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

特許第3943931号

(P3943931)

(45)発行日　平成19年7月11日(2007.7.11)　(24)登録日　平成19年4月13日(2007.4.13)

(51)Int.Cl.			F I		
H04N	1/387	(2006.01)	H04N	1/387	
G06T	1/00	(2006.01)	G06T	1/00	500B
G09C	1/00	(2006.01)	G09C	1/00	640D
G09C	5/00	(2006.01)	G09C	5/00	

請求項の数 14　(全 19 頁)

(21)出願番号　特願2001-397873(P2001-397873)
(22)出願日　平成13年12月27日(2001.12.27)
(65)公開番号　特開2003-92677(P2003-92677A)
(43)公開日　平成15年3月28日(2003.3.28)
審査請求日　平成16年11月18日(2004.11.18)
(31)優先権主張番号　特願2001-211186(P2001-211186)
(32)優先日　平成13年7月11日(2001.7.11)
(33)優先権主張国　日本国(JP)

(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
(72)発明者　岩村　恵市
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

最終頁に続く

(54)【発明の名称】画像処理方法及び装置とそのプログラム及び記憶媒体

(57)【特許請求の範囲】

【請求項１】

順序付けられた複数の画素から構成される画像データに対して自己同期性をもつ第１パターンを生成する第１パターン生成手段と、

前記画像データに依存した第２パターンを生成する第２パターン生成手段と、

前記第１パターンと前記第２パターンとから埋め込みパターンを生成する埋め込みパターン生成手段と、

前記埋め込みパターンを前記画素に埋め込むパターン埋め込み手段と、

を有することを特徴とする画像処理装置。

【請求項２】

前記第２パターン生成手段は、初期値を基に擬似乱数を発生する擬似乱数発生手段を有し、前記画像データと前記擬似乱数とに基づいて前記第２パターンを生成することを特徴とする請求項１に記載の画像処理装置。

【請求項３】

前記パターン埋め込み手段は、前記画素のＬＳＢに埋め込むことを特徴とする請求項１又は２に記載の画像処理装置。

【請求項４】

前記第１パターン生成手段は、画像毎に異なる初期値を用いて、自己同期パターン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項に記載の画像処理装置。

【請求項５】

更に、前記画像の拡大縮小及び回転を補正する補正手段を有することを特徴とする請求項１乃至４のいずれか１項に記載の画像処理装置。
【請求項６】
前記順序付けられた複数の画素から第３パターンを抽出する第３パターン抽出手段と、
前記第３パターン抽出手段により抽出された前記第３パターンの自己同期性を検査する検査手段とを更に有することを特徴とする請求項１に記載の画像処理装置。
【請求項７】
順序付けられた複数の画素から構成される画像データに対して自己同期性をもつ第１パターンを生成する第１パターン生成工程と、
前記画像データに依存した第２パターンを生成する第２パターン生成工程と、
前記第１パターンと前記第２パターンとから埋め込みパターンを生成する埋め込みパターン生成工程と、
前記埋め込みパターンを前記画素に埋め込むパターン埋め込み工程と、
を有することを特徴とする画像処理方法。
【請求項８】
前記第２パターン生成工程は、初期値を基に擬似乱数を発生し、前記画像データと前記擬似乱数とに基づいて前記第２パターンを生成することを特徴とする請求項７に記載の画像処理方法。
【請求項９】
前記パターン埋め込み工程では、前記画素のＬＳＢに埋め込むことを特徴とする請求項７又は８に記載の画像処理方法。
【請求項１０】
前記第１パターン生成工程は、画像毎に異なる初期値を用いて、自己同期パターンを生成することを特徴とする請求項７乃至９のいずれか１項に記載の画像処理方法。
【請求項１１】
更に、前記画像の拡大縮小及び回転を補正する補正工程を有することを特徴とする請求項７乃至１０のいずれか１項に記載の画像処理方法。
【請求項１２】
前記順序付けられた複数の画素から第３パターンを抽出する第３パターン抽出工程と、
前記第３パターン抽出工程で抽出された前記第３パターンの自己同期性を検査する検査工程とを更に有することを特徴とする請求項７に記載の画像処理方法。
【請求項１３】
請求項１乃至６のいずれか１項に記載の画像処理装置の機能をコンピュータに実行させるためのプログラム。
【請求項１４】
請求項１３に記載のプログラムを格納し、コンピュータが読み取り可能なコンピュータ可読記憶媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、画像データの処理に関するもので、特にデジタル画像データのようなデジタルデータの改竄位置を検出するための画像処理方法及び装置、並びにその方法を実行するプログラム、及びそのプログラムを記憶した記憶媒体に関するものである。
【０００２】
【従来の技術】
近年、コンピュータやインターネットの普及に伴い、従来の銀塩写真や紙の書類などに代わって、情報をデジタル化しデジタル画像として利用する形態が一般化しつつある。更に、目覚しい画像処理技術の進歩により、デジタル画像の編集・改竄は、フォトレタッチツール等を使用すれば容易に行うことが可能になった。このため、デジタル画像の原本性（オリジナリティ）は従来の銀塩写真や紙の書類等と比較して低く、証拠としての能力に

乏しいという問題点が生じてきている。従来、写真画像は、例えば保険会社などでは事故の証拠写真として用いられたり、建設会社などにおいては、建築現場の進捗状況の記録として用いられたりと、その証拠能力、即ち、その原本性は重要な役割を果たしてきている。しかし、上述したように、情報のデジタル化によってその証拠性が失われることは大きな問題である。

【０００３】

一般に、デジタル画像の原本性の保証は、米国特許第５４９９２９４号に示されるように、デジタル画像のハッシュ値に公開鍵暗号を用いた電子署名を作成することによって実現されるが、この手法では改竄の有無は分かっても、その改竄位置の検出まではできなかった。

【０００４】

それに対して、ある特定のパターンを画像全体に電子透かしとして埋め込み、その画像に対して改竄や編集が行われると、その埋め込まれたパターンが破壊されて、それにより改竄位置を検出する手法が知られている。その一例としては、ＬＳＢにあるスタンプ画像と呼ばれる特定画像を埋め込む手法であり、例えば特開２００１－２４８７６号公報には、秘密の鍵情報から生成される擬似乱数によって画像中のスタンプ画像の埋め込み位置を特定する手法が提案されている。

【０００５】

【発明が解決しようとする課題】

このような従来の電子透かしによる改竄位置の検出手法は、そのアルゴリズムや埋め込みパターンが知られないことを安全性の根拠としている。従って、そのアルゴリズムや埋め込みパターンが知られてしまった場合には、例え改竄が行われても、その改竄が行われていないようにする、即ち、デジタル画像の偽造を行うことが可能である。つまり、ＬＳＢにスタンプ画像を埋め込む手法では、そのアルゴリズムを知った人物は、まずＬＳＢのスタンプ画像を抽出して保存し、その画像を改竄した後で、最初に保存したスタンプ画像を読み出して再び、その改竄後の画像データのＬＳＢに埋め込むことにより、改竄が検知されない偽造が可能になる。

【０００６】

また、特開２００１－２４８７６号公報に示されるような、擬似乱数によってスタンプ画像の埋め込み位置を定める手法は、その擬似乱数に埋め込み位置が依存するので、上述のＬＳＢに埋め込む手法よりは安全であるが、原画像と電子透かし画像の差分をとるなどによって、一旦そのスタンプ画像の埋め込み位置が知られてしまうと同様の偽造が可能になる。

【０００７】

一般に、電子透かしの安全性は、そのアルゴリズムが秘密であることを前提とする場合が多く、そのアルゴリズムが知られても安全であると言える手法は今まで提案されていないと言っても過言ではない。

