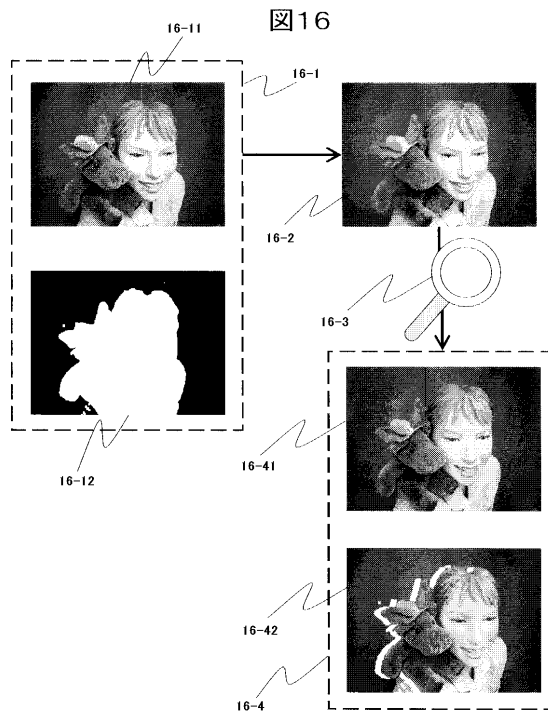
図14



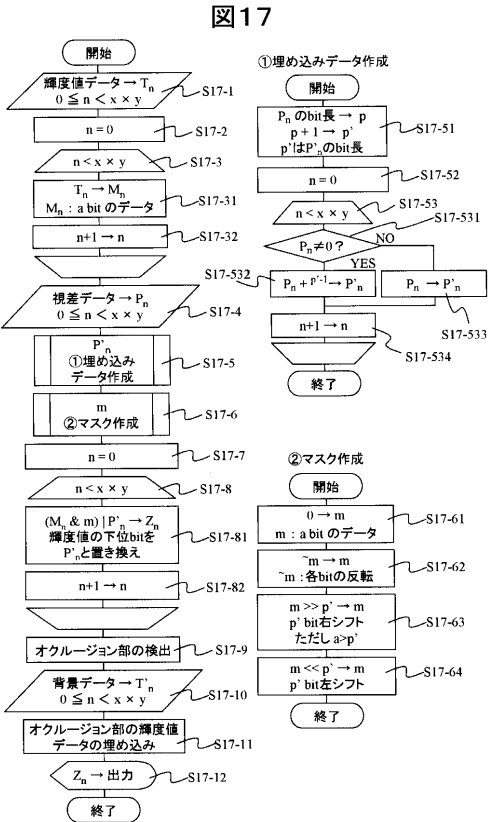
【 図 15 】



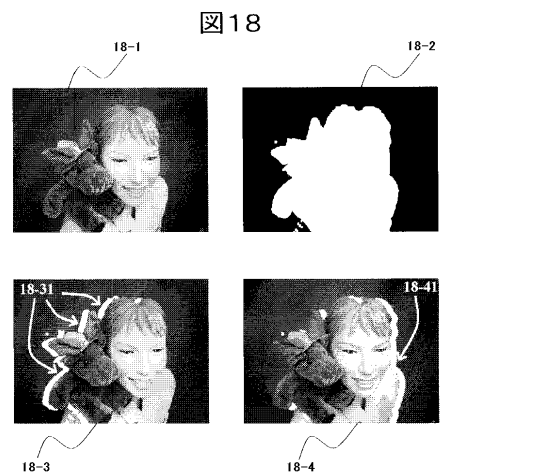
【 図 16 】



【 図 17 】

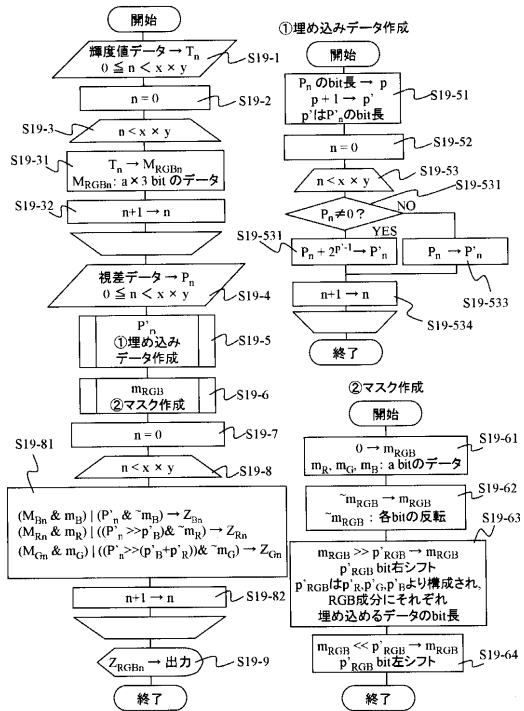


【 図 18 】



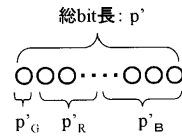
【図19】

図19



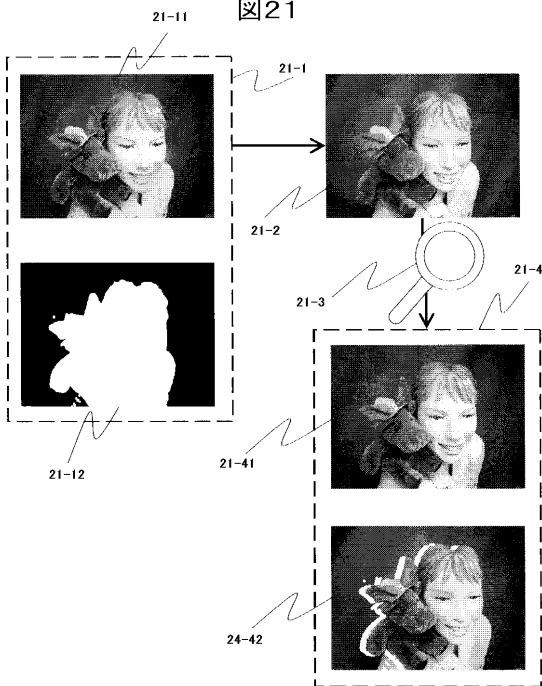
【図20】

図20



