



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) **DE 602 15 427 T2** 2007.03.15

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 283 640 B1**

(21) Deutsches Aktenzeichen: **602 15 427.8**

(96) Europäisches Aktenzeichen: **02 015 690.7**

(96) Europäischer Anmeldetag: **18.07.2002**

(97) Erstveröffentlichung durch das EPA: **12.02.2003**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **18.10.2006**

(47) Veröffentlichungstag im Patentblatt: **15.03.2007**

(51) Int Cl.<sup>8</sup>: **H04N 7/30** (2006.01)

**G06T 1/00** (2006.01)

**G06T 5/00** (2006.01)

(30) Unionspriorität:

**2001223612**      **24.07.2001**      **JP**

**2002200577**      **09.07.2002**      **JP**

(73) Patentinhaber:

**Seiko Epson Corp., Tokyo, JP**

(74) Vertreter:

**Hoffmann, E., Dipl.-Ing., Pat.-Anw., 82166  
Gräfelfing**

(84) Benannte Vertragsstaaten:

**AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB,  
GR, IE, IT, LI, LU, MC, NL, PT, SE, SK, TR**

(72) Erfinder:

**Ishikawa, Masaki c/o Seiko Epson Corporation,  
Suwa-shi, Nagano-ken 392-8502, JP; Miura,  
Hirotsuna c/o Seiko Epson Corporatio, Suwa-shi,  
Nagano-ken 392-8502, JP**

(54) Bezeichnung: **Bildverarbeitungsprozessor, Bildverarbeitungsprogramm und Bildverarbeitungsverfahren**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

**Beschreibung****HINTERGRUND DER ERFINDUNG****Gebiet der Erfindung**

**[0001]** Die Erfindung betrifft eine Verarbeitungseinrichtung und ein Verfahren zum Durchführen einer Filterverarbeitung bei einem expandierten Bild, wenn eine Expansion von Bildern in der Einheit von Blöcken basierend auf komprimierten Bilddaten erfolgt, die in JPEG (Joint Photographic Coding Experts Group) oder MPEG (Moving Picture Experts Group), etc. komprimiert sind, und betrifft insbesondere eine Bildverarbeitungseinrichtung, ein Bildverarbeitungsprogramm und ein Bildverarbeitungsverfahren, die beim Dekodieren eines Bildes eine Filterverarbeitung vereinfachen können und deren Geschwindigkeit vergrößern können und die für eine effektiven Filtereffekt geeignet sind.

**Beschreibung der verwandten Technik**

**[0002]** JPEG ist bereits verbreitet als Kompressionsverfahren für ein unbewegtes Bild bekannt, und MPEG ist bereits verbreitet als Kompressionsverfahren für ein Bewegtbild bekannt. Bei derartigen Bildverarbeitungsverfahren wird ein Block von  $8 \times 8$  Pixeln als Einheit verwendet und es wird eine Diskrete Kosinustransformation (DCT) durchgeführt. Die DCT ist eine Verarbeitung zum Zerlegen eines ursprünglichen Bildes in räumliche Frequenzkomponenten. Ein Bild kann dadurch komprimiert werden, dass räumlich redundante Information reduziert wird.

**[0003]** Im Übrigen werden, wenn JPEG oder MPEG verwendet wird, zwei Arten von Rauschen erzeugt, wie nachstehend erläutert wird, und das Rauschen beeinträchtigt die Bildqualität. Und zwar ist bei JPEG und MPEG, da ein Block von  $8 \times 8$  Pixeln als Verarbeitungseinheit verwendet wird, bei einigen Bildern eine Grenze des Blockes als Rauschen zu beobachten. Dies liegt daran, dass DCT eine Verarbeitung ist, die einige Komponenten, die zu DC-(Direct Current)-Komponenten gehören, aus den räumlichen Frequenzkomponenten eines Bildes entfernt, die Verarbeitung für jeden Block durchgeführt wird, und ein Helligkeitsdurchschnittswert von Block zu Block variiert. Ein an der Grenze auftretendes Rauschen wird als Blockrauschen (Blockartefakt) bezeichnet.

**[0004]** Weiter werden bei einer DCT, die bei JPEG und MPEG durchgeführt wird, harmonische Komponenten als redundante Information aus den räumlichen Frequenzkomponenten eines Bildes entfernt. Somit wird Rauschen in einem Teil erzeugt, der einen großen Helligkeitsunterschied gegenüber den umgebenden Teilen aufweist. Im Fall großer Helligkeitsunterschiede in einem Block, beispielsweise wenn Personen im Hintergrund eines natürlichen Bildes vor-

handen sind, werden harmonische Komponenten durch die DCT aus den Daten, die eine ursprüngliche Grenze bilden, eliminiert, was zu einem Unschärfeferauschen an der Grenze führt. Ein derartiges Rauschen wird als Moskito-Effekt bezeichnet.

**[0005]** Die japanische Patentoffenlegungsschrift Nr. 3-46482 offenbart ein Verfahren zum Verringern von Blockartefakten und des Moskito-Effektes. Bei dem in der Publikation offenbarten Verfahren (nachfolgend als erstes herkömmliches Beispiel bezeichnet) wird eine Filterverarbeitung durchgeführt, wenn Grenzpixel, die so positioniert sind, dass sich zwischen ihnen eine Grenze benachbarter Blöcke befindet, eine größere Signalpegeldifferenz als ein erster Schwellenwert haben, und wenn eine Signalpegeldifferenz zwischen einem Grenzpixel und einem Pixel benachbart zum Grenzpixel im gleichen Block kleiner ist als ein zweiter Schwellenwert in Bezug auf Grenzpixel und Pixel in Arrays in Blöcken.

**[0006]** In diesem Fall wird, wie dargestellt in [Fig. 9](#), basierend auf den Differenzen  $d_0$ ,  $d_1$  und  $d_2$  der Signalpegel  $S_1$ ,  $S_0$ ,  $S'_0$  und  $S'_1$  in Bezug auf die Pixelanordnungen  $X_1$  und  $X_0$  und die Pixelanordnungen  $X'_0$  und  $X'_1$ , die auf beiden Seiten einer Blockgrenze positioniert sind, ein Signalpegel der Pixelanordnungen  $X_0$  und  $X'_0$  korrigiert. Und zwar werden neue Signalpegel  $S_{0\text{-new}}$  und  $S'_{0\text{-new}}$  durch die nachfolgend dargestellten Gleichungen erhalten.

$$S_{0\text{-new}} = (S_1 + 2 \cdot S_0 + S'_0) / 4$$

$$S'_{0\text{-new}} = (S_0 + 2 \cdot S'_0 + S'_1) / 4$$

**[0007]** Als weiteres Verfahren zum Verringern von Blockartefakten und eines Moskitoeffektes sind ein VM (Verifikationsmodell) betreffend MPEG-4 und ein Verfahren bekannt, das im offengelegten japanischen Patent Nr. 11-98505 beschrieben ist (nachfolgend als zweites herkömmliches Beispiel bezeichnet). Im zweiten herkömmlichen Beispiel wird beurteilt, ob sich die in [Fig. 10](#) dargestellten vertikalen und horizontalen Blockgrenzen in einem Default-Modus oder ein DC-Offset-Modus befinden, und eine Verarbeitung erfolgt für jeden der Modi.

**[0008]** In diesem Fall sind Pixelsätze  $S_0$ ,  $S_1$  und  $S_2$  definiert, wobei dabei Blockgrenzen  $B_1$  und  $B_2$  als Basispunkte verwendet werden, und es wird wahlweise bestimmt, ob ein Deblocking-Modus der Default-Modus oder der DC-Offset-Modus ist, und zwar gemäß einem Modusbestimmungswert, der auf einem Ausmaß eines Blockverzerrungsphänomens basiert. Und dann wird, wenn bestimmt wird, dass der Modus der Default-Modus ist, ein Vier-Punkt DCT-Kernel verwendet, um Frequenzinformation in der Blockgrenzenumgebung der Pixel zu finden. Wenn im Modusbestimmungsschritt der Modus als der DC-Offset-Modus bestimmt wird, wird bestimmt,

ob eine Ausführung des DC-Offset-Modus erforderlich ist, und falls erforderlich, wird das Blockverzerrungsphänomen entfernt.

**[0009]** Jedoch kann beim ersten herkömmlichen Beispiel, da lediglich die Grenzpixel korrigiert werden, eine Verzerrung der Blöcke nicht in ausreichender Weise entfernt werden. Außerdem ist beim VM und dem zweiten herkömmlichen Beispiel die Verarbeitung kompliziert und eine Berechnungslast ist zu groß.

**[0010]** Außerdem ist es, wenn derartige Verfahren hardwaremäßig implementiert werden, da die Verarbeitung relativ kompliziert ist, schwierig, die Schaltungsgröße zu reduzieren. In Hinblick auf eine Miniatursierung wird angestrebt, die Verarbeitung soweit wie möglich zu vereinfachen, sogar wenn dies in einem gewisse Maße zu Lasten der Bildqualität geht. Außerdem wird angestrebt, da eine Grenze für die installierbare Speicherkapazität besteht, die Speicherkapazität zu minimieren, die für eine Berechnung bei der Filterverarbeitung belegt wird.

**[0011]** US 6 188 799 B1 offenbart eine Bildverarbeitungseinrichtung und ein Bildverarbeitungsverfahren gemäß dem Oberbegriff der unabhängigen Ansprüche. Das Deblocking-Filter zum Entfernen von Blockartefakten, das bei diesem Stand der Technik auf Grenzpixel benachbarter Blöcke angewandt wird, verwendet eine lineare Interpolation. Die Interpolation wird auf Pixel in einem linearen Array angewandt, das vier Pixel auf jeder Seite einer Blockgrenze, d. h. zwei Gruppen von jeweils vier Pixeln beinhaltet. Der gefilterte Wert für ein spezielles Pixel in einer dieser Gruppen wird aus einer gewichteten Version von dessen eigenem Pixelwert und einer gewichteten Version des Pixelwertes dieses Pixels in der anderen Gruppe bestimmt, das sich unmittelbar benachbart zur Blockgrenze befindet. Das Deblocking-Filter wird für ein derartiges Array von Pixeln lediglich dann angewandt, wenn der Absolutwert der Differenz zwischen dem Durchschnittswert der vier Pixel, die zu einem der zwei Blöcke gehören, und dem Durchschnittswert der vier Pixel, die zu dem anderen Block gehören, unterhalb eines vorbestimmten Schwellenwertes liegt. Außerdem wird das Deblocking-Filter auf eine spezielle Gruppe eines derartigen Array lediglich angewandt, wenn eine weitere Bedingung erfüllt ist. Diese weitere Bedingung besteht darin, dass die Summe der absoluten Differenzen zwischen jedem Pixel in der Gruppe und dem Durchschnittswert dieser Gruppe unterhalb eines weiteren vorbestimmten Schwellenwertes liegt. Somit ist eine Berechnung des durchschnittlichen Pixelwertes für jede der zwei Gruppen in jedem Array, die möglicherweise zu filtern sind, eine wichtige Vorbedingung dieses Standes der Technik.

## INHALT DER ERFINDUNG

**[0012]** Somit wurde die Erfindung in Hinblick auf die zuvor beschriebenen ungelösten Probleme des herkömmlichen Verfahrens erdacht. Die Erfindung hat als ihr erstes Ziel das Vorsehen einer Bildverarbeitungseinrichtung und eines Bildverarbeitungsverfahrens, die beim Decodieren eines Bildes eine Filterverarbeitung vereinfachen und deren Geschwindigkeit vergrößern können, und die zum Realisieren eines effektiven Filtereffektes geeignet sind. Weiter ist das zweite Ziel der Erfindung das Bereitstellen einer Bildverarbeitungseinrichtung und eines Bildverarbeitungsprogramms, sowie eines Bildverarbeitungsverfahrens, die zur Verminderung einer Speicherkapazität geeignet sind, welche beim Decodieren eines Bildes für eine Berechnung bei der Filterverarbeitung belegt wird.