【０００８】

また、従来の電子透かしによる改竄位置の検出手法では、スタンプ画像を埋め込んでスタンプ画像を保存し、その保存したスタンプ画像と抽出したスタンプ画像を比較することによって改竄位置を検出している。このようなスタンプ画像は２次元データであるのでデータ量が多く、それを保存するために大きなメモリ容量が必要となり効率的でない。

【０００９】

更に従来手法において、同じ鍵と同じスタンプ画像を用いて埋め込み処理された複数の異なる画像を切り貼りする場合、安全な手法は提案されていない。例えば、同じサイズの複数の異なる画像を、それぞれ同じ大きさのブロックに分解し、同じ位置にあるブロックを入れ替えることにより１つの画像を合成する攻撃を考える。このとき、各画像が同じ鍵、同じスタンプ画像を用いていれば、改竄は検出できず、この攻撃は成功する。このような攻撃に対して、画像毎に異なる鍵又は異なるスタンプ画像を用いて埋め込み処理をすることが考えられる。しかし、画像毎に異なる鍵又は異なるスタンプ画像を用いることは電

子透かしのアプリケーションとして使い易くない。それは、改竄されている疑いがある画像が発見された場合、まず最初に、その画像がどの鍵又はどのスタンプ画像で処理されているかを特定する必要があるためである。検証対象の画像が大きく改竄されていて原画像が推定しにくい場合、又は異なる鍵で処理された画像が合成されている場合、どの画像が原画像であるかを特定することは困難である。即ち、どの鍵又はどのスタンプ画像を用いてよいか分からない場合が出てくる。

【００１０】

本発明は上記従来例に鑑みてなされたもので、電子透かし手法のアルゴリズムや埋め込みパターンが知られたとしても、確実に改竄位置を検出できる画像処理方法及び装置とそのプログラム及び記憶媒体を提供することを目的とする。

【００１１】

また本発明の目的は、スタンプ画像を保存する必要がなく、かつ同じ鍵を用いて異なるデータに埋め込み処理を行っても、確実にそのデータにおける改竄位置を検出できる画像処理方法及び装置とそのプログラム及び記憶媒体を提供することにある。

【００１２】

また本発明の他の目的は、スタンプ画像を保存する必要がなく、かつ、数パターン分のずれや画像の挿入、削除を検出できる画像処理方法及び装置とそのプログラム及び記憶媒体を提供することにある。

【００１３】

【課題を解決するための手段】

上記目的を達成するために本発明の画像処理装置は以下のような構成を備える。即ち、

順序付けられた複数の画素から構成される画像データに対して自己同期性をもつ第１パターンを生成する第１パターン生成手段と、

前記画像データに依存した第２パターンを生成する第２パターン生成手段と、

前記第１パターンと前記第２パターンとから埋め込みパターンを生成する埋め込みパターン生成手段と、

前記埋め込みパターンを前記画素に埋め込むパターン埋め込み手段と、

を有することを特徴とする。

【００１９】

上記目的を達成するために本発明の画像処理方法は以下のような工程を備える。即ち、

順序付けられた複数の画素から構成される画像データに対して自己同期性をもつ第１パターンを生成する第１パターン生成工程と、

前記画像データに依存した第２パターンを生成する第２パターン生成工程と、

前記第１パターンと前記第２パターンとから埋め込みパターンを生成する埋め込みパターン生成工程と、

前記埋め込みパターンを前記画素に埋め込むパターン埋め込み工程と、

を有することを特徴とする。

【００２５】

【発明の実施の形態】

［実施の形態１］

図１は、本発明の実施の形態１に係る画像埋め込み処理を説明するための概念図である。

【００２６】

ここでは原画像Ｉ（ｉ,ｊ）をＭ×Ｎ画素からなる多値画像（ここでは、１画素が８ビットの多値画像として説明する）とする。また、下記において、◎はＥＸＯＲ（排他的論理和）を表す。但し、図１における処理１０３以下の処理は、ｉ＝０，ｊ＝０からｉ＝Ｍ－１，ｊ＝Ｎ－１まで、各画素毎に行われる。

【００２７】

＜埋め込み処理（図１）＞

まず処理１０１で、画像毎に定める値ｋｉを初期値として自己同期パターンＣ（ｉ,ｊ）を

生成する。

【００２８】

次に処理１０２で、鍵ｋを初期値として擬似乱数を生成し、８ビットデータを入力して１ビットデータを出力するルックアップテーブルＬＵＴ（）を作成する。これは各テーブルのアドレス順に、各アドレスに１ビットずつ、その生成した擬似乱数を割り当てていくことによって実現される。

【００２９】

次に処理１０３で、画像Ｉ（i,j）のＬＳＢを除いた画像Ｉ7（i,j）からＢ（i,j）＝ＬＵＴ（Ｉ7（i,j））を計算する。ここで、ＬＵＴ（Ｉ7（i,j））は、画像Ｉ7（i,j）がルックアップテーブルＬＵＴ（）に入力されたとき、それに対応して出力される１ビットの乱数値を示している。

【００３０】

次に処理１０４で、画像Ｉ（i,j）のＬＳＢに｛Ｂ（i,j）◎Ｃ（i,j）｝を埋め込み、電子透かしが埋め込まれた画像Ｉ'（i,j）を生成する。

【００３１】

この電子透かしの埋め込み処理によって得られる電子透かしが埋め込まれた画像Ｉ'（i,j）は、画像Ｉ（i,j）のＢ成分のＬＳＢを、処理１０４において、｛Ｂ（i,j）◎Ｃ（i,j）｝に変化させた画像である。ここで、Ｂ成分のＬＳＢのみを変化させた電子透かし埋め込み画像とする理由は、人間の視覚特性を考慮し、最も原画像の画質劣化が少ない電子透かしの埋め込み処理を実現するためである。

【００３２】

次に、こうして埋め込まれた電子透かしを抽出する手法について説明する。検証対象画像をＶ（i,j）で表す。また、検証者は埋め込み処理で用いた鍵ｋを共有しているとする。

【００３３】

この埋め込み処理をより理解し易くするために、その処理の流れを図７の模式図に示す。

【００３４】

図７において、４１０は自己同期パターン発生器で、ｋiを初期値として自己同期パターンＣ（i,j）を発生している。４１１は擬似乱数発生器で、値ｋを初期値として擬似乱数を発生し、それをルックアップテーブルＬＵＴ４１２のアドレス順に格納している。このＬＵＴ４１２には画像Ｉ（i,j）のＬＳＢを除いた画像Ｉ7（i,j）が入力され、それに対応して１ビットのデータＢ（i,j）が出力される。このＢ（i,j）と自己同期パターンＣ（i,j）との排他的論理和が取られ、それが原画像Ｉ（i,j）のＬＳＢに挿入されて、電子透かしが埋め込まれた画像Ｉ'（i,j）となる。

【００３５】

＜抽出処理（図２）＞

図２は、電子透かしが埋め込まれた画像から電子透かしを抽出して改竄位置を検出するための処理を説明する図である。

【００３６】

まず処理２０１で、鍵ｋを初期値として擬似乱数を生成し、埋め込み処理と同様に、入力が８ビットで出力が１ビットのルックアップテーブルＬＵＴ（）を作成する。こうして作成されたルックアップテーブルは、図７のルックアップテーブル４１２と同じものとなる。