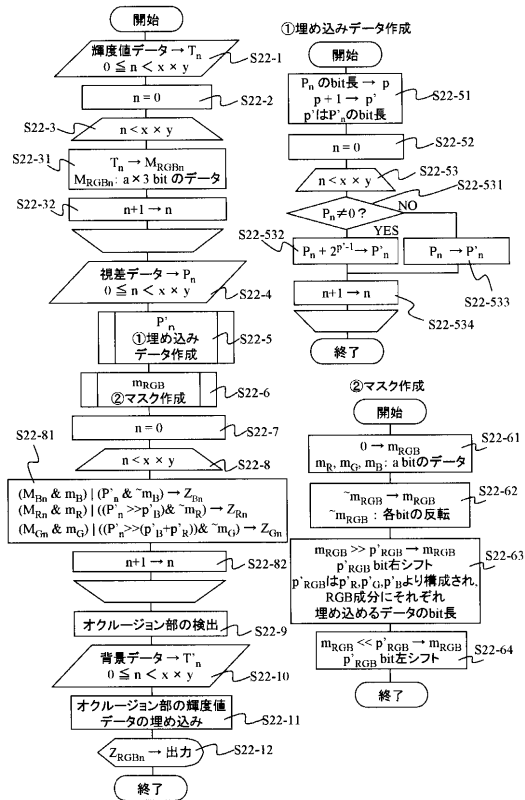
【図21】

図21

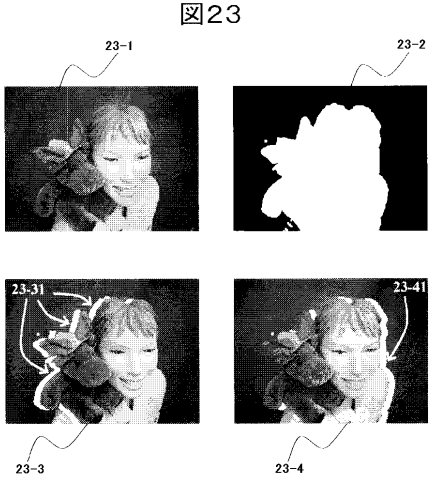


【図22】

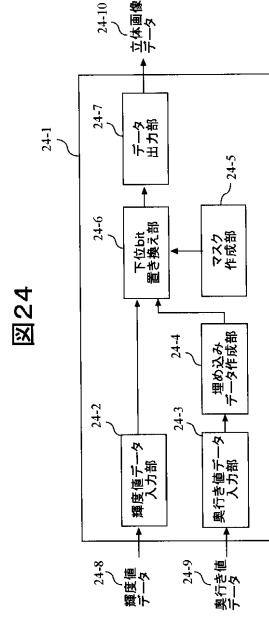
図22



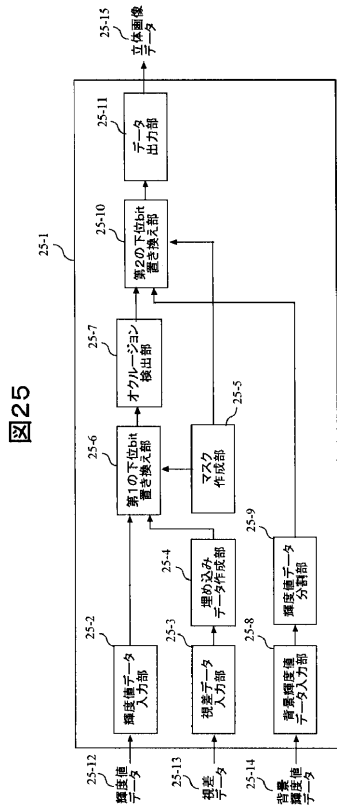
【 図 2 3 】



【 図 2 4 】



【 図 2 5 】



フロントページの続き

Fターム(参考) 5B057 CA01 CA02 CA08 CA16 CB01 CB02 CB08 CB16 CB19 CE08
CE17 CH01 CH11
5C076 AA14 AA26 BA06