**[0013]** Um die zuvor beschriebenen Ziele zu erreichen, führt eine Bildverarbeitungseinrichtung gemäß Anspruch 1 der Erfindung eine Filterverarbeitung mit einem expandierten Bild durch, und zwar während oder nachdem das Bild in der Einheit von vorbestimmten Blöcken expandiert wurde, die auf komprimierten Bilddaten basieren, welche mittels einer Bildkompression, bei der eine Diskrete Kosinustransformation und eine Quantisierung in der Einheit von Blöcken durchgeführt wird, komprimiert wurden.

**[0014]** Während oder nachdem ein Bild in der Einheit von Blöcken, die auf komprimierten Bilddaten basieren, expandiert wird, kann eine arithmetische Filteroperation für ein einziges Pixeln in einem Block lediglich basierend auf einem Wert des Pixels und einem Wert von einem beliebigen der Pixel im benachbarten Block durchgeführt werden.

**[0015]** Gemäß der Erfindung werden, wenn die arithmetische Filteroperation für ein einziges Pixel durchgeführt wird, die Werte von lediglich zwei Pixeln ohne eine Verwendung eines Durchschnittswertes einer Anzahl von Pixeln verwendet. Daher kann, im Vergleich zur herkömmlichen Technik, der Filterprozess zur Reduzierung des Blockrauschens beim Decodieren eines Bildes relativ vereinfacht werden.

**[0016]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 2 wird die arithmetische Filteroperation für eines der Pixel in einem eine Grenze schneidenden Pixel-Array durchgeführt, wenn bei keiner Differenz zwischen Pixeln ein Schwellenwert überschritten wird, wobei die Pixel zwischen dem einen Pixel und dem Grenzpixel liegen, der auf der Grenze in demselben Pixel-Array des benachbarten Blockes liegt. Dabei ist es, da der Schwellenwert ein konstanter Wert ist, beim der Filterverarbeitung nicht erforderlich, einen Schwellenwert zu berechnen.

**[0017]** Bei der Bildverarbeitungseinrichtung gemäß

Anspruch 3 wird, wenn eine Differenz zwischen einem Grenzpixel und einem benachbarten Pixel verglichen wird, der Vergleich basierend auf einem Schwellenwert vorgenommen, der größer als ein Schwellenwert zum Vergleichen einer Differenz zwischen dem Grenzpixel und einem Pixel außer dem benachbarten Pixel ist. Das Grenzpixel und das benachbarte Pixel bilden jeweils Grenzen unterschiedlicher Blöcke, und die Helligkeit kann große Schwankung aufweisen. Daher ist es besser, einen Schwellenwert für einen Vergleich größer als einen Schwellenwert für einen Vergleich der anderen Pixel festzulegen, um eine Filterverarbeitung in einer relativ positiven Weise durchzuführen.

**[0018]** Bei einer Bildverarbeitungseinrichtung gemäß Anspruch 4 wird, wenn eine Differenz zwischen dem Grenzpixel und dem benachbarten Pixel verglichen wird, ein Schwellenwert auf einen Wert äquivalent zu " $2^{n-3}$ " oder einen Näherungswert festgelegt, wobei dabei ein Pixel einen Maximalwert von " $2^n-1$ " hat, und ein Vergleich wird basierend auf dem Schwellenwert vorgenommen. Dabei wird, wenn der Schwellenwert größer als ein Wert äquivalent zu " $2^{n-3}$ " oder einen Näherungswert festgelegt ist, eine Filterverarbeitung durchgeführt, auch wenn dies nicht erforderlich ist. Somit steigt die Verarbeitungslast, und die Geschwindigkeit der Filterverarbeitung kann nicht erhöht werden. Im umgekehrten Fall wird, wenn der Schwellenwert kleiner als ein Wert äquivalent zu " $2^{n-3}$ " oder einen Näherungswert festgelegt ist, eine Filterverarbeitung nicht durchgeführt, auch wenn dies erforderlich ist. Somit kann in effektiver Filtereffekt nicht erwartet werden.

**[0019]** Außerdem ist, wenn eine Differenz zwischen dem Grenzpixel und einem Pixel außer dem benachbarten Pixel verglichen wird, ein Schwellenwert auf einen Wert äquivalent zu " $2^{n-4}$ " oder einen Näherungswert festgelegt, wobei dabei ein Pixel einen Maximalwert von " $2^n-1$ " hat, und ein Vergleich wird basierend auf dem Schwellenwert vorgenommen. Dabei wird, wenn der Schwellenwert größer als ein Wert äquivalent zu " $2^{n-4}$ " oder einen Näherungswert festgelegt ist, eine Filterverarbeitung durchgeführt, auch wenn dies nicht erforderlich ist. Somit steigt die Verarbeitungslast, und die Geschwindigkeit der Filterverarbeitung kann nicht erhöht werden. Im umgekehrten Fall wird, wenn der Schwellenwert kleiner als ein Wert äquivalent zu " $2^{n-4}$ " oder einen Näherungswert festgelegt ist, eine Filterverarbeitung nicht durchgeführt, auch wenn dies erforderlich ist. Somit kann in effektiver Filtereffekt nicht erwartet werden.

**[0020]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 5 erfolgt eine Beurteilung, ob die arithmetische Filteroperation für ein Pixel durchgeführt werden sollte, in einer Richtung eines sequentiellen Wegbewegens von der Grenze, wobei die Beurteilung von einem benachbarten Pixel beginnt, das be-

nachbart zum Grenzpixel ist. Und dann wird bei der Beurteilung, wenn beurteilt wird, dass eine arithmetische Filteroperation nicht für ein Pixel durchgeführt werden sollte, keine arithmetische Filteroperation für das Pixel und die Pixel in den späteren Stufen durchgeführt.

**[0021]** Ursprünglich ist es bei der Beurteilung, ob eine arithmetische Filteroperation für ein Pixel durchgeführt werden sollte, erforderlich, zu beurteilen, ob eine Differenz zwischen Pixeln einen Schwellenwert überschreitet, wobei sich die Pixel zwischen dem einen Pixel und einem Grenzpixel befinden. Beispielsweise ist, wenn vier Pixel zwischen einem Zielpixel und einem Grenzpixel vorhanden sind (einschließlich dem Zielpixel und dem Grenzpixel), eine dreimalige Beurteilung erforderlich. Jedoch ist, wenn eine Beurteilung in der obigen Reihenfolge vorgenommen wird, abgesehen von einer Beurteilung des Zielpixels und des Grenzpixels, eine Beurteilung bereits bei der Beurteilung für die Pixel der vorhergehenden Stufe erfolgt. Somit ist es möglich, eine Beurteilung für die Pixel der späteren Stufe wegzulassen. Daher ist es, wenn vier Pixel zwischen dem am weitesten entfernt befindlichen Pixel und dem Grenzpixel der Zielpixel vorhanden sind, eine maximal dreimalige Beurteilung ausreichend, wohingegen normalerweise eine Beurteilung für 3! Male über ein Pixel-Array erforderlich ist.

**[0022]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 6 werden Daten aller Pixel, die sich zwischen dem einen Pixel und dem Grenzpixel im Pixel-Array befinden, aus der Bildspeichereinrichtung in die Arbeitsspeichereinrichtung in einer integrierten Weise eingelesen, und die arithmetische Filteroperation wird basierend auf den Pixeldaten der Arbeitsspeichereinrichtung durchgeführt.

**[0023]** Dabei speichert die Arbeitsspeichereinrichtung Pixeldaten mittels einer beliebigen Einrichtung zu einer beliebigen Zeit. Pixeldaten können vorab gespeichert werden, oder Pixeldaten können durch eine Eingabe, etc. von außen während einer Operation der Verarbeitungseinrichtung gespeichert werden, ohne dass Pixeldaten vorab gespeichert werden.

**[0024]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 7 wird die arithmetische Filteroperation entweder für ein Pixel-Array in einer horizontalen Richtung oder ein Pixel-Array in einer vertikalen Richtung im Block durchgeführt, und dann wird die arithmetische Filteroperation für das jeweils andere durchgeführt.

**[0025]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 8 wird die arithmetische Filteroperation für das ein einziges Pixel basierend auf einem Wert des Zielwertes und Werten der benachbarten Pixel durchgeführt. Und dann wird bei der arithmetischen Filter-

operation, wenn eine Differenz zwischen einem Pixel und dem Zielpixel der benachbarten Pixel einen Schwellenwert überschreitet, der Wert des Zielpixels anstelle des Wertes des einen Pixels verwendet.

**[0026]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 9 wird, wenn eine Differenz zwischen einem einzigen Pixel und dem Zielpixel der benachbarten Pixel verglichen wird, ein konstanter Wert als Schwellenwert verwendet und der Vergleich basierend auf dem Schwellenwert vorgenommen. Dabei ist es, da der Schwellenwert ein konstanter Wert ist, nicht erforderlich, bei der Filterverarbeitung einen Schwellenwert zu berechnen.

**[0027]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 10 wird, wenn eine Differenz zwischen einem einzigen Pixel und dem Zielpixel der benachbarten Pixel verglichen wird, der Schwellenwert auf einen Wert äquivalent zu  $2^{n-4}$  oder einen Näherungswert festgelegt, wobei dabei ein Pixel einen Maximalwert von  $2^n - 1$  hat, und der Vergleich wird basierend auf dem Schwellenwert vorgenommen. Dabei wird, wenn der Schwellenwert größer als ein Wert äquivalent zu  $2^{n-4}$  oder einen Näherungswert festgelegt ist, ein Wert des benachbarten Pixels nicht durch einen Wert des Zielpixels ersetzt, selbst wenn das Ersetzen des Wertes erforderlich ist. Somit kann kein effektiver Filtereffekt erwartet werden. Im umgekehrten Fall wird, wenn der Schwellenwert kleiner als ein Wert äquivalent zu  $2^{n-4}$  oder einen Näherungswert festgelegt ist, ein Wert des benachbarten Pixels durch einen Wert des Zielpixels ersetzt, selbst wenn das Ersetzen des Wertes nicht erforderlich ist. Somit kann kein effektiver Filtereffekt erwartet werden.

**[0028]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 11 wird die arithmetische Filteroperation für ein Pixel basierend auf einem Wert eines Zielpixels und Werten für vier, zwei oder einem einzigen benachbarten Pixel, der zum Zielpixel benachbart ist, durchgeführt.

**[0029]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 12 wird die arithmetische Filteroperation für ein einziges Pixel durchgeführt, und zwar dadurch, dass ein Wert des Zielpixels durch den vorbestimmten Wert dividiert wird, ein Wert eines benachbarten Pixels durch den vorbestimmten Wert dividiert wird, und die Divisionsergebnisse addiert werden.

**[0030]** Bei der Bildverarbeitungseinrichtung gemäß Anspruch 13 werden Daten aller Pixel ausgehend von dem einen Pixel bis zum Grenzpixel in dem Pixel-Array aus der Bildspeichereinrichtung in die Arbeitsspeichereinrichtung in einer integrierten Weise eingelesen, und die arithmetische Filteroperation wird basierend auf den Pixeldaten in der Arbeitsspeichereinrichtung ausgeführt.

## KURZE BESCHREIBUNG DER ZEICHNUNGEN

**[0031]** [Fig. 1](#) ist ein Blockdiagramm der Konfiguration des Computersystems, auf das die Erfindung angewandt wird;

**[0032]** [Fig. 2](#) zeigt das Prinzip der Deblocking-Filterverarbeitung;

**[0033]** [Fig. 3](#) ist ein Balkendiagramm, das den Wert eines jeden der Pixel der Pixel-Arrays PG1 und PG2 zeigt;

**[0034]** [Fig. 4](#) ist ein Ablaufdiagramm der Deblocking-Filterverarbeitung;

**[0035]** [Fig. 5](#) ist ein Ablaufdiagramm des Filterprozesses bei Schritt S102;

**[0036]** [Fig. 6](#) ist ein Ablaufdiagramm des Filterprozesses bei Schritt S104;

**[0037]** [Fig. 7](#) zeigt das Prinzip der Deringing-Filterverarbeitung;

**[0038]** [Fig. 8](#) ist ein Ablaufdiagramm der Deringing-Filterverarbeitung;

**[0039]** [Fig. 9](#) zeigt den durch das herkömmliche Bildverarbeitungssystem ausgeführten Filterprozess; und

**[0040]** [Fig. 10](#) zeigt das Pixel, auf das beim herkömmlichen Bildverarbeitungssystem Bezug genommen wird.

## DETAILLIERTE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

**[0041]** Nachfolgend wird eine erste Ausführungsform der Erfindung anhand der anliegenden Zeichnungen erläutert. [Fig. 1](#) bis [Fig. 8](#) sind Zeichnungen, die eine Bildverarbeitungseinrichtung und ein Bildverarbeitungsprogramm sowie ein Bildverarbeitungsverfahren gemäß der ersten Ausführungsform der Erfindung zeigen.