【００３７】

次に処理２０２で、検証対象画像Ｖ（i,j）のＬＳＢを除いた画像Ｖ7（i,j）をＬＵＴに入力してＵ（i,j）＝ＬＵＴ（Ｖ7（i,j））を求める。ここで、Ｕ（i,j）は、画像Ｖ7（i,j）がルックアップテーブルＬＵＴ（）に入力されたとき、それに対応して出力される１ビットの乱数値を示す。

【００３８】

次に処理２０３で、Ｕ（i,j）＝ＬＳＢＶならばＤ（i,j）＝０とし、Ｕ（i,j）≠ＬＳＢＶならばＤ（i,j）＝１とする。ここでＬＳＢＶは、画像Ｖ（i,j）のＬＳＢである。

【００３９】

次に処理２０４で、処理２０３における出力結果Ｄ（i,j）から自己同期をとり、同期がはずれた位置を改竄位置とする。

【００４０】

ここで自己同期パターンＣ（i,j）とは、周期が長く自己相関性の強いビット系列であり、例えばＭ系列と呼ばれるビット系列が知られている。Ｍ系列は情報長ｍ、符号長ｎ＝２ｍ－１となる巡回符号であり、ｍ段のシフトレジスタを用いることにより最大長の周期をもつビット列を簡単に生成できる。

【００４１】

図４は、以下の式（１）を用いたＭ系列生成器の構成を示すブロック図である。尚、図４において、４０１はシフトレジスタ、４０２は加算器（EXOR）を示す。

【００４２】

$H(x) = h_{m-1} \cdot x^{m-1} + h_{m-2} \cdot x^{m-2} + \ldots + h_1 \cdot x + h_0$ ...式（１）

ここで図１の処理１０１で行われる自己同期パターン生成処理は以下のように行われる。

【００４３】

図４の複数のシフトレジスタ４０１のそれぞれの初期値ｃ1, ..., ｃｍをｋｉとし、その出力結果ｃｍ+1を計算する。この出力結果ｃｍ+1を最初のシフトレジスタ４０１にフィードバックさせて、演算を順次繰り返し、Ｍ×Ｎの長さの自己同期パターンＣ（i,j）＝［ｃ1, ..., ｃMxN］を生成する。ここで、各シフトレジスタ４０１内の値と、その出力値ｃｍ+1は下記の関係を持つことは明らかである。

【００４４】

$c_{i+m} = c_{i+m-1} \cdot x^{m-1} + c_{i+m-2} \cdot x^{m-2} + \ldots + c_{i+1} \cdot x + c_i$ ...式（２）

そして、図２の処理２０４における自己同期のとりかたは以下のようにして実行される。

【００４５】

図５は、Ｍ系列を検査するためのＭ系列計算器の構成を示すブロック図である。図５において、５０１はシフトレジスタ、５０２は加算器（EXOR）、５０３はスイッチを夫々示している。

【００４６】

図５において、スイッチ５０３を入力側（端子ａ）に接続すると、抽出されたビット系列が入力される。一方、スイッチ５０３をフィードバック側（端子ｂ）に接続すると、図４に示すような通常のＭ系列データの生成器になる。

【００４７】

まず最初に、改竄がなされていないビット系列が入力される場合を考える。

【００４８】

スイッチ５０３を入力側（端子ａ）に接続して、抽出されたビット列Ｄ（i,j）の最初のビット系列であるｄ1, ..., ｄｍをＭ系列生成器に入力すると、上述の式（２）の関係が成り立つのでｃｍ+1が計算される。その計算値ｃｍ+1は、抽出されたビット列Ｄ（i,j）の次のビットであるｄｍ+1と一致するので、同期がとれていることがわかる。そしてそれ以後、Ｍ×Ｎまで繰り返しても全てのビットが一致するので改竄が行われていないことがわかる。

【００４９】

次に、ｄ1, ..., ｄｉまで同期がとれていて、ｄｉ+1で同期がはずれた場合、即ち、ｃｉ+1≠ｄｉ+1となった場合には、ｄｉ+1が改竄位置であることが分かる。この後、スイッチ５０３を入力側にしたまま計算を継続すると、ｄｉ+1の影響で、それ以後のｍビットは、計算値ｃと抽出値ｄとが一致しなくなることは明らかである。

【００５０】

よって、改竄が検出されるとスイッチ５０３をフィードバック側（端子ｂ）に接続し、改竄値である抽出値ｄi+1の代わりに計算値ｃi+1をM系列生成器に入力して計算を続ける。この時、抽出値ｄi+1のみが改竄位置である場合には、ｄi+1の代わりに正しいｃi+1によって計算が継続されるので、ｄi+2以降の値は一致することになる。但し、偶然ｃi+2とｄi+2とが一致することがあるので、ｔ回以上連続して一致した場合のみ再び同期がとれた、即ち、ｄi+2以降は改竄されていないと判定する。

【００５１】

また、ｄi+1の後に集中して改竄されている場合も同様に、ｔ回以上連続して一致した場合、ｔまで遡って同期がとれた、即ち改竄はないと判定する。ここでｔを大きくとればｔ回連続して偶然に一致する確率は2^{-t}となるので、偶然に一致する確率を非常に小さくすることもできる。

【００５２】

また、初期状態において最初のｄ1, ..., ｄｍの中で１ビットでも改竄されている場合は上記の式（２）は成立しないので、最初から計算されたビットと抽出されたビットｄｍ+1とは一致せず、同期がはずれる。この場合は、正しい計算値ｃｍ+1が得られないので、式（２）が成立するまで、スイッチ５０３を入力側（端子ａ）に接続して同期が取れるまで計算を継続する。こうしてｔ回連続で同期がとれた状態が続けば、同期がとれたものとして、後は上記の途中から同期が外れた場合と同様の処理を行う。

【００５３】

また、自己同期パターンはM系列のような線形演算に限らず、シフトレジスタの値を非線形関数によって演算したり、複数のM系列パターンを非線形関数を用いて演算したりしてもよい。

【００５４】

これによって、画像がずらされていたり、部分的な削除・挿入が行われていてもその部分で同期がはずれることにより検出でき、改ざん位置とすることができる。

【００５５】

次に、本発明の課題で述べたような、同じ鍵を用いて埋め込み処理をした画像を切り貼りする攻撃の場合を考える。ここでは簡単のために他の改竄はないとする。

【００５６】

まず最初の画像の自己同期パターンから同期が取れはじめる。本実施の形態では、図１の埋め込み処理１０１において、画像毎に異なる初期値ｋiを用いて異なる自己同期パターンを埋め込んでいる。よって、異なる画像を切り貼りした部分で同期がはずれることになる。ここで全て異なる画像を組み合わせている場合、スイッチ５０３をフィードバック側（端子ｂ）にしてパターンの一致を検査しても同期が戻ることはなく、最初の画像以外の部分は改竄箇所として検出される。これは、スタンプ画像として予め定められたものを用いず、抽出したパターンから検出される自己同期性を用いているため、自己同期性の違い、即ち画像の違いが検出できるためである。

【００５７】

ここで、切り貼りをはじめとする攻撃をより厳密に検証するために、スイッチ５０３をフィードバック側に接続して計算する状態（状態１）と、スイッチ５０３を入力側に接続して計算する状態（状態２）とを並行して行うことが考えられる。例えば、同期がはずれて、それ以降、スイッチ５０３を入力側に接続して計算を継続すると、切り貼り部分からｍビットを過ぎると再び同期が戻ってくる。これは、埋め込まれているパターンが別の自己同期パターンであるので前画像の影響がなくなった時から次の画像のパターンで自己同期が取れ始めるためである。よって、「状態１」と「状態２」の計算を並行して行うことによって、異なる画像が組み合わさされていることが識別可能になる。この手法は画像をずらしたり、画像のラインを挿入したり、削除した場合などにも有効であることは明らかである。