**[0042]** Bei der vorliegenden Ausführungsform werden die Bildverarbeitungseinrichtung und das Bildverarbeitungsprogramm sowie das Bildverarbeitungsverfahren der Erfindung auf den folgenden Fall angewandt: wie in [Fig. 1](#) dargestellt, werden in einem Computer **100**, wenn in einem MPEG-Format komprimierte Bewegtbilddaten durch ein Betriebssystem (OS = Operating System) decodiert werden, das im Multitasking-Betrieb arbeitet, zur Wiedergabe eines Bewegtbildes eine Filterverarbeitung zur Reduzierung von Blockartefakten und eine Filterverarbeitung zum Reduzieren eines Moskito-Effektes beim wiedergegebenen Bild durchgeführt. Nachfolgend wird ers-

tere Filterverarbeitung als Deblocking-Filterverarbeitung bezeichnet und letztere Filterverarbeitung wird als Deringing-Filterverarbeitung bezeichnet.

**[0043]** Als Erstes wird die Konfiguration eines für die Erfindung verwendeten Computersystems in Übereinstimmung mit [Fig. 1](#) erläutert. [Fig. 1](#) ist ein Blockdiagramm, das die Konfiguration des Computersystems für eine Anwendung der Erfindung darstellt.

**[0044]** Wie in [Fig. 1](#) dargestellt, besteht der Computer **100** aus: einer CPU **30** zum Steuern von Berechnungen und des gesamten Systems gemäß den Steuerprogrammen, einem ROM **32**, um die Steuerprogramme und dergleichen der CPU **30**, etc. in einem vorbestimmten Bereich vorab zu speichern, einem RAM **34**, um aus dem ROM **32**, etc. gelesene Daten und Berechnungsergebnisse zu speichern, die bei einem Berechnungsprozess der CPU **30** benötigt werden, einer LCDC **36**, welche die Daten eines speziellen Bereiches des RAM **34** in ein Bildsignal umwandelt und das Signal an die LCD (Flüssigkristallanzeigeeinrichtung) **44** ausgibt, einem Decoder **38** zum Decodieren von Bewegtbilddaten, einem Post-Filterglied **40** zum Lesen der Daten eines speziellen Bereiches des RAM **34**, um eine Deblocking-Filterverarbeitung und eine Deringing-Filterverarbeitung durchzuführen, einem RGB-Umwandlungsglied **42** zum Durchführen einer Farbumwandlung mit den Daten vom Post-Filterglied **40**, und einer LCD **44** zum Bewerkstelligen einer Anzeige basierend auf den Daten vom RGB-Umwandlungsglied **42**. Die CPU **30**, das ROM **32**, das RAM **34**, die LCDC **36** und der Decoder **38** sind miteinander über einen Bus **39** verbunden, der eine Signalleitung zum Übertragen der Daten ist, und zwar derart, dass die Daten gesendet und empfangen werden können.

**[0045]** Das RAM **34** weist ein VRAM **35** als speziellen Bereich auf, um die auf der LCD **44** anzuzeigenden Daten zu speichern. Auf das VRAM **35** kann unabhängig durch die Post-Filtereinheit **40** und durch die über den Bus **39** verbundenen Vorrichtungen, wie beispielsweise die CPU **30** zugegriffen werden. Außerdem weist das VRAM **35** mindestens zwei Bereiche auf, die in der Lage sind, Bilddaten eines einzigen Bildschirms der LCD **44** zu speichern. Der eine der Bereiche wird verwendet, um dem Decoder **38** das Expandieren von Bewegtbilddaten zu ermöglichen, und der andere Bereich wird verwendet, um der Post-Filtereinheit ein Auslesen zu ermöglichen, und diese Bereiche werden abwechselnd zu einer vorbestimmten Zeit umgeschaltet.

**[0046]** Die LCDC **36** liest Daten aus dem VRAM **35** sequentiell ausgehend von der vorderen Adresse zu einem vorbestimmten Zeitraum durch Steuern der Post-Filtereinheit **40** und der RGB-Umwandlungseinheit **42** aus, und die LCDC **36** wandelt die gelesenen Daten in ein Bildsignal um und gibt das Signal an die

LCD **44** aus. Und zwar liest die Post-Filtereinheit **40** Daten aus dem VRAM **35** reagierend auf die Steuerung der LCDC **36** aus, führt eine Deblocking-Filterverarbeitung und eine Deringing-Filterverarbeitung bei den gelesenen Daten durch, und gibt die Daten an die RGB-Umwandlungseinheit **42** aus. Die RGB-Umwandlungseinheit **42** führt eine RGB-Umwandlung mit den Daten von der Post-Filtereinheit **40** reagierend auf die Steuerung der LCDC **36** durch und gibt die Daten an die LCD **44** aus.

**[0047]** Der Decoder **38** führt ein sequentielles Decodieren der in den Bewegtbilddaten enthaltenen Bilddaten bei jedem vorbestimmten Zeitpunkt T durch (z. B. 20 ms). Insbesondere liest der Decoder **38** im MPEG-Format komprimierte Bewegtbilddaten aus dem RAM **34** reagierend auf einen Decodierstartbefehl aus der CPU **30** aus und decodiert die Bilddaten an das VRAM **35**, basierend auf den gelesenen Bewegtbilddaten. Im VRAM **35** erfolgt ein Expandieren der Bilddaten, wobei Bilddaten (YUV-Daten), die einen Farbgrad eines einzigen Pixels in der LCD **44** angeben, als Einheit verwendet werden. Außerdem werden im Decoder **38** eine IDCT (Inverse Diskrete Kosinustransformation) und eine Bewegungskompensation zwischen Decodieroperationen beim MPEG-Format durchgeführt. Im Übrigen sind eine Kompression und ein Decodieren beim MPEG-Format dieselben wie die des herkömmlichen Beispiels.

**[0048]** Als Nächstes wird die Konfiguration der Post-Filtereinheit **40** detailliert erörtert.

**[0049]** In der Post-Filtereinheit **40** sind die Konfiguration und die interne Verarbeitung hardwaremäßig realisiert. Ein Puffer, der in der Lage ist, Bilddaten von 17 Pixeln zu speichern, ist in dieser enthalten. Während Bilddaten in den Puffer aus dem VRAM **35** eingelesen werden, werden Pixel für jeden Block (in der vorliegenden Ausführungsform 64 Pixel, die aus einer Quadratmatrix von 8 Pixeln in horizontaler Richtung und 8 Pixeln in vertikaler Richtung bestehen) basierend auf den ausgelesenen Bilddaten verarbeitet. Speziell wird eine Deblocking-Filterverarbeitung bei aus dem VRAM **35** ausgelesenen Bilddaten durchgeführt, und ein Deringing-Filterverarbeitung wird bei Bilddaten ausgeführt, die bereits einer Deblocking-Filterverarbeitung unterzogen wurden. Im Übrigen ist die Reihenfolge der Filterverarbeitung nicht speziell eingeschränkt. Beide Typen der Verarbeitung können durchgeführt werden, oder ein einziger Typ der Verarbeitung kann selektiv durchgeführt werden. Wenn die Verarbeitung selektiv durchgeführt wird, kann die Auswahl durch eine softwaremäßige oder eine hardwaremäßige Einstellung, oder eine dynamische Einstellung vorgenommen werden, und zwar basierend auf Wiedergabegeschwindigkeit, Qualität oder Datenkapazität eines Bildes.

**[0050]** Das Prinzip der Deblocking-Filterverarbei-

tung der Erfindung wird nachfolgend detailliert anhand der [Fig. 2](#) und 3 erläutert. Die [Fig. 2](#) und 3 sind Zeichnungen zur Erläuterung des Prinzips der Deblocking-Filterverarbeitung.

**[0051]** In [Fig. 2](#) sind, betreffend ein Pixel-Array PG1, das eine Grenze B1 zwischen benachbarten Blöcken schneidet, Pixelwerte auf der linken Seite der Grenze B1 in der Zeichnung als  $v_4$ ,  $v_3$ ,  $v_2$  und  $v_1$  bezeichnet, und zwar in der Reihenfolge ihrer Nähe zur Grenze B1, und die Pixelwerte auf der rechten Seite der Grenze B1 in der Zeichnung sind als  $v_5$ ,  $v_6$ ,  $v_7$  und  $v_8$  in der Reihenfolge ihrer Nähe zur Grenze B1 bezeichnet. Außerdem werden in ähnlicher Weise, betreffend ein Pixel-Array PG2, das eine Grenze B2 zwischen benachbarten Blöcken schneidet, Pixelwerte am oberen Teil von der Grenze B2 in der Zeichnung als  $v_4$ ,  $v_3$ ,  $v_2$  und  $v_1$  bezeichnet, und zwar in der Reihenfolge ihrer Nähe zur Grenze B2, und Pixelwerte beim unteren Teil von der Grenze B2 in der Zeichnung sind als  $v_5$ ,  $v_6$ ,  $v_7$  und  $v_8$  in der Reihenfolge ihrer Nähe zur Grenze B2 bezeichnet.

**[0052]** In diesem Fall werden betreffend die Pixel-Arrays PG1 und PG2 die Pixelwerte  $v_1'$  bis  $v_8'$  nach der Deblocking-Filterverarbeitung durch die unten dargestellten arithmetischen Gleichungen (1) bis (8) berechnet.

$$v_1' = v_1 \quad (1)$$

$$v_2' = (v_2 + v_5)/2 + (v_2 - v_5)/4 \quad (2)$$

$$v_3' = (v_3 + v_5)/2 + (v_3 - v_5)/8 \quad (3)$$

$$v_4' = (v_4 + v_5)/2 \quad (4)$$

$$v_5' = (v_5 + v_4)/2 + (v_5 - v_4)/8 \quad (5)$$

$$v_6' = (v_6 + v_4)/2 + (v_6 - v_4)/4 \quad (6)$$

$$v_7' = v_7 - (v_7 - v_4)/8 \quad (7)$$

$$v_8' = v_8 \quad (8)$$

**[0053]** Die zuvor beschriebenen arithmetischen Filtergleichungen (1) bis (8) werden wie nachstehend beschrieben hergeleitet. **Fig. 3** ist ein Balkendiagramm, das die Pixelwerte der Pixel-Arrays PG1 und PG2 angibt.

**[0054]** Wie in [Fig. 3\(A\)](#) dargestellt, werden zwischen benachbarten Blöcken A und B, da DC-(Direct Current)-Komponenten aus den räumlichen Frequenzkomponenten eines Bildes durch eine DCT-Verarbeitung während der Kompression entfernt werden, Pegeldifferenzen durch Schwankungen des durchschnittlichen Helligkeitswertes erzeugt. Die Pegeldifferenzen führen zu Blockrauschen (Blockartefakten). Dabei wird, um Blockartefakte zu verrin-

gern, in Betracht gezogen, dass Bilder der Blöcke A und B mit "glattem Übergang" verbunden werden. Ein Durchschnittswert  $V_A$  von vier Pixeln im Block A und ein Durchschnittswert  $V_B$  von vier Pixeln im Block B werden durch die nachstehend dargestellten Gleichungen (9) und (10) berechnet.

$$V_A = (v_1 + v_2 + v_3 + v_4)/4 \quad (9)$$

$$V_B = (v_5 + v_6 + v_7 + v_8)/4 \quad (10)$$

**[0055]** Nimmt man an, dass eine Verschiebung eines Pixelwertes  $V_i$  von Durchschnittswerten  $V_A$  und  $V_B$  den Wert  $\delta v_i$  hat, ist der Pixelwert  $v_i$  durch die nachfolgend dargestellten Gleichungen (11) und (12) repräsentiert.

$$v_i = V_A + \delta v_i (i = 1 - 4) \quad (11)$$

$$v_i = V_B + \delta v_i (i = 5 - 8) \quad (12)$$

**[0056]** Um die Bilder der Blöcke A und B "glatt" zu verbinden, ist es zu bevorzugen, eine Differenz zwischen Durchschnittswerten  $V_A$  und  $V_B$  auf "0" festzulegen. Wie in [Fig. 3\(B\)](#) dargestellt, ist es zu bevorzugen, einen Pixelwert  $v_i$  stufenweise zu korrigieren. Eine Differenz  $\Delta$  zwischen Stufen wird wie nachstehend dargestellt berechnet.

$$\Delta = (V_B - V_A)/8 \quad (13)$$

**[0057]** Somit kann beispielsweise ein korrigierter Wert  $v_5'$  eines Pixelwertes  $V_5$  durch die nachstehend dargestellte Gleichung (14) berechnet werden.

$$\begin{aligned} v_5' &= (V_B + V_A)/2 + \Delta + \delta v_5 \\ &= (5V_8 + 3V_A)/8 + \delta v_5 \\ &= \{5(v_5 - \delta v_5) + 3(v_4 - \delta v_4)\}/8 + \delta v_5 \\ &= \{5v_5 + 3v_4 + 3(\delta v_5 - \delta v_4)\}/8 \end{aligned} \quad (14)$$

**[0058]** Um einen korrekten Wert zu berechnen, muss eine Berechnung gemäß der zuvor beschriebenen Gleichung (14) ausgeführt werden. Bei der zuvor beschriebenen Gleichung (14) wird angenommen, dass  $(\delta v_5 - \delta v_4)$  klein ist, und es wird der folgende angenäherte Ausdruck (15) erhalten.

$$v_5' = 5v_5 + 3v_4/8 \quad (15)$$

**[0059]** Weiter können, in ähnlicher Weise bei den anderen Pixeln im Block B, korrigierte Werte  $v_6'$  bis  $v_8'$  durch die nachstehend dargestellten Gleichungen (16) bis (18) berechnet werden.

$$\begin{aligned} v_6' &= (V_B + V_A)/2 + 2\Delta + \delta v_6 \\ &= \{6v_6 + 2v_4 + 2(\delta v_6 - \delta v_4)\}/8 \end{aligned} \quad (16)$$

$$\begin{aligned} v_7' &= (V_B + V_A)/2 + 3\Delta + \delta v_7 \\ &= \{7v_7 + v_4 + (\delta v_7 - \delta v_4)\}/8 \end{aligned} \quad (17)$$

$$\begin{aligned} v_8' &= (V_B + V_A)/2 + 4\Delta + \delta v_8 \\ &= v_8 \end{aligned} \quad (18)$$

**[0060]** Somit werden die folgenden angenäherten Ausdrücke (19) und (20) erzielt.

$$v_6' = (3v_6 + v_4)/4 \quad (19)$$

$$v_7' = (7v_7 + v_4)/8 \quad (20)$$

**[0061]** Außerdem ist betreffend die Pixel im Block A die Berechnung dieselbe, abgesehen davon, dass ein betrachteter Pixel einen Wert von  $v_5$  hat. Korrigierte Werte  $v_1'$  bis  $v_4'$  können durch die nachfolgend dargestellten Gleichungen (21) bis (24) berechnet werden.

$$\begin{aligned} v_1' &= (V_B + V_A)/2 - 3\Delta + \delta v_1 \\ &= \{7v_1 + v_5 + (\delta v_1 - \delta v_5)\}/8 \end{aligned} \quad (21)$$

$$\begin{aligned} v_2' &= (V_B + V_A)/2 - 2\Delta + \delta v_2 \\ &= \{6v_2 + 2v_5 + 2(\delta v_2 - \delta v_5)\}/8 \end{aligned} \quad (22)$$

$$\begin{aligned} v_3' &= (V_B + V_A)/2 - \Delta + \delta v_3 \\ &= \{5v_3 + 3v_5 + (\delta v_3 - \delta v_5)\}/8 \end{aligned} \quad (23)$$

$$\begin{aligned} v_4' &= (V_B + V_A)/2 + \delta v_4 \\ &= (v_4 + v_5 + 2\delta v_4)/2 \end{aligned} \quad (24)$$

**[0062]** Somit werden die folgenden angenäherten Gleichungen (25) bis (28) erhalten.

$$v_1' = v_1 \quad (25)$$

$$v_2' = (3v_2 + v_5)/4 \quad (26)$$

$$v_3' = (5v_3 + 3v_5)/8 \quad (27)$$

$$v_4' = (v_4 + v_5)/2 \quad (28)$$

**[0063]** Außerdem werden bei der tatsächlichen Berechnung, da durch eine Bitverschiebung die Verarbeitungsgeschwindigkeit stärker als bei einer Division vergrößert werden kann, vor der Verwendung die Gleichungen (25) bis (28), (15), (19), (20) und (18) in die Gleichungen (1) bis (8) transformiert. Eine derartige Deblocking-Filterverarbeitung wird bei einem Pixel-Array in einer horizontalen Richtung bzw. bei einem Pixel-Array in einer vertikalen Richtung durchgeführt. Der den Berechnungsaufwand beinhaltet zweimalige Addition bzw. Subtraktion und eine viermalige Bitverschiebung. Die Bitverschiebung führt nicht zu einer großen Last, wenn sie hardwaremäßig implementiert wird. Bei herkömmlichen Post-Filtern weist ein einziges Pixel den maximalen gewichteten Durchschnitt von  $7 \times 7$  Pixeln auf. Somit kann die vorliegende Ausführungsform den Berechnungsaufwand im Vergleich zu den herkömmlichen Post-Filtern beträchtlich verringern.

**[0064]** Die Deblocking-Filterverarbeitung gemäß der Erfindung kann durch den im Ablaufdiagramm von [Fig. 4](#) dargestellten Prozess realisiert werden. [Fig. 4](#) ist ein Ablaufdiagramm der Deblocking-Filterverarbeitung.

**[0065]** Wenn die Deblocking-Filterverarbeitung durch die Post-Filtereinheit **40** durchgeführt wird, geht die Steuerung zuerst auf Schritt S100, wie in [Fig. 4](#) dargestellt.

**[0066]** Bei Schritt S100 wird die Startadresse des vorderen Blocks von der LCDC **36** erhalten, die Steuerung geht weiter auf Schritt S102, die Deblocking-Filterverarbeitung wird mit dem Pixel-Array durchgeführt, das zum Gebiet des oberen rechten Abschnittes des Blockes gehört, die Steuerung geht weiter auf Schritt S104, die Deblocking-Filterverarbeitung wird bei dem Pixel-Array durchgeführt, das zum Gebiet des oberen linken Abschnittes des Blockes gehört, die Steuerung geht weiter auf Schritt S106, die Deblocking-Filterverarbeitung wird bei dem Pixel-Array durchgeführt, das zum Gebiet des unteren rechten Abschnittes des Blockes gehört, die Steuerung geht weiter auf Schritt S108, die Deblocking-Filterverarbeitung wird bei dem Pixel-Array durchgeführt, das zum unteren linken Abschnitt des Blockes gehört, und die Steuerung geht weiter auf Schritt S110.

**[0067]** Bei Schritt S110 wird bestimmt, ob die Prozesse von Schritt S102 bis Schritt S108 betreffend alle Blöcke, die einen einzigen Bildschirm des Bildes bilden, abgeschlossen sind, oder nicht. Falls bestimmt wird, dass alle Blöcke vollständig verarbeitet wurden (JA), dann endet eine Abfolge der Prozesse. Falls die Bestimmung nicht so erfolgt (NEIN), dann geht die Steuerung auf Schritt S112, und die Startadresse des nächsten Blockes wird von der LCDC **36** erhalten, wodurch die Steuerung auf Schritt S102 weitergeht.

**[0068]** Dann wird der Filterprozess bei Schritt S102 nachfolgend detailliert Bezug nehmend auf [Fig. 5](#) beschrieben. [Fig. 5](#) ist ein Ablaufdiagramm des Filterprozesses bei Schritt S102.

**[0069]** Der Filterprozess bei Schritt S102 ist die Deblocking-Filterverarbeitung, die bei dem Pixel-Array durchgeführt wird, das zu dem Gebiet des oberen rechten Abschnittes des Blockes gehört. Wenn der Prozess bei Schritt S102 durchgeführt wird, geht die Steuerung zuerst auf Schritt S200, wie in [Fig. 5](#) dargestellt.

**[0070]** Bei Schritt S200 wird gemäß der in den Schritten S100 und S112 erhaltenen Startadresse die Adresse des vorderen Pixel-Array im Gebiet des oberen rechten Abschnittes des Blockes berechnet, die Steuerung geht weiter auf Schritt S202, die Pixeldat-

ten, die bei der Operation in der horizontalen Richtung benötigt werden, und die Pixeldaten, die bei der Operation in der vertikalen Richtung im Pixel-Array benötigt werden, bei dem die Deblocking-Filterverarbeitung durchzuführen ist, werden gemeinsam in den Puffer eingelesen, und zwar gemäß der berechneten Adresse des Pixel-Array, und die Steuerung geht weiter auf Schritt S204. Wenn beispielsweise das Pixel-Array, bei dem die Deblocking-Filterverarbeitung durchzuführen ist, die dritte Reihe von oben ist, werden die Pixeldaten, die insgesamt 17 Pixel beinhalten, d. h. die vier von rechts nach links aneinander grenzenden Pixel ausgehend vom oberen rechten Pixel im Block, die acht Pixel zwei Reihen unterhalb des Vier-Pixel-Array, die vier von rechts nach links aneinander grenzenden Pixel ausgehend vom unteren rechten Pixel im obigen benachbarten Block, und die Pixel zwei Reihen unterhalb des oberen linken Pixels im rechten benachbarten Block ausgelesen.

**[0071]** Dann wird bei den Schritten S204, S208 und S212 bestimmt, ob die Deblocking-Filterverarbeitung bei jedem Pixel durchzuführen ist, oder nicht. Die Bestimmung, ob die Deblocking-Filterverarbeitung bei jedem Pixel durchzuführen ist, oder nicht, wird basierend auf dem Differenzwert benachbarter Pixel durchgeführt. Die Deblocking-Filterverarbeitung gemäß der Erfindung wird basierend auf dem Wert der Grenzpixel zwischen benachbarten Blöcken durchgeführt. Falls es eine große Flanke an der Grenze zwischen den Blöcken gibt, wird nach der Korrektur eine Unschärfe oder ein Schatten im Umriss erzeugt. Um dies zu vermeiden, wird bestätigt, dass es keine große Differenz in Daten an der Grenze zwischen den Blöcken gibt. Wenn es eine gestufte Differenz gibt, wird die Deblocking-Filterverarbeitung bei den betreffenden Pixeln nicht durchgeführt. Die Bestimmung wird sowohl in der horizontalen als auch in der vertikalen Richtung ausgeführt.

**[0072]** Bei Schritt S204 wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_4$  und  $v_5$  größer als der konstante Wert  $C_2$  ist, oder nicht. Falls bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_4$  und  $v_5$  gleich groß oder kleiner als der konstante Wert  $C_2$  ist (NEIN), dann geht die Steuerung auf Schritt S206. Der konstante Wert  $C_2$  ist auf "32" festgelegt, unter der Annahme, dass der maximale Wert des Pixelwertes "255" ist. Falls der konstante Wert  $C_2$  größer als "32" gesetzt ist, dann wird der Filterprozess durchgeführt, auch wenn dieser nicht erforderlich ist, wodurch die Prozesslast vergrößert wird und eine Durchführung des Filterprozesses mit hoher Geschwindigkeit nicht gelingt. Andererseits wird, falls dieser kleiner als "32" gesetzt ist, keine Durchführung des Filterprozesses, auch wenn dieser erforderlich ist, wodurch kein effizienter Filtereffekt bewerkstelligt wird.