【００５８】

以上、図２の処理２０４において行われる改竄位置の判定は、図６に示すような状態遷

移図で表される。但し、攻撃をより詳細に検証する場合は、改竄状態６０２と準同期状態６０４では「状態１」と「状態２」が並行して存在する。

【００５９】

まず初期状態６０１は処理開始状態となり、スイッチ５０３を入力側に接続し、最初の抽出パターンｄ1～ｄmを各シフトレジスタ５０１の初期値としてセットし、ｃm+1を計算して、その計算結果を抽出されたｄm+1と比較する。ここでｃm+1＝ｄm+1ならば同期状態６０３へ遷移し（６１０）、ｃm+1≠ｄm+1ならば改竄状態６０２へ遷移する（６１１）。

【００６０】

改竄状態６０２は、前状態の位置が改竄されている状態であり、この場合にはスイッチ５０３をフィードバック側（端子ｂ）に接続して計算（状態１）を継続し、計算された値ｃm+1と抽出値ｄm+1とを比較する。これらの値が一致すれば準同期状態６０４へ遷移する。これと同時に、スイッチ５０３を入力側に接続した計算（状態２）も継続し、その計算された値ｃm+1と抽出値ｄm+1とを比較する。ここで、これらの値が一致すれば準同期状態６０４へ移行する（６１２）。

【００６１】

準同期状態６０４は、「状態１」と「状態２」の結果が一致しない状態を示し、ここでは「状態１」と「状態２」の計算を継続し、「状態１」の結果がｔ1回連続で一致する場合、または「状態２」の結果がt^2回連続して一致する場合には、改竄なしとして同期状態６０３へ遷移し（６１３）、一致しない場合には改竄状態６０２へ遷移する（６１４）。

【００６２】

同期状態６０３では、スイッチ５０３を入力側（端子ａ）に接続し、抽出値ｄi～ｄi+m-1から計算値ｃi+mを計算して、ｃi+m＝ｄi+mとなっている。ここで、これらが一致しなければ改竄状態６０２へ遷移する（６１５）。

【００６３】

以上説明した画像の埋め込み処理、及び抽出処理は、図３に示す画像処理装置を用いることによって実現できる。

【００６４】

図３は、本発明の実施の形態に係る画像処理装置の構成を示すブロック図である。

【００６５】

図３において、ホストコンピュータ３０１は、例えば一般に普及しているパーソナルコンピュータであり、例えば、スキャナ３１４から読み取られた画像データを入力し、その画像データを編集、保管することが可能である。更に、ここで得られた画像データを、プリンタ３１５により印刷させることが可能である。また、ユーザからの各種マニュアル指示等は、マウス３１２、キーボード３１３からの入力により行われる。このホストコンピュータ３０１の内部では、バス３１６により後述する各ブロックが接続され、種々のデータの受け渡しが可能である。

【００６６】

図中、３０２はＣＲＴ、液晶、或いはプラズマ等の表示器（モニタ）である。３０３はＣＰＵで、内部の各ブロックの動作を制御、或いは内部に記憶されたプログラムを実行する。３０４はＲＯＭで、印刷されることが認められていない特定画像を記憶したり、予め必要な画像処理プログラムや各種データ等を記憶している。３０５はＲＡＭで、ＣＰＵ３０３にて処理を行うために一時的にプログラムや処理対象のデータを格納する。３０６はハードディスク（ＨＤ）で、ＲＡＭ３０５等に転送されるプログラムや画像データを予め格納したり、処理後の画像データを保存する。３０７はインターフェース部で、原稿或いはフィルム等をＣＣＤにて読み取って画像データを生成するスキャナ３１４に接続され、そのスキャナ３１４で得られた画像データを入力している。３０８はＣＤドライブで、外部記憶媒体の一つであるＣＤ（ＣＤ－Ｒ）に記憶されたデータを読み込み或いは書き出すことができる。３０９はＦＤドライブで、ＣＤドライブ３０８と同様に、フロッピィディスク（ＦＤ）からのデータの読み込み、ＦＤへのデータ書き出しを行う。３１０はＤＶＤ

ドライブで、ＣＤドライブ３０８と同様に、ＤＶＤからのデータ読み込み、ＤＶＤへのデータ書き出しができる。尚、これらＣＤ、ＦＤ、ＤＶＤ等に画像編集用のプログラム、或いはプリンタドライバが記憶されている場合には、これらプログラムは、一旦ＨＤ３０６上にインストールされ、必要に応じてＲＡＭ３０５に転送されて保持され、このプログラムなどに基づいてＣＰＵ３０３が実行可能となっている。３１１は、マウス３１２或いはキーボード３１３からの入力指示を受け付けるためにこれらと接続されるインターフェース部（Ｉ／Ｆ）である。３１８はモデムで、インターフェース部３１９（Ｉ／Ｆ）を介して外部のネットワークと接続されている。

【００６７】

以上の構成において、処理対象の画像データは、ＣＤ－ＲＯＭ、ＤＶＤ等の記憶媒体、又はスキャナ３１４、或いはインターフェース部３１９を介してネットワークから入力されて、一旦ＲＡＭ３０５に保持される。そしてキーボード３１３或いはマウス３１２などから入力される指示に従って、上述或いは後述する処理を実行するプログラムをＨＤ３０６から読み出してＲＡＭ３０５に記憶させ、そのプログラムを実行させることにより、本実施の形態１～４に係る処理がＣＰＵ３０３の制御の下に実行される。こうして電子透かし或いは画像が埋め込まれた画像は、ネットワークに伝送されたり、或いはＣＤやＤＶＤなどの記憶媒体に記憶される。また、ネットワーク或いは上述の記憶媒体から入力した画像データに対して、埋め込み画像の抽出処理を実行することにより、その画像データが不正に改竄されているかどうかを検出することができる。こうして得られた結果は、モニタ３０２に表示されてオペレータに警告しても良く、或いはプリンタ３１５により印刷されても良い。

【００６８】

以上説明したように本実施の形態１によれば、スタンプ画像を保存しておく必要がなく、かつ同じ鍵を用いて異なる画像に埋め込み処理を行っても安全な手法が実現できる。この手法は、鍵を初期値とする疑似乱数発生手法から生成される変換テーブルが安全であるならば、鍵を除く全てのアルゴリズムを公開しても安全な手法と言うことができる。

【００６９】

［実施の形態２］

前述の実施の形態１では、原画像Ｉ（ｉ,ｊ）の各画素のＬＳＢを除く画素値Ｉ７（ｉ,ｊ）が同じ場合、生成されるＢ（ｉ,ｊ）は同じになるので、解析がしやすい場合が考えられる。そこで、画素位置に応じて乱数で変換することによって解析を困難にする手法を以下に示す。

【００７０】

図８は本発明の実施の形態２に係る電子透かしの埋め込み処理を説明する模式図である。

【００７１】

＜埋め込み処理（図８）＞

まず処理７０１で、画像毎に定める値ｋｉを初期値として自己同期パターンＣ（ｉ,ｊ）を生成する。

【００７２】

次に処理７０２で、鍵ｋを初期値として擬似乱数を生成し、入力が８ビットで１ビット出力のルックアップテーブルＬＵＴ（）を作成する。これは各テーブルのアドレス順に、生成した擬似乱数（１ビット）を割り当てていくことによって実現される。さらに、上記疑似乱数を生成し続け、Ｍ×Ｎの２値の疑似乱数画像Ｒ（ｉ,ｊ）を生成する。

【００７３】

次に処理７０３に進み、原画像Ｉ（ｉ,ｊ）のＬＳＢを除いた画像Ｉ７（ｉ,ｊ）からＢ（ｉ,ｊ）＝ＬＵＴ（Ｉ７（ｉ,ｊ））◎Ｒ（ｉ,ｊ）を計算する。ここで、ＬＵＴ（Ｉ７（ｉ,ｊ））は、画像Ｉ７（ｉ,ｊ）がルックアップテーブルＬＵＴ（）に入力されたとき、それに対応して出力される１ビットの乱数値を示す。

【００７４】

次に処理７０４に進み、原画像Ｉ（i,j）のＬＳＢに｛Ｂ（i,j）◎Ｃ（i,j）｝を埋め込むことにより、電子透かし画像Ｉ'（i,j）を生成する。

【００７５】

ここで前述の実施の形態１と異なる点は、Ｂ（i,j）を、単に画像Ｉ7（i,j）に対応する１ビットデータ（ＬＵＴ（Ｉ7（i,j））とするのではなく、その１ビットデータと、Ｍ×Ｎの２値の疑似乱数画像Ｒ（i,j）、即ち、画素位置に応じた乱数値との排他的論理和としている点にある。

【００７６】

この実施の形態２に係る埋め込み処理の概要を図１０に示す。尚、図１０において、図７と共通する部分は同じ記号で示し、それらの説明を省略する。

【００７７】

図１０において、原画像Ｉ（i,j）のＬＳＢには、ルックアップテーブル４１２の出力（ＬＵＴ（Ｉ7（i,j））と、鍵ｋを初期値として生成された画像Ｉ（i,j）の画素数分の擬似乱数Ｒ（i,j）との排他的論理和が、原画像のＬＳＢに挿入される点が特徴である。

【００７８】

＜抽出処理（図９）＞

図９は本発明の実施の形態２に係る電子透かしの抽出及び改竄位置の検出処理を説明する模式図である。

【００７９】

まず処理８０１で、鍵ｋを初期値として擬似乱数を生成し、図８の埋め込み処理と同様に、入力が８ビットで１ビット出力のルックアップテーブルＬＵＴ（）を作成する。さらに、上記疑似乱数を生成し続け、Ｍ×Ｎの２値疑似乱数画像Ｒ（i,j）を生成する。

【００８０】

次に処理８０２で、検証対象画像Ｖ（i,j）のＬＳＢを除いた画像Ｖ7（i,j）からＵ（i,j）＝ＬＵＴ（Ｖ7（i,j））◎Ｒ（i,j）を計算する。ここで、ＬＵＴ（Ｖ7（i,j））は、画像Ｖ7（i,j）を入力したときのルックアップテーブルの出力（１ビット）を示す。

【００８１】

次に処理８０３で、Ｕ（i,j）＝ＬＳＢVならばＤ（i,j）＝０、Ｕ（i,j）≠ＬＳＢVならばＤ（i,j）＝１とする。なお、ここで、ＬＳＢVは、検証対象画像Ｖ（i,j）のＢ成分のＬＳＢを示している。

【００８２】

次に処理８０４で、Ｄ（i,j）から自己同期をとり、同期がはずれた位置を改竄位置とする。

【００８３】

本実施の形態２では、画素値のみに依存し、画素位置に依存しないルックアップテーブルの出力値（ＬＵＴ（Ｉ7（i,j））又はＬＵＴ（Ｖ7（i,j））に対して、図８の処理７０２或いは図９の処理８０２で生成した疑似乱数Ｒ（i,j）との排他的論理和を求める（処理７０３，８０３）ことにより、同じ画素値であっても異なるＢ（i,j），Ｕ（i,j）を出力できるようにして解析を困難にしている。但し、擬似乱数Ｒ（i,j）は、自己同期パターンＣ（i,j）や抽出パターンＤ（i,j）又は画像Ｉ（i,j）やＶ（i,j）と排他的論理和を求めてもよく、いずれの場合も効果は同じである。

【００８４】

［実施の形態３］

従来のスタンプ画像を保存する改竄位置の検出手法では、画像の拡大縮小や回転、切取りなどが起こった場合、保存されているスタンプ画像の形状を参考に改竄画像の拡大縮小や回転，切取りを補正することができる。

【００８５】

これに対し本実施の形態では、スタンプ画像を保存しないので、スタンプ画像の形状を参考にできない。よって、下記のような手段によって拡大縮小や回転、切取りに対する対

応を行う。

【００８６】

１）電子透かしが埋め込まれた画像Ｉ'（i,j）に拡大縮小、回転などの全体的な改竄が行われている場合

改竄の定義によるが、拡大縮小、回転も改竄と見なす場合は、画像サイズや形状の変化を気にせず抽出処理を行えば、同じ画素位置における検証対象画像Ｖ（i,j）と埋め込み画像Ｉ'（i,j）はほとんど異なっており、さらに疑似乱数との対応も異なるために、画像全体の改竄として検出される。よって、この場合何の手段も必要としない。

【００８７】

しかし、拡大縮小、回転は改竄と見なさない場合は、予め後述のようなレジストレーション信号を埋め込んでおき、拡大縮小や回転などを補正した後に、改竄位置検出処理を行えば、拡大縮小、回転などの復元可能な変更は改竄と見なさないことも可能である。よってこの場合、図１１に示すようにレジストレーション信号による画像補正を画像補正部１２０１で行った後に、前述の実施の形態１及び２に示した機能を有する改竄位置検出部１１０２による検証及び改竄位置の検出処理を実行する。

【００８８】

２）埋め込まれた画像Ｉ'（i,j）に切り取りを行う場合

図１の処理１０１において生成される自己同期パターンＣ（i,j）を、例えば最初ｉ＝０，ｊ＝０からｉ＝ｉ＋１としてｉ＝Ｍまで対応させ、次にｊ＝ｊ＋１，ｉ＝０の位置から同様の対応をｉ＝Ｍ，ｊ＝Ｎまで行う。即ち、画像の上列から順に自己同期パターンＣ（i,j）を対応させていく場合、埋め込み画像Ｉ'（i,j）の下部のみの切取りは、全ての位置でＵ（i,j）＝ＬＳＢ∀となり、改竄は検出されない。よって、自己同期パターンＣ（i,j）の画像位置への対応を画像Ｉ（i,j）の切取りが行いにくくなるように縦横１ラインずつ行うなどしておく、又は、予めランダムに定めておくことが必要がある。または、画像サイズに応じて自己同期パターンを変えるなどすることも可能である。

【００８９】

また前述の実施の形態では、多値画像の場合でのみ説明したが、カラー画像であっても、それをＲＧＢに分解することによって本発明を実施することができる。この場合、ＲＧＢ毎に別々に行っても、ＲＧＢの結果を合成して行っても良い。

【００９０】

また、本実施の形態では、自己同期パターンを例にして説明したが、本発明はパターンの違いが容易に識別可能であればよく、これに限定されるものではない。例えば、自己同期パターンでなく、通常の画像を色成分毎に強さを変えて挿入し、色識別によって改竄位置を検出することも可能である。

【００９１】

また、自己相関演算は周波数変換して演算することもできるので、自己同期検出は図５に示すような構成に限定されない。

【００９２】

また、同期確立のためのｔは、初期状態６０１からの同期状態６０３に遷移する場合と、改竄状態６０２からの同期状態６０３に遷移する場合で変えることも可能である。一般に、初期状態６０１からの同期確立のためのｔの方が、改竄状態６０２から同期確立のためのｔよりも大きく設定すべきである。また、改竄状態６０２からの同期確立のためのｔの値を"１"とする場合には、準同期状態６０４は経由しないので、ｔの値に応じて図６の状態遷移図は可変となる。よって、本実施の形態における状態の遷移は図６に限定されるものではない。