**[0073]** Bei Schritt S206 wird die arithmetische Filter-

operation durch die obige Gleichung (4) unter Verwendung von  $v_4$  und  $v_5$  durchgeführt, um  $v_4'$  zu berechnen, die Steuerung geht weiter auf Schritt S208, und es wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_3$  und  $v_4$  größer als der konstante Wert  $C_1$  ist, oder nicht. Wenn bestimmt wird, dass der Absolutwert die Differenz zwischen  $v_3$  und  $v_4$  gleich groß oder kleiner als der konstante Wert  $C_1$  ist (NEIN), dann geht die Steuerung weiter auf Schritt S210. Der konstante Wert  $C_1$  ist auf "16" festgelegt, unter der Annahme, dass der Maximalwert des Pixelwertes "255" ist. Falls der konstante Wert  $C_1$  größer als "16" gesetzt ist, dann wird der Filterprozess durchgeführt, auch wenn dieser nicht erforderlich ist, wodurch die Prozesslast vergrößert wird und eine Durchführung des Filterprozesses mit hoher Geschwindigkeit nicht gelingt. Andererseits wird, falls dieser kleiner als "16" gesetzt ist, keine Durchführung des Filterprozesses, auch wenn dieser erforderlich ist, wodurch kein effizienter Filtereffekt bewerkstelligt wird.

**[0074]** Bei Schritt S210 wird die arithmetische Filteroperation durch die obige Gleichung (3) unter Verwendung von  $v_3$  und  $v_5$  durchgeführt, um  $v_3'$  zu berechnen, die Steuerung geht weiter auf Schritt S212, und es wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_2$  und  $v_3$  größer als der konstante Wert  $C_1$  ist, oder nicht. Wenn bestimmt wird, dass der Absolutwert die Differenz zwischen  $v_2$  und  $v_3$  gleich groß oder kleiner als der konstante Wert  $C_1$  ist (NEIN), dann geht die Steuerung weiter auf Schritt S214, die arithmetische Filteroperation wird mittels der obigen Gleichung (2) unter Verwendung von  $v_2$  und  $v_5$  durchgeführt, um  $v_2'$  zu berechnen, und die Steuerung geht auf Schritt S216.

**[0075]** Bei Schritt S216 wird die Deblocking-Filterverarbeitung bei dem Pixel-Array in vertikaler Richtung durchgeführt. Bei dieser Deblocking-Filterverarbeitung, wie später noch beschrieben, wird die Operation in gleicher Weise wie bei den Schritten S306 bis S314 durchgeführt.

**[0076]** Dann geht die Steuerung weiter auf Schritt S218, die Daten der Pixel, die zu dem Gebiet des oberen rechten Abschnittes des Blockes in den Pixeldaten im Puffer gehören, werden an die RGB-Umwandlungseinheit 42 ausgegeben, die Steuerung geht weiter auf Schritt S210, und es wird bestimmt, ob die Prozesse bei den Schritten S204 bis S218 bei allen Pixel-Arrays, die zu dem Gebiet des oberen rechten Abschnittes des Blockes gehören, abgeschlossen wurden, oder nicht. Falls der Prozess bei allen Pixel-Arrays abgeschlossen wurde (JA), dann endet die Abfolge der Prozesse, und es wird der ursprüngliche Prozess wieder aufgenommen. Falls bestimmt wird, dass der Prozess nicht bei allen Pixel-Arrays abgeschlossen wurde (NEIN), dann geht die Steuerung weiter auf Schritt S222, es erfolgt, ge-

maß der bei den Schritten S100 und S112 erhaltenen Startadresse, eine Berechnung der Adresse des nächsten Pixel-Arrays, das zum oberen rechten Abschnitt des Gebietes in dem Block gehört, und die Steuerung geht weiter auf Schritt S202.

**[0077]** Falls andererseits bei Schritt S204 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_4$  und  $v_5$  größer als der konstante Wert  $C_2$  ist (JA), falls bei Schritt S208 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_3$  und  $v_4$  größer als der konstante Wert  $C_1$  ist (JA), und falls bei Schritt S212 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_2$  und  $v_3$  größer als der konstante Wert  $C_1$  ist (JA), dann geht die Steuerung weiter auf Schritt S216.

**[0078]** Der Prozess in horizontaler Richtung des Filterprozesses bei Schritt S106 ist ähnlich den Prozessen bei den Schritten S204 bis S214, und der Prozess in vertikaler Richtung des Filterprozesses bei Schritt S104 ist ähnlich dem Prozess bei Schritt S216.

**[0079]** Nun wird der Filterprozess bei Schritt S104 nachfolgend detailliert Bezug nehmend auf [Fig. 6](#) beschrieben. [Fig. 6](#) ist ein Ablaufdiagramm des Filterprozesses bei Schritt S104.

**[0080]** Der Filterprozess bei Schritt S104 ist die Deblocking-Filterverarbeitung, die bei dem Pixel-Array durchgeführt wird, das zum Gebiet des oberen linken Abschnittes des Blockes gehört. Wenn der Prozess bei Schritt S104 durchgeführt wird, dann geht die Steuerung zuerst auf Schritt S300, wie in [Fig. 6](#) dargestellt.

**[0081]** Bei Schritt **300** wird, gemäß der bei den Schritten S100 und S112 erhaltenen Startadresse, die Adresse des vorderen Pixel-Array im Gebiet des oberen linken Abschnittes des Blockes berechnet, die Steuerung geht weiter auf Schritt S302, die Pixel-daten, die bei der Operation in horizontaler Richtung benötigt werden, und die Pixeldaten, die bei der Operation in vertikaler Richtung in dem Pixel-Array benötigt werden, bei dem die Deblocking-Filterverarbeitung durchzuführen ist, werden in den Puffer gemäß der berechneten Adresse des Pixel-Array eingelesen, und die Steuerung geht weiter auf Schritt S304.

**[0082]** Bei Schritt S304 wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_5$  und  $v_4$  größer als der konstante Wert  $C_2$  ist, oder nicht. Falls bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_5$  und  $v_4$  gleich groß oder kleiner als der konstante Wert  $C_2$  ist (NEIN), dann geht die Steuerung weiter auf Schritt S306. Die arithmetische Filteroperation wird mittels der obigen Gleichung (5) unter Verwendung von  $v_5$  und  $v_4$  durchgeführt, um  $v_5'$  zu berechnen, die Steuerung geht weiter auf Schritt S308, und es wird

bestimmt, ob der Absolutwert der Differenz zwischen  $v_6$  und  $v_5$  größer als der konstante Wert  $C_1$  ist, oder nicht. Falls bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_6$  und  $v_5$  gleich groß oder kleiner als der konstante Wert  $C_1$  ist (NEIN), dann geht die Steuerung auf Schritt S310.

**[0083]** Bei Schritt S310 wird die arithmetische Filteroperation durch die obige Gleichung (6) unter Verwendung von  $v_6$  und  $v_4$  durchgeführt, um  $v_6'$  zu berechnen, die Steuerung geht weiter auf Schritt S312, und es wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_7$  und  $v_6$  größer als der konstante Wert  $C_1$  ist, oder nicht. Wenn bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_7$  und  $v_6$  gleich groß oder kleiner als der konstante Wert  $C_1$  ist (NEIN), dann geht die Steuerung weiter auf Schritt S314, die arithmetische Filteroperation wird mittels der obigen Gleichung (7) unter Verwendung von  $v_7$  und  $v_4$  durchgeführt, um  $v_7'$  zu berechnen, und die Steuerung geht auf Schritt S316.

**[0084]** Bei Schritt S216 wird die Deblocking-Filterverarbeitung bei dem Pixel-Array in vertikaler Richtung durchgeführt. Bei dieser Deblocking-Filterverarbeitung wird die Operation in gleicher Weise wie bei den Schritten S306 bis S314 durchgeführt.

**[0085]** Dann geht die Steuerung weiter auf Schritt S218, die Daten der Pixel, die zu dem Gebiet des oberen linken Abschnittes des Blockes in den Pixel-daten im Puffer gehören, werden an die RGB-Umwandlungseinheit **42** ausgegeben, die Steuerung geht weiter auf Schritt S320, und es wird bestimmt, ob die Prozesse bei den Schritten S304 bis S318 bei allen Pixel-Arrays, die zu dem Gebiet des oberen linken Abschnittes des Blockes gehören, abgeschlossen wurden, oder nicht. Falls der Prozess bei allen Pixel-Arrays abgeschlossen wurde (JA), dann endet die Abfolge der Prozesse, und es wird der ursprüngliche Prozess wieder aufgenommen. Falls bestimmt wird, dass der Prozess nicht bei allen Pixel-Arrays abgeschlossen wurde (NEIN), dann geht die Steuerung weiter auf Schritt S322, es erfolgt eine Berechnung der Adresse des nächsten Pixel-Arrays, das zum oberen linken Abschnitt des Gebietes in dem Block gehört, und zwar basierend auf der bei den Schritten S100 und S112 erhaltenen Startadresse, und die Steuerung geht weiter auf Schritt S302.

**[0086]** Falls andererseits bei Schritt S304 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_5$  und  $v_4$  größer als der konstante Wert  $C_2$  ist (JA), falls bei Schritt S308 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_6$  und  $v_5$  größer als der konstante Wert  $C_1$  ist (JA), und falls bei Schritt S312 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_7$  und  $v_6$  größer als der konstante Wert  $C_1$  ist (JA), dann geht die Steuerung weiter auf Schritt S316.

**[0087]** Der Prozess in horizontaler Richtung des Filterprozesses bei Schritt S108 ist ähnlich den Prozessen bei den Schritten S304 bis S314, und der Prozess in vertikaler Richtung des Filterprozesses bei den Schritten S106, S108 ist ähnlich den Prozessen bei den Schritten S204 bis S214.

**[0088]** Das Prinzip der Deringing-Filterverarbeitung gemäß der Erfindung wird nachfolgend detailliert mit Bezug auf [Fig. 7](#) beschrieben. [Fig. 7](#) zeigt das Prinzip der Deringing-Filterverarbeitung.

**[0089]** In [Fig. 7](#) ist der Wert des Zielpixels, bei dem die Deringing-Filterverarbeitung durchzuführen ist, auf  $v_0$  gesetzt, und die Werte der Pixel benachbart zu dem Zielpixel sind auf  $v_1$  bis  $v_8$  gesetzt, und zwar für das obere linke Pixel, das Pixel oberhalb, das obere rechte Pixel, das linke Pixel, das rechte Pixel, das untere linke Pixel, das Pixel unterhalb, bzw. das untere rechte Pixel.

**[0090]** In diesem Fall wird der Wert  $v_0'$  nach der Deringing-Filterverarbeitung des Zielpixels durch die folgende Gleichung (30) der arithmetischen Filteroperation berechnet.

$$V = (v_1 + v_2 + v_3 + v_4 + v_5 + v_6 + v_7 + v_8)/8 \quad (29)$$

$$v_0' = (v_0 + V)/2 \quad (30)$$

**[0091]** Wenn jedoch eine Flanke erfasst wird, wird  $v_i$  ( $i = 1$  bis  $8$ ) durch  $v_0$  ersetzt.  $v_i$  wird durch  $v_0$  aus dem gleichen Grund wie bei der Deblocking-Filterverarbeitung ersetzt, die nicht durchgeführt wird, wenn eine Flanke erfasst wird. Bei dem in [Fig. 7](#) dargestellten Beispiel werden  $v_7$  und  $v_8$  durch  $v_0$  ersetzt.

**[0092]** Die Deringing-Filterverarbeitung gemäß der Erfindung kann mittels des Verfahrens gemäß dem in [Fig. 8](#) dargestellten Ablaufdiagramm realisiert werden. [Fig. 8](#) ist ein Ablaufdiagramm der Deringing-Filterverarbeitung.

**[0093]** Wenn die Deringing-Filterverarbeitung durch die Post-Filtereinheit **40** durchgeführt wird, geht die Steuerung weiter auf Schritt S400, wie in [Fig. 8](#) dargestellt.

**[0094]** Bei Schritt S400 wird die Adresse des vorderen Zielpixels von der LCDC **36** erhalten, die Steuerung geht weiter auf Schritt S402, die Zielpixel und die Werte  $v_0$  bis  $v_8$  der benachbarten Pixel werden gemäß den Adressen der Zielpixel in den Puffer eingelesen, die Steuerung geht weiter auf Schritt S404, die Variable  $V$  wird auf "0" gesetzt, die Variable  $i$  wird auf "1" gesetzt, und die Steuerung geht weiter auf Schritt S406.

**[0095]** Bei Schritt S406 wird bestimmt, ob der Absolutwert der Differenz zwischen  $v_i$  (der Index  $i$  von  $v$  ist

durch den Wert der Variablen  $i$  festgelegt) und  $v_0$  größer als der konstante Wert  $C_3$  ist. Wenn bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_i$  und  $v_0$  größer als der konstante Wert  $C_3$  ist (JA), dann geht die Steuerung weiter auf Schritt S408. Der konstante Wert  $C_3$  ist auf "16" gesetzt, unter der Annahme, dass der Maximalwert des Pixels "255" beträgt. Falls der konstante Wert  $C_3$  größer als "16" gesetzt ist, dann kann der Wert  $v_i$  des benachbarten Pixels nicht durch den Wert  $v_0$  des Zielpixels ersetzt werden, auch wenn er ersetzt werden sollte. Als Ergebnis kann kein effektiver Filtereffekt erzielt werden. Fall andererseits der konstante Wert  $C_3$  kleiner als "16" gesetzt ist, wird der Wert  $v_i$  des benachbarten Pixels durch den Wert  $v_0$  des Zielpixels ersetzt, auch wenn dies nicht erforderlich ist. Als Ergebnis kann kein effektiver Filterungseffekt erzielt werden.

**[0096]** Bei Schritt S408 wird der Wert  $v_i$  des benachbarten Pixels durch den Wert  $v_0$  des Zielpixels ersetzt, die Steuerung geht weiter auf Schritt S410, der Wert  $v_i$  wird um drei Bit nach rechts verschoben, um eine Division durch "8" durchzuführen, die Steuerung geht weiter auf Schritt S412, der Wert, der durch Addieren des Wertes  $v_i$  des benachbarten Pixels zum Wert der Variablen  $V$  erzielt wird, wird als neuer Wert der Variablen  $V$  gesetzt, die Steuerung geht weiter auf Schritt S414, der Wert, der durch Addieren von "1" zum Wert der Variablen  $i$  erhalten wird, wird als neuer Wert gesetzt und die Steuerung geht weiter auf Schritt S416.

**[0097]** Bei Schritt S416 wird bestimmt, ob der Wert der Variablen  $i$  kleiner als 9 ist, oder nicht. Falls bestimmt wird, dass der Wert der Variablen  $i$  gleich groß oder größer als 9 ist (NEIN), dann geht die Steuerung weiter auf Schritt S418, der Wert  $v_0$  des Zielpixels wird zum Wert der Variablen  $V$  addiert, der Wert, der durch Verschieben der Summe um 1 Bit nach rechts erhalten wird, wird als Wert  $v_0'$  berechnet, nachdem die Deringing-Filterverarbeitung beim Zielpixel durchgeführt wurde, die Steuerung geht weiter auf Schritt S420, und der berechnete Wert  $v_0'$  wird an die RGB-Umwandlungseinheit **42** ausgegeben, und die Steuerung geht weiter auf Schritt S422.

**[0098]** Bei Schritt S422 wird bestimmt, ob die Prozesse von Schritt S402 bis Schritt S420 bei allen Pixeln, die einen einzigen Bildschirm des Bildes bilden, abgeschlossen sind. Falls bestimmt wird, dass alle Pixel vollständig verarbeitet werden (JA), dann endet die Prozessabfolge. Falls dies nicht so bestimmt wurde (NEIN), dann geht die Steuerung weiter auf Schritt S424, und die Adresse des nächsten Zielpixels wird von der LCDC **36** erhalten, und damit geht die Steuerung auf Schritt S402 weiter.

**[0099]** Falls bei Schritt S416 bestimmt wird, dass der Wert der Variablen  $i$  kleiner als 9 ist (JA), dann geht die Steuerung weiter auf Schritt S406.

**[0100]** Falls andererseits bei Schritt S406 bestimmt wird, dass der Absolutwert der Differenz zwischen  $v_1$  und  $v_0$  gleich groß oder kleiner als der konstante Wert  $C_3$  ist (NEIN), dann geht die Steuerung weiter auf Schritt S410.

**[0101]** Nachfolgend wird die Funktionsweise gemäß der ersten Ausführungsform der Erfindung beschrieben.

**[0102]** Als Erstes wird der Fall, bei dem die Deblocking-Filterverarbeitung durchgeführt wird, nachfolgend mit Bezug auf ein Beispiel der Durchführung der Deblocking-Filterverarbeitung bei jedem Pixel des in [Fig. 1](#) dargestellten Pixel-Array PG1 beschrieben.

**[0103]** Wenn die Deblocking-Filterverarbeitung bei jedem Pixel des Pixel-Array PG1 durchgeführt wird, werden die Prozesse bei den Schritten S100, S102, S200 und S202 zuerst durchgeführt, und dann wird die Adresse des Pixel-Array PG1 im Gebiet des oberen rechten Abschnittes des Blockes berechnet, und die Pixeldaten, die bei der Operation in horizontaler Richtung benötigt werden, und die Pixeldaten, die bei der Operation in vertikaler Richtung beim Pixel-Array PG1 benötigt werden, werden gemäß der berechneten Adresse des Pixel-Array PG1 gemeinsam in den Puffer eingelesen. Da das Pixel-Array PG1 die dritte Reihe von oben ist, werden die Pixeldaten, die insgesamt 17 Pixel beinhalten, d. h. die vier von rechts nach links aneinandergrenzenden Pixel ausgehend vom oberen rechten Pixel im Block, die acht Pixel zwei Reihen unterhalb des Vier-Pixel-Array, die vier von rechts nach links aneinandergrenzenden Pixel ausgehend vom unteren rechten Pixel im obigen benachbarten Block, und die Pixel zwei Reihen unterhalb des oberen linken Pixels im rechten benachbarten Block ausgelesen.

**[0104]** Als Erstes wird nach Schritt S204 bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_4$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_4$  und  $v_5$  gleich groß oder kleiner als  $C_2$  ist, dann wird nach Schritt S206 die arithmetische Filteroperation anhand der obigen Gleichung (4) basierend auf  $v_4$  und  $v_5$ , und  $v_4'$  berechnet. Falls bestimmt wird, dass die Deblocking-Filterverarbeitung nicht bei  $v_4$  durchgeführt wird, dann wird die Deblocking-Filterverarbeitung nicht bei den Pixelwerten  $v_2$  und  $v_3$  durchgeführt.

**[0105]** Falls die Deblocking-Filterverarbeitung bei dem Pixelwert  $v_4$  durchgeführt wird, dann wird nach Schritt S208 bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_3$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_3$  und  $v_4$  gleich groß oder kleiner als  $C_1$  ist, dann wird, nach Schritt S210, die arithmetische Filteroperation mittels der obigen Gleichung (3) basierend auf  $v_3$  und  $v_5$  durchgeführt, und  $v_3'$  wird berechnet.

Falls bestimmt wird, dass die Deblocking-Filterverarbeitung nicht bei  $v_3$  durchgeführt wird, dann wird die Deblocking-Filterverarbeitung nicht beim Pixelwert  $v_2$  durchgeführt.

**[0106]** Falls die Deblocking-Filterverarbeitung beim Pixelwert  $v_3$  durchgeführt wird, dann wird, nach Schritt S212, bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_2$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_2$  und  $v_3$  gleich groß oder kleiner als  $C_1$  ist, dann wird, nach Schritt S214, die arithmetische Filteroperation mittels der obigen Gleichung (2) basierend auf  $v_2$  und  $v_5$  durchgeführt, und  $v_2'$  wird berechnet.

**[0107]** Nach den Schritten S216 und S218 wird die Deblocking-Filterverarbeitung in vertikaler Richtung bei den Pixeln der linken Hälfte im Pixel-Array PG1 durchgeführt, und der Prozess wird in der horizontalen Richtung und der vertikalen Richtung abgeschlossen. Dann werden, bei den Pixeldaten im Puffer, die Pixeldaten im Puffer, die zum Gebiet des oberen rechten Abschnittes des Blockes gehören, an die RGB-Umwandlungseinheit 42 ausgegeben.

**[0108]** Somit ist die Deblocking-Filterverarbeitung bei den Pixeln der linken Hälfte des Pixel-Array PG1 abgeschlossen.

**[0109]** Die Prozesse der Schritte S100, S104, S300 und S302 werden als Erstes durchgeführt, und dann wird die Adresse des Pixel-Array PG1 im Gebiet des oberen linken Abschnittes des Blockes berechnet, die Pixeldaten, die bei der Operation in vertikaler Richtung beim Pixel-Array PG1 benötigt werden, werden gemäß der berechneten Adresse des Pixel-Array PG1 gemeinsam in den Puffer eingelesen. Da das Pixel-Array PG1 die dritte Reihe von oben ist, werden die Pixeldaten, die insgesamt 17 Pixel beinhalten, d. h. die vier von links nach rechts aneinandergrenzenden Pixel ausgehend vom oberen linken Pixel im Block, die acht Pixel zwei Reihen unterhalb des Vier-Pixel-Array, die vier von links nach rechts aneinandergrenzenden Pixel ausgehend vom unteren linken Pixel im obigen benachbarten Block, und die Pixel zwei Reihen unterhalb des oberen rechten Pixels im linken benachbarten Block ausgelesen.

**[0110]** Als Erstes wird nach Schritt S304 bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_5$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_5$  und  $v_4$  gleich groß oder kleiner als  $C_2$  ist, dann wird nach Schritt S306 die arithmetische Filteroperation anhand der obigen Gleichung (5) basierend auf  $v_5$  und  $v_4$ , und  $v_5'$  berechnet. Falls bestimmt wird, dass die Deblocking-Filterverarbeitung nicht bei  $v_5$  durchgeführt wird, dann wird die Deblocking-Filterverarbeitung nicht bei den Pixelwerten  $v_6$  und  $v_7$  durchgeführt.

**[0111]** Falls die Deblocking-Filterverarbeitung bei dem Pixelwert  $v_5$  durchgeführt wird, dann wird nach Schritt S308 bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_6$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_6$  und  $v_5$  gleich groß oder kleiner als  $C_1$  ist, dann wird, nach Schritt S310, die arithmetische Filteroperation mittels der obigen Gleichung (6) basierend auf  $v_6$  und  $v_4$  durchgeführt, und  $v_6'$  wird berechnet. Falls bestimmt wird, dass die Deblocking-Filterverarbeitung nicht bei  $v_6$  durchgeführt wird, dann wird die Deblocking-Filterverarbeitung nicht beim Pixelwert  $v_7$  durchgeführt.

**[0112]** Falls die Deblocking-Filterverarbeitung beim Pixelwert  $v_6$  durchgeführt wird, dann wird, nach Schritt S312, bestimmt, ob die Deblocking-Filterverarbeitung beim Pixelwert  $v_7$  durchzuführen ist, oder nicht. Das heißt, falls der Absolutwert der Differenz zwischen  $v_7$  und  $v_6$  gleich groß oder kleiner als  $C_1$  ist, dann wird, nach Schritt S314, die arithmetische Filteroperation mittels der obigen Gleichung (7) basierend auf  $v_7$  und  $v_4$  durchgeführt, und  $v_7'$  wird berechnet.

**[0113]** Nach den Schritten S316 und S318 wird die Deblocking-Filterverarbeitung in vertikaler Richtung bei den Pixeln der rechten Hälfte im Pixel-Array PG1 durchgeführt, und der Prozess wird in der horizontalen Richtung und der vertikalen Richtung abgeschlossen. Dann werden, bei den Pixeldaten im Puffer, die Pixeldaten im Puffer, die zum Gebiet des oberen linken Abschnittes des Blockes gehören, an die RGB-Umwandlungseinheit 42 ausgegeben.