【００９３】

また自己同期パターンは、簡単のためにＭ系列を例にして説明したが、疑似乱数は一般に自己同期性を持つために疑似乱数生成器を用いることができる。例えば、図４に示すＭ系列生成器はｍビットを入力し、１ビットを出力する変換テーブルに置換えられる。よって、ある初期値ｋ1から疑似乱数を生成し、新たなｍビット入力で１ビット出力のルック

アップテーブルの出力に対応させることもできる。

【００９４】

また、鍵ｋ0を初期値として疑似乱数を生成し、画像Ｉ（i,j）のＢ成分のＬＳＢを除く各ビットとの排他的論理和をとることによって、画像Ｉ（i,j）を画素位置に応じて暗号化して、画像Ｉ（i,j）の代わりＬＳＢへの埋め込みパターンを生成しても良い。このとき、抽出処理では同じ鍵ｋ0から生成される疑似乱数を用いて画像Ｖ（i,j）を暗号化して同様の処理を行う。

【００９５】

［レジストレーション信号］

電子透かしには種々の攻撃が施される可能性がある。このような攻撃としては例えば、ＪＰＥＧなどの非可逆圧縮、拡大・縮小或いは回転などの幾何変換などが挙げられる。レジストレーション信号とは、それらの種々の攻撃によって生じた幾何的歪みを補正するために埋め込まれる信号である。レジストレーション処理は、電子透かしを埋め込む際に、付加情報とは別に、特定の信号（レジストレーション信号）を画像に付加し、電子透かしを抽出する際には、付加情報を抽出する前に、前記レジストレーション信号を用いて付加情報の抽出を助長する処理である。

【００９６】

このレジストレーションを用いた方式としては、米国特許第５６３６２９２に提案されている方式が挙げられる。これは、予め埋め込んである幾何パターンを用いて、画像に施された幾何変換を自動的に算出する方式である。また、特開平１１－３５５５４７号公報に提案されている対称軸を持たない２次元波から構成される方式などもある。

【０１１２】

［実施の形態４］

図１３は、本発明の実施の形態４に係る画像埋め込み処理を説明するための概念図である。尚、この実施の形態４及びこれ以降の実施の形態においても、画像処理装置及びＭ系列生成器及びＭ系列計算器の構成は、前述の図３乃至図５を参照して説明したものと同じであるため、それらの説明を省略する。

【０１１３】

ここでは原画像Ｉ（i,j）をＭ×Ｎ画素からなる多値画像（ここでは、１画素が８ビットの多値画像として説明する）とする。

【０１１４】

＜埋め込み処理（図１３）＞

まず処理１６０１で、画像毎に定める値ｋiを初期値として自己同期パターンＣ（i,j）を生成する。次に処理１６０２で、その生成した自己同期パターンＣ（i,j）を画像Ｉ（i,j）のＬＳＢに埋め込み、電子透かしが埋め込まれた画像Ｉ'（i,j）を生成する。

【０１１５】

この電子透かしの埋め込み処理によって得られる電子透かしが埋め込まれた画像Ｉ'（i,j）は、画像Ｉ（i,j）のＬＳＢを、処理１６０２において、自己同期パターンＣ（i,j）に変化させた画像である。ここで、画像Ｉ（i,j）がカラー画像である場合、Ｂ成分のＬＳＢを変化させる。これは、人間の視覚特性を考慮して、最も原画像の画質劣化が少ない電子透かしの埋め込み処理を実現するためである。

【０１１６】

次に、こうして埋め込まれた電子透かしを抽出する手法について説明する。尚、ここでは検証対象画像をＶ（i,j）で表す。

【０１１７】

＜抽出処理（図１４）＞

図１４は、電子透かしが埋め込まれた画像Ｖ（i,j）から電子透かしを抽出して改竄位置を検出するための処理を説明する図である。

【０１１８】

まず処理１７０１で、検証対象画像Ｖ（i,j）のＬＳＢを抽出し、それをＤ（i,j）とす

る。次に処理１７０２で、処理１７０１における出力結果Ｄ（ｉ,ｊ）から自己同期をとり、その同期がはずれた位置を改竄位置とする。

【０１１９】

ここで自己同期パターンＣ（ｉ,ｊ）は、周期が長く自己相関性の強いビット系列であり、例えばＭ系列と呼ばれるビット系列が知られている。Ｍ系列は情報長ｍ、符号長ｎ＝２^{m}－１となる巡回符号であり、ｍ段のシフトレジスタを用いることにより最大長の周期をもつビット列を簡単に生成できる。

【０１２０】

この様なＭ系列生成器及びＭ系列計算機の構成は、前述の図４及び図５を参照して説明しているので、ここではその説明を省略する。

【０１２１】

このようにして、画像がずらされていたり、部分的な削除・挿入が行われていてもその部分で同期がはずれることにより検出でき、改ざん位置とすることができる。以上説明した画像の埋め込み処理及び抽出処理は、前述の図３に示す画像処理装置を用いることによって実現できる。

【０１２２】

以上説明したように本実施の形態４によれば、スタンプ画像を保存しておく必要がなく、かつ数パターン分のずれ、挿入、削除があっても、自己同期を検出することにより改ざん位置を検出できるという効果がある。

【０１２３】

［実施の形態５］

前述の実施の形態４では、原画像Ｉ（ｉ,ｊ）のＬＳＢに直接自己同期パターンを埋め込んだが、そのままでは、解析がしやすい場合が考えられる。そこで、他のパターンと組み合わせたり、自己同期パターンをさらに変換するなどが考えられる。そこで本実施の形態５では、自己同期パターンに別の鍵を初期値とする擬似乱数を用いて解析しにくくした例で説明する。

【０１２４】

図１５は、本発明の実施の形態５に係る電子透かしの埋め込み処理を説明する模式図である。

【０１２５】

＜埋め込み処理（図１５）＞

まず処理１８０１で、画像毎に定める値ｋｉを初期値として自己同期パターンＣ（ｉ,ｊ）を生成する。次に処理１８０２で、鍵ｋを初期値として自己同期パターンと同じ長さの擬似乱数Ｂ（ｉ,ｊ）を生成する。次に処理１８０３に進み、Ｂ（ｉ,ｊ）◎Ｃ（ｉ,ｊ）を計算する。ただし、ここで◎は、EXOR（排他的論理和）の演算を表す。次に処理１８０４に進み、原画像Ｉ（ｉ,ｊ）のＬＳＢに、｛Ｂ（ｉ,ｊ）◎Ｃ（ｉ,ｊ）｝を埋め込むことにより、電子透かし画像Ｉ'（ｉ,ｊ）を生成する。

【０１２６】

ここで前述の実施の形態４と異なる点は、自己同期パターンＣ（ｉ,ｊ）を擬似乱数Ｂ（ｉ,ｊ）によってストリーム暗号化している点である。これによって、鍵ｋを知らない第３者は自己同期パターンの解析が困難になる。

【０１２７】

更に、鍵ｋを用いて生成した乱数によって埋め込み位置をＬＳＢ以外とすることも可能である。例えば、生成した擬似乱数を２ビット毎に区切り、それを埋め込み位置とすることが考えられる。即ち、“００”をＬＳＢ，“０１”をＬＳＢの１つ上位のビットに埋め込み、更に“１０”，“１１”をそれぞれ１つずつ上位のビットに埋め込むことなどが考えられる。