**[0114]** Somit ist die Deblocking-Filterverarbeitung bei den Pixeln der rechten Hälfte des Pixel-Array PG1 abgeschlossen.

**[0115]** Nachfolgend wird der Fall beschrieben, bei dem die Deringing-Filterverarbeitung durchgeführt wird.

**[0116]** Wenn die Deringing-Filterverarbeitung durchgeführt wird, werden zuerst die Prozesse bei den Schritten S400 bis S404 durchgeführt, und dann werden ein Zielpixel und die Werte  $v_0$  bis  $v_8$  der benachbarten Pixel aus dem Puffer ausgelesen, und zwar gemäß der Adresse des Zielpixels.

**[0117]** Dann werden, nachdem die Prozesse der Schritte S406 bis S416 wiederholt wurden, der Wert  $v_i$  eines jeden benachbarten Pixels durch den Wert  $v_0$  eines Zielpixels ersetzt, wenn der Absolutwert der Differenz zwischen  $v_i$  und  $v_0$  größer als der konstante Wert  $C_3$  ist. Im anderen Fall wird der Wert  $v_i$  des benachbarten Pixels unverändert verwendet, der Wert  $v_i$  des benachbarten Pixels wird um 3 Bit nach rechts verschoben, und der Wert, der durch Addieren des Wertes  $v_i$  des benachbarten Pixels zum Wert der Variablen V erhalten wird, wird als neuer Wert der Vari-

ablen V gesetzt. Das heißt der Wert  $v_i$  eines jeden benachbarten Pixels wird durch 8 geteilt, und die Divisionsergebnisse werden aufaddiert. Somit wird beim zuvor erwähnten arithmetischen Prozess ein Durchschnittswert V der benachbarten Pixel erhalten.

**[0118]** Dann wird nach den Schritten S418 und S420 der Wert  $v_0$  eines Zielpixels zum Wert der Variablen V addiert, und die Summe wird um ein einziges Bit nach rechts verschoben, und das Ergebnis wird als Wert  $v_0'$  nach dem Filterprozess des Zielpixels erhalten, und der erhaltene Wert  $v_0'$  wird an die RGB-Umwandlungseinheit 42 ausgegeben. Somit berechnet gemäß der vorliegenden Ausführungsform die Deblocking-Filterverarbeitung die Werte  $v_2'$  bis  $v_7'$  der Pixel, nachdem die Deblocking-Filterverarbeitung bei dem Pixel-Array senkrecht zur Grenze zwischen den benachbarten Blöcken unter Verwendung der arithmetischen Filteroperationsgleichungen (2) bis (7) durchgeführt wurde.

**[0119]** Somit wird, wenn eine arithmetische Filteroperation für das Pixel durchgeführt wird, kein Durchschnittswert einer Anzahl von Pixeln verwendet, sondern es werden lediglich die Werte von zwei Pixeln verwendet. Dadurch kann, im Vergleich zur herkömmlichen Technik, die Deblocking-Filterverarbeitung vereinfacht werden. Beispielsweise kann, wie bei der vorliegenden Ausführungsform, wenn die Deblocking-Filterverarbeitung hardwaremäßig implementiert wird, eine geringere Schaltungsgröße erzielt werden, und wenn diese softwaremäßig realisiert wird, kann die Programmgröße reduziert werden. Außerdem kann, da eine Operation durch eine Addition oder eine Subtraktion unter Verwendung des Wertes eines Pixels und des Wertes eines Pixels im gleichen Pixel-Array und eines Grenzpixel im benachbarten Block, und durch eine Bitverschiebung (Division, bei der ein Divisor ein ganzzahliges Vielfaches von 2 ist), durchgeführt wird, die Deblocking-Filterverarbeitung weiter vereinfacht werden, und es kann in einem gewissen Ausmaß ein effektiver Filtereffekt realisiert werden.

**[0120]** Außerdem wird bei der Deblocking-Filterverarbeitung gemäß der vorliegenden Ausführungsform eine arithmetische Filteroperation für ein Pixel im Pixel-Array senkrecht zur Grenze der benachbarten Blöcke durchgeführt, wenn der Differenzwert zwischen dem Pixel und jedem Pixel im gleichen Pixel-Array und an der Grenze im benachbarten Block die Schwellenwerte  $C_1$  und  $C_2$ , für die konstante Werte verwendet werden, nicht überschreitet.

**[0121]** Da die Schwellenwerte  $C_1$  und  $C_2$  konstante Werte sind, ist es nicht erforderlich, die Schwellenwerte  $C_1$  und  $C_2$  während der Deblocking-Filterverarbeitung zu berechnen. Falls irgendeiner der Differenzwerte der Pixel die Schwellenwerte  $C_1$  und  $C_2$

überschreitet, werden keine arithmetischen Filteroperationen durchgeführt. Daher kann, zusätzlich zu einer weiteren Vereinfachung der Deblocking-Filterverarbeitung, die Deblocking-Filterverarbeitung mit relativ größerer Geschwindigkeit als bei der herkömmlichen Technik durchgeführt werden.

**[0122]** Außerdem ist gemäß der vorliegenden Ausführungsform der Schwellenwert  $C_2$  zur Verwendung beim Vergleich der Differenzwerte zwischen einem Grenzpixel und einem Pixel benachbart zum Grenzpixel größer als der Schwellenwert  $C_1$  festgelegt, der bei einem Vergleich der Differenzwerte zwischen dem Grenzpixel und einem Pixel außer den benachbarten Pixeln verwendet wird.

**[0123]** Somit kann die Deblocking-Filterverarbeitung mit der Eigenschaft durchgeführt werden, dass berücksichtigt wird, dass die durchschnittlichen Helligkeitswerte zwischen Blöcken unterschiedlich sind, wodurch ein effektiverer Filtereffekt realisiert wird.

**[0124]** Außerdem ist gemäß der vorliegenden Ausführungsform der Schwellenwert  $C_2$  zur Verwendung beim Vergleichen der Differenzwerte zwischen einem Grenzpixel und einem Pixel benachbart zum Grenzpixel auf 32 gesetzt, und zwar unter der Annahme, dass der Maximalwert eines Pixels den Wert 255 hat, und der Schwellenwert  $C_1$  zum Vergleichen der Differenzwerte zwischen einem Grenzpixel und einem Pixel außer den benachbarten Pixeln ist auf 16 gesetzt.

**[0125]** Somit kann die Deblocking-Filterverarbeitung mit der Eigenschaft durchgeführt werden, dass berücksichtigt wird, dass die durchschnittlichen Helligkeitswerte zwischen Blöcken unterschiedlich sind, wodurch ein effektiverer Filtereffekt realisiert wird.

**[0126]** Weiter wird bei der Deblocking-Filterverarbeitung gemäß der Erfindung bestimmt, ob eine arithmetische Filteroperation für ein Pixel durchzuführen ist, oder nicht. Falls bestimmt wird, dass der Prozess in der entfernten Richtung von einem Grenzpixel und einem Pixel benachbart zum Grenzpixel durchgeführt wird, und dass keine arithmetische Filteroperation für Pixel durchgeführt wird, dann wird keine arithmetische Filteroperation für das Pixel und die anschließenden Pixel durchgeführt.

**[0127]** Somit kann, da die Häufigkeit der Bestimmung, ob eine arithmetische Filteroperation für ein Pixel durchzuführen ist, oder nicht, reduziert werden kann, die Deblocking-Filterverarbeitung bei einer höheren Geschwindigkeit durchgeführt werden.

**[0128]** Gemäß der vorliegenden Ausführungsform beinhaltet die Post-Filtereinheit **40** einen Puffer, der in der Lage ist, Pixeldaten von 17 Pixeln zu speichern, und die Deblocking-Filterverarbeitung wird durch gemeinsames Auslesen aller Pixeldaten in ei-

nem Pixel-Array von einem Pixel bis zu einem Grenzpixel aus dem VRAM **35** in den Puffer, und durch Ausführen einer arithmetischen Filteroperation durchgeführt.

**[0129]** Somit kann, da die erforderlichen Pixeldaten bei der Durchführung der Deblocking-Filterverarbeitung bei jedem Pixel im Pixel-Array gemeinsam in den Puffer eingelesen und verarbeitet werden können, die Deblocking-Filterverarbeitung mit höherer Geschwindigkeit durchgeführt werden.

**[0130]** Außerdem wird bei der Deblocking-Filterverarbeitung gemäß der vorliegenden Ausführungsform eine arithmetische Filteroperation für ein Pixel-Array entweder in der horizontalen Richtung oder in der vertikalen Richtung des Blockes durchgeführt wird, und wird dann beim Pixel-Array in der jeweils anderen Richtung durchgeführt.

**[0131]** Da die Deblocking-Filterverarbeitung bei den Pixel-Arrays sowohl in horizontaler als auch in vertikaler Richtung des Blockes durchgeführt wird, kann ein effektiverer Filtereffekt realisiert werden.

**[0132]** Außerdem wird bei der Deringing-Filterverarbeitung gemäß der vorliegenden Ausführungsform eine bei einem Pixel durchgeführte arithmetische Filteroperation mit dem Wert des Zielpixels, bei dem die Operation durchzuführen ist, und mit dem Wert des zum Zielpixel benachbarten Pixels durchgeführt. Falls der Differenzwert zwischen einem der benachbarten Pixel und einem Zielpixel den Schwellenwert  $C_3$  bei der arithmetischen Filteroperation übersteigt, dann kann der Wert des Pixels durch den Wert des Zielpixels ersetzt werden.

**[0133]** Somit ist ungeachtet der Tatsache, ob der Differenzwert zwischen jedem benachbarten Pixel und einem Zielpixel den Schwellenwert  $C_3$  überschreitet, der Divisor, der bei der Berechnung eines Durchschnittswertes eines benachbarten Pixels verwendet wird, konstant (8), der Divisor ist ein ganzzahliges Vielfaches von 2, und die Operation kann durch eine Bitverschiebung durchgeführt werden. Daher kann, im Vergleich zur herkömmlichen Technik, die Deringing-Filterverarbeitung vereinfacht werden. Beispielsweise kann, wie bei der vorliegenden Ausführungsform, wenn die Deringing-Filterverarbeitung hardwaremäßig implementiert ist, eine geringere Schaltungsgröße realisiert werden, und wenn die Deringing-Filterverarbeitung softwaremäßig implementiert ist, kann die Programmgröße reduziert werden.

**[0134]** Außerdem wird gemäß der vorliegenden Ausführungsform bei der Deringing-Filterverarbeitung ein konstanter Wert als Schwellenwert  $C_3$  verwendet.

**[0135]** Somit ist es, da der Schwellenwert  $C_3$  ein

konstanter Wert ist, nicht erforderlich, den Schwellenwert  $C_3$  während der Deringing-Filterverarbeitung zu berechnen. Daher kann die Deringing-Filterverarbeitung weiter vereinfacht werden.

**[0136]** Gemäß der vorliegenden Ausführungsform ist der Schwellenwert  $C_3$  auf 16 gesetzt, und zwar unter der Annahme, dass der Maximalwert des Pixels bei der Deringing-Filterverarbeitung den Wert 255 hat.

**[0137]** Somit kann in einem gewissen Ausmaß ein effektiver Filtereffekt realisiert werden.

**[0138]** Außerdem wird bei der Deringing-Filterverarbeitung gemäß der vorliegenden Ausführungsform die für ein Pixel durchgeführte arithmetische Filteroperation dadurch durchgeführt, dass der Wert eines Zielpixels, bei dem die Operation durchgeführt werden soll, durch 8 dividiert wird, der Wert des zum Zielpixel benachbarten Pixels durch 8 dividiert wird, und die Divisionsergebnisse aufaddiert werden.

**[0139]** Da die Addition nach der Division durchgeführt wird, wird im Vergleich zur herkömmlichen Technik zur Durchführung einer Operation bei der Deringing-Filterverarbeitung weniger Speicher benötigt.

**[0140]** Gemäß der ersten Ausführungsform entsprechen die Bewegtbilddaten im MPEG-Format den komprimierten Bilddaten gemäß Anspruch 1, das VRAM 35 repräsentiert eine Bildspeichereinrichtung, und der Puffer der Post-Filtereinheit 40 repräsentiert die Arbeitsspeichereinrichtung.