【０１２８】

＜抽出処理（図１６）＞

図１６は、本発明の実施の形態５に係る電子透かしの抽出及び改竄位置の検出処理を説

明する模式図である。

【０１２９】

まず処理１９０１で、鍵ｋを初期値として擬似乱数を生成し、Ｂ（ｉ,ｊ）を生成する。次に処理１９０２で、検証対象画像Ｖ（ｉ,ｊ）のＬＳＢを抽出し、Ｕ（ｉ,ｊ）とする。次に処理１９０３で、Ｄ（ｉ,ｊ）＝Ｕ（ｉ,ｊ）◎Ｂ（ｉ,ｊ）を計算する。次に処理１９０４で、Ｄ（ｉ,ｊ）から自己同期をとり、同期がはずれた位置を改竄位置とする。

【０１３０】

従来のスタンプ画像を保存する改竄位置の検出手法では、画像の拡大縮小や回転、切取りなどが起こった場合、保存されているスタンプ画像の形状を参考に改竄画像の拡大縮小や回転，切取りを補正することができる。これに対し本実施の形態では、スタンプ画像を保存しないので、スタンプ画像の形状を参考にできない。よって、前述の図１１を参照して説明したような手段によって拡大縮小や回転、切取りに対する対応を行う。

【０１３１】

また、本実施の形態では、自己同期パターンを例にして説明したが、本発明はパターンの違いが容易に識別可能であればよく、これに限定されるものではない。例えば、自己同期パターンでなく、通常の画像を色成分毎に強さを変えて挿入し、色識別によって改竄位置を検出することも可能である。

【０１３２】

また、自己相関演算は周波数変換して演算することもできるので、自己同期検出は図５に示すような構成に限定されない。

【０１３３】

前述の実施の形態では、画像を例に取り、改竄位置の検出手法を説明したが、本発明において対象するデータは画像に限定されるものではなく、例えばデジタルデータをブロック化し、それに対応する改竄位置を検出する検査情報を付加する場合等すべて含まれる。その一例として、前述の図１２に示すように、複数のデータブロックによって１つのコンテンツが表されるも同様に実施できる。よって、上記の実施の形態で説明した画素をデータブロックとすれば、任意のコンテンツのデータブロックごとの改ざん位置検出ができることは明らかである。すなわち、複数画素をデータブロックとしてもよいし、音楽情報のように時系列の情報に対しても改ざん位置検出が行える。さらには、コンテンツをデータブロックに分解してパケット通信する場合などに対してもパケット中の改ざん位置が検出可能であることは明らかである。

【０１３４】

また、ＪＰＥＧやＭＰＥＧなどは外見上一つのデータストリームとして構成されているが、実質は８×８の画素ブロックごと、又はフレーム毎のデータブロックに分解でき、それらのブロックが連続して１つのコンテンツを構成している。図１２では簡単のため、データブロックを分解して示したが、外見上１つのデータストリームであっても、或いは実質的に複数のデータブロックから構成されるコンテンツに対しても本発明が適用可能であることも明らかである。

【０１３５】

本発明は上記実施の形態を実現するための装置及び方法及び実施の形態で説明した方法を組み合わせて行う方法のみに限定されるものではなく、上記システム又は装置内のコンピュータ（ＣＰＵあるいはＭＰＵ）に、上記実施の形態を実現するためのソフトウエアのプログラムコードを供給し、このプログラムコードに従って上記システムあるいは装置のコンピュータが上記各種デバイスを動作させることにより上記実施の形態を実現する場合も本発明の範疇に含まれる。

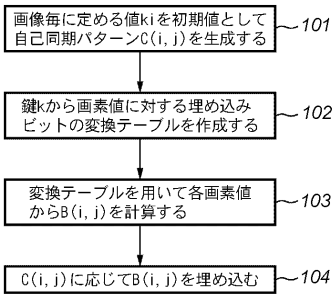
【０１３６】

またこの場合、前記ソフトウエアのプログラムコード自体が上記実施の形態の機能を実現することになり、そのプログラムコード自体、及びそのプログラムコードをコンピュータに供給するための手段、具体的には上記プログラムコードを格納した記憶媒体は本発明の範疇に含まれる。

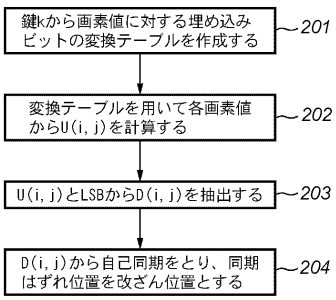
【０１３７】
このようなプログラムコードを格納する記憶媒体としては、例えばフロッピィディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ、磁気テープ、不揮発性のメモリカード、ＲＯＭ等を用いることができる。
【０１３８】
また、上記コンピュータが、供給されたプログラムコードのみに従って各種デバイスを制御することにより、上記実施の形態の機能が実現される場合だけではなく、上記プログラムコードがコンピュータ上で稼働しているＯＳ（オペレーティングシステム）、あるいは他のアプリケーションソフト等と共同して上記実施の形態が実現される場合にも、係るプログラムコードは本発明の範疇に含まれる。
【０１３９】
更に、この供給されたプログラムコードが、コンピュータの機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに格納された後、そのプログラムコードの指示に基づいてその機能拡張ボードや機能格納ユニットに備わるＣＰＵ等が実際の処理の一部または全部を行い、その処理によって上記実施の形態が実現される場合も本発明の範疇に含まれる。
【０１４０】
以上説明したように本実施の形態によれば、従来はできなかったような、同じ鍵を用いて異なる画像の埋め込み処理をしても安全な電子透かしによる改竄位置の検出手法が実現できる。
【０１４１】
更に、本実施の形態によれば、スタンプ画像は保存する必要がなく、鍵のみの秘匿でよいため効率的である。
【０１４２】
また本実施の形態によれば、従来はできなかったような、デジタルコンテンツのずれや挿入・削除といった改ざん部分のみを検出することができる。
【０１４３】
【発明の効果】
以上説明したように本発明によれば、電子透かし手法のアルゴリズムや埋め込みパターンが知られたとしても、確実に改竄位置を検出できる。
【０１４４】
また本発明によれば、スタンプ画像を保存する必要がなく、かつ同じ鍵を用いて異なる画像データに埋め込み処理を行っても、その画像データにおける改竄位置を確実に検出できるという効果がある。
【０１４５】
また本発明によれば、従来はできなかったような、デジタル画像のずれや挿入或は削除といった改ざん部分のみを検出することができる。
【図面の簡単な説明】
【図１】　本発明の実施の形態１に係る画像の埋め込み処理の概要を説明する概念図である。
【図２】　実施の形態１における埋め込み画像の抽出及び改竄位置の特定処理を説明する概要図である。
【図３】　本発明の実施の形態に係る画像処理装置の構成を示すブロックである。
【図４】　本発明の実施の形態１に係る自己同期パターン生成器の構成を示すブロック図である。
【図５】　本発明の実施の形態１に係る自己同期パターン計算器の構成を示すブロック図である。
【図６】　本発明の実施の形態１に係る自己同期をとるための状態遷移を説明する図である。
【図７】　本発明の実施の形態１に係る画像の埋め込み処理を説明する概念図である。

【図８】　本発明の実施の形態２に係る画像の埋め込み処理の概要を説明する概念図である。
【図９】　実施の形態２における埋め込み画像の抽出及び改竄位置の特定処理を説明する概要図である。
【図１０】　本発明の実施の形態２に係る画像の埋め込み処理を説明する概念図である。
【図１１】　本発明の実施の形態３に係る抽出処理の概要を説明する図である。
【図１２】　本発明の実施の形態３に係るデータブロックへの検査ビットの埋め込みを説明する図である。
【図１３】　本発明の実施の形態４に係る画像の埋め込み処理の概要を説明する概念図である。
【図１４】　実施の形態４における埋め込み画像の抽出及び改竄位置の特定処理を説明する概要図である。
【図１５】　本発明の実施の形態５に係る画像の埋め込み処理の概要を説明する概念図である。
【図１６】　実施の形態５における埋め込み画像の抽出及び改竄位置の特定処理を説明する概要図である。

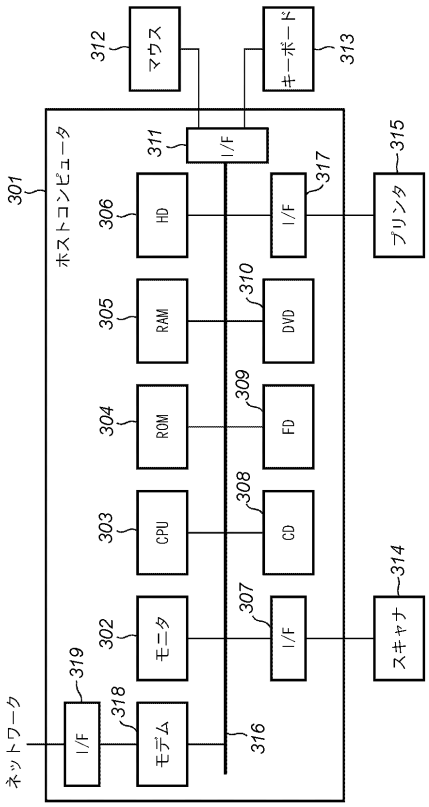
【図１】



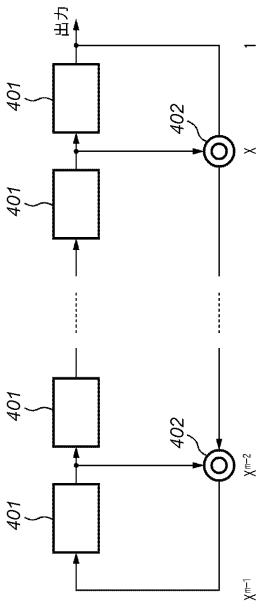
【図２】



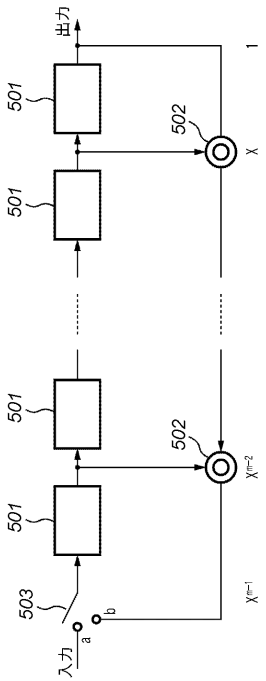
【図３】



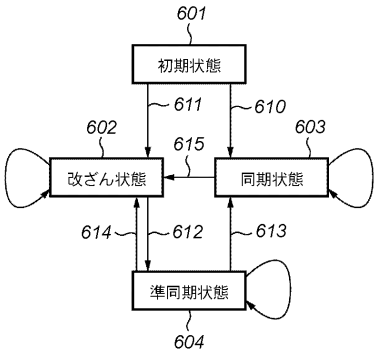
【図４】



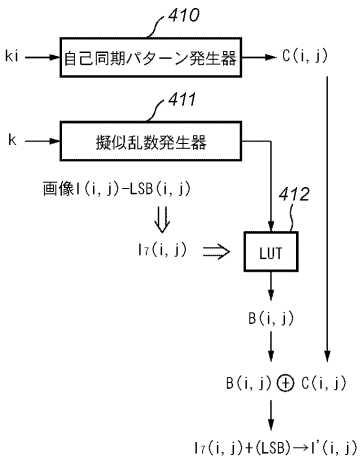
【図５】



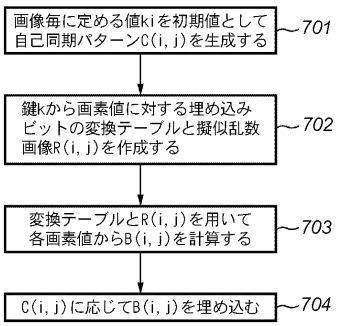
【図６】



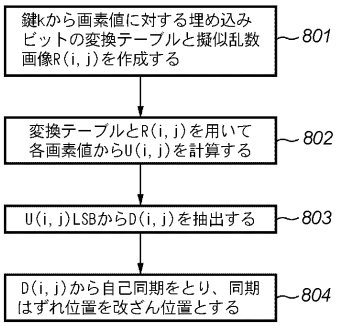
【図７】



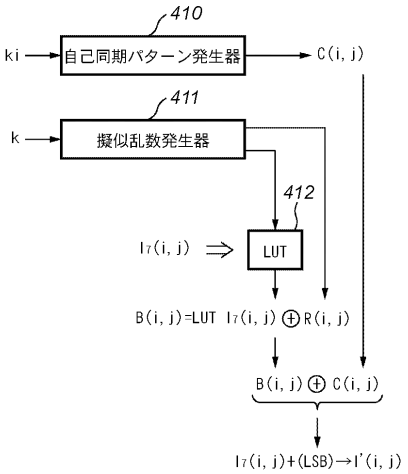
【図８】



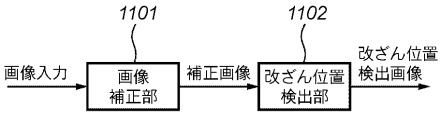
【図９】



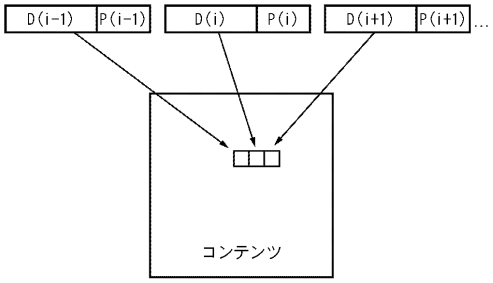
【図１０】



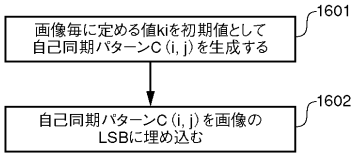
【図１１】



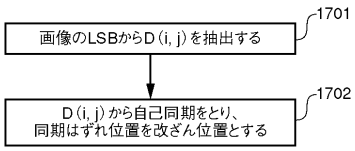
【図１２】



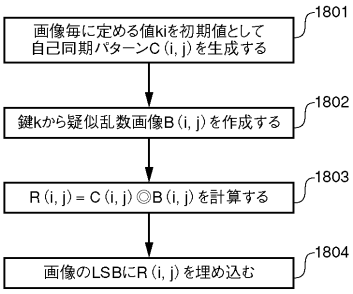
【図１３】



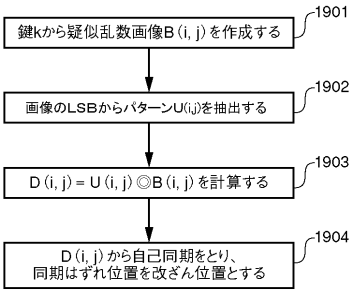
【図１４】



【図１５】



【図１６】



フロントページの続き

審査官　手島　聖治

(56)参考文献　特開平１１－０７５０５５（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04N1/38-1/409
G06T1/00-1/40
G06T3/00-5/50
G06T9/00-9/40
G09C1/00-5/00
H04K1/00-3/00
H04L9/00-9/38