**[0141]** Bei der zuvor beschriebenen ersten Ausführungsform werden die Prozesse in dem in den [Fig. 4](#), [Fig. 5](#), [Fig. 6](#) und [Fig. 8](#) dargestellten Ablaufdiagramm durch die Hardware der Post-Filtereinheit 40 durchgeführt. Jedoch ist die Erfindung nicht auf diese Anwendung eingeschränkt. Beispielsweise kann die CPU 30 diese Prozesse durchführen. In diesem Fall kann ein vorab im ROM 32 gespeichertes Steuerprogramm wie bei der ersten Ausführungsform ausgeführt werden, und das Programm der Prozedur kann aus dem Speichermedium ausgelesen werden, indem man das Programm zur Ausführung im RAM 34 speichert.

**[0142]** Speichermedium bezieht sich auf ein Halbleiterspeichermedium wie beispielsweise ein RAM, ein ROM, etc., ein magnetisches Speichermedium, wie beispielsweise eine FD, eine HD, etc., und ein Speichermedium eines optischen Lesesystems, wie beispielsweise eine CD, eine CDV, eine LD, eine DVD, etc., und ein Speichermedium vom magnetischen Speichertyp/optischen Lesesystem, wie beispielsweise eine MO, etc., und ungeachtet eines elektronischen, magnetischen oder optischen Lesesystems kann ein beliebiges computerlesbares Medi-

um verwendet werden.

**[0143]** Gemäß der ersten Ausführungsform wird die Bildverarbeitungseinrichtung, das Bildverarbeitungsprogramm und das Bildverarbeitungsverfahren angewandt, wenn die Deblocking-Filterverarbeitung und die Deringing-Filterverarbeitung bei dem wiedergegebenen Bild durchgeführt werden, und zwar bei der Wiedergabe eines Bewegtbildes durch Decodieren von im MPEG-Format komprimierten Bewegtbilddaten durch den Computer 100 mit dem im Multitasking-Betrieb arbeitenden OS, wie dargestellt in [Fig. 1](#). Jedoch ist die Erfindung nicht auf diese Anwendung eingeschränkt, ohne vom Schutzzumfang der Erfindung abzuweichen. Beispielsweise kann die Erfindung beim Anzeigen eines Bildes, das auf im JPEG-Format komprimierten Bilddaten basiert, angewandt werden, oder wenn ein Bild in einer Blockeinheit basierend auf den komprimierten Bilddaten entwickelt wird, die beim Bildkomprimierungsprozess komprimiert wurden, bei dem die Diskrete Kosinustransformation und die Quantisierungstransformation in einer vorbestimmten Blockeinheit durchgeführt werden kann.

**[0144]** Bei der Bildverarbeitungseinrichtung werden, wenn der Filterprozess bei einem einzigen Pixel durchgeführt wird, der Wert des Pixels und der Wert eines Grenzpixel in demselben Pixel-Array und im benachbarten Block verwendet. Daher kann in einem gewissen Ausmaß ein effektiver Filtereffekt realisiert werden.

**[0145]** Außerdem wird bei der Bildverarbeitungseinrichtung, wenn der Filterprozess bei einem einzigen Pixel durchgeführt wird, eine Operation durch eine Addition und eine Subtraktion und durch Verschieben von Bits (Division durch einen Divisor eines Vielfachen von 2) unter Verwendung des Wertes des Pixels und des Wertes eines Grenzpixels in demselben Pixel-Array und im benachbarten Block durchgeführt. Daher kann der Filterprozess zur Verringerung der Blockartefakte weiter vereinfacht werden. Beim Decodieren eines Bildes kann ein effektiverer Filtereffekt realisiert werden.

**[0146]** Außerdem ist bei der Bildverarbeitungseinrichtung ein Schwellenwert ein konstanter Wert. Daher ist es nicht erforderlich, den Schwellenwert während des Filterprozesses zu berechnen. Falls irgendeiner der Differenzwerte zwischen Pixeln den Schwellenwert überschreitet, wird keine arithmetische Filteroperation durchgeführt. Daher kann der Filterprozess zur Verringerung von Blockartefakten beim Decodieren eines Bildes weiter vereinfacht werden, und, im Vergleich zur herkömmlichen Technik, kann der Filterprozess mit relativ größerer Geschwindigkeit durchgeführt werden.

**[0147]** Außerdem kann in der Bildverarbeitungsein-

richtung der Filterprozess mit der Eigenschaft durchgeführt werden, dass berücksichtigt wird, dass die durchschnittlichen Helligkeitswerte zwischen Blöcken unterschiedlich sind. Daher kann in einem gewissen Ausmaß ein effektiver Filtereffekt realisiert werden.

**[0148]** Außerdem kann bei der Bildverarbeitungseinrichtung die Häufigkeit der Bestimmung, ob die arithmetische Filteroperation für ein Pixel durchzuführen ist, oder nicht, reduziert werden. Dadurch kann beim Decodieren eines Bildes der Filterprozess zur Verminderung des Blockrauschens mit größerer Geschwindigkeit durchgeführt werden.

**[0149]** Außerdem werden bei der Bildverarbeitungseinrichtung die benötigten Bilddaten beim Filterprozess eines jeden Pixels in einem Pixel-Array gemeinsam in die Arbeitsspeichereinrichtung eingelesen und in einer einzigen Einheit verarbeitet. Daher kann, im Vergleich zur herkömmlichen Technik, beim Decodieren eines Bildes der Filterprozess zur Reduzierung der Blockartefakte mit relativ hoher Geschwindigkeit durchgeführt werden.

**[0150]** Weiter wird bei der Bildverarbeitungseinrichtung der Filterprozess sowohl bei einem Pixel-Array in horizontaler Richtung eines Blockes und einem Pixel-Array in vertikaler Richtung eines Blockes durchgeführt. Daher kann in einem gewissen Ausmaß ein effektiver Filtereffekt realisiert werden.

**[0151]** Außerdem ist bei der Bildverarbeitungseinrichtung, wenn ein Durchschnittswert von benachbarten Pixeln bei einer arithmetischen Filteroperation verwendet wird, ein Divisor zur Verwendung bei der Berechnung eines Durchschnittswertes benachbarter Pixel konstant, ungeachtet der Tatsache, ob die Differenz zwischen jedem benachbarten Pixel und einem Zielpixel einen Schwellenwert überschreitet, oder nicht, und eine Operation kann durch eine Bitverschiebung durchgeführt werden, wenn der Divisor ein ganzzahliges Vielfaches von 2 ist. Daher kann, im Vergleich zur herkömmlichen Technik, beim Decodieren eines Bildes der Filterprozess zur Verminderung des Moskito-Effektes relativ vereinfacht werden.

**[0152]** Außerdem werden bei der Bildverarbeitungseinrichtung die benötigten Bilddaten beim Filterprozess eines jeden Pixels in einem Pixel-Array gemeinsam in die Arbeitsspeichereinrichtung eingelesen und in einer einzigen Einheit verarbeitet. Daher kann, im Vergleich zur herkömmlichen Technik, beim Decodieren eines Bildes der Filterprozess zur Reduzierung der Blockartefakte mit relativ hoher Geschwindigkeit durchgeführt werden.

### Patentansprüche

1. Bildverarbeitungseinrichtung zur Durchfüh-

rung einer Filterverarbeitung bei einem expandierten Bild, nachdem das Bild in Einheiten von vorbestimmten Blöcken expandiert wurde, die auf komprimierten Bilddaten basieren, welche durch eine Bildkompression komprimiert wurden, bei der eine Diskrete Kosinustransformation und eine Quantisierung in Einheiten von Blöcken durchgeführt wurden, **dadurch gekennzeichnet**, dass der Filterprozess derart durchgeführt wird, dass für ein Pixel-Array, das eine Grenze zwischen benachbarten Blöcken schneidet, wenn Pixelwerte auf der einen Seite der Grenze als  $v_4$ ,  $v_3$  und  $v_2$  in der Reihenfolge ihrer Nähe zur Grenze bezeichnet sind, und Pixelwerte auf der gegenüberliegenden Seite der Grenze als  $v_5$ ,  $v_6$  und  $v_7$  in der Reihenfolge ihrer Nähe zur Grenze bezeichnet sind, Pixelwerte  $v_2'$  bis  $v_7'$  nach dem Filtern durch eine arithmetische Filteroperation berechnet werden, und zwar durch Anwenden folgender arithmetischer Gleichungen:

$$v_2' = (v_2 + v_5)/2 + (v_2 - v_5)/4,$$

$$v_3' = (v_3 + v_5)/2 + (v_3 - v_5)/8,$$

$$v_4' = (v_4 + v_5)/2,$$

$$v_5' = (v_5 + v_4)/2 + (v_5 - v_4)/8,$$

$$v_6' = (v_6 + v_4)/2 + (v_6 - v_4)/4, \text{ and}$$

$$v_7' = v_7 - (v_7 - v_4)/8.$$

2. Bildverarbeitungseinrichtung nach Anspruch 1, bei der die arithmetische Filteroperation für jedes Pixel in einem Pixel-Array, das die Grenze zwischen benachbarten Blöcken schneidet, durchgeführt wird, wenn keine Differenz zwischen dem jeweiligen Pixel und einem beliebigen der Pixel, von dem jeweiligen Pixel bis zu einem Grenzpixel, das an der Grenze in demselben Pixel-Array im benachbarten Block positioniert ist, einen Schwellenwert überschreitet, und ein konstanter Wert als der Schwellenwert verwendet wird.

3. Bildverarbeitungseinrichtung nach Anspruch 2, bei welcher der Schwellenwert für einen Vergleich mit der Differenz zwischen dem Grenzpixel und einem benachbarten Pixel, das sich benachbart zum Grenzpixel befindet, größer festgelegt ist als der Schwellenwert für einen Vergleich mit der Differenz zwischen dem Grenzpixel und einem Pixel außer diesem benachbarten Pixel.

4. Bildverarbeitungseinrichtung nach Anspruch 2 oder 3, bei welcher der Schwellenwert zum Vergleichen der Differenz zwischen dem Grenzpixel und einem benachbarten Pixel, das sich benachbart zum Grenzpixel befindet, gleich " $2^{n-3}$ " ist, während das jeweilige Pixel einen maximalen Wert von " $2^n-1$ " hat, mit  $n \geq 4$ , und der Schwellenwert zum Vergleichen

der Differenz zwischen dem Grenzpixel und einem Pixel außer dem benachbarten Pixel gleich " $2^{n-4}$ " ist.

5. Bildverarbeitungseinrichtung nach einem der Ansprüche 2 bis 4, bei der eine Beurteilung sequentiell pixelweise vorgenommen wird, ob die arithmetische Filteroperation für ein jeweiliges Pixel durchgeführt werden sollte, wobei die Beurteilung mit einem benachbarten Pixel beginnt, das sich benachbart zum Grenzpixel befindet, und sich dann sequentiell in einer Richtung von der Grenze weg bewegt, und, wenn beurteilt wird, dass die arithmetische Filteroperation für ein jeweiliges Pixel nicht durchgeführt werden sollte, die arithmetische Filteroperation für dieses Pixel und die in der Sequenz folgenden Pixel nicht durchgeführt wird.

6. Bildverarbeitungseinrichtung nach einem der Ansprüche 1 bis 5, welche weiter eine Arbeitsspeichereinrichtung aufweist, die mit einer Bildspeichereinrichtung zum Speichern des expandierten Bildes verbunden ist, wobei die Arbeitsspeichereinrichtung eine ausreichende Speicherkapazität zum Speichern von Daten aller Pixel aufweist, die von dem einen Pixel, das in einem Pixel-Array, das die Grenze zwischen benachbarten Blöcken schneidet, am weitesten weg von der Grenze liegt, bis zu einem Grenzpixel reichen, das in demselben Pixel-Array liegt und an der Grenze in einem benachbarten Block positioniert ist, die Filterverarbeitung derart durchgeführt wird, dass Daten aller Pixel im Bereich von dem einen Pixel bis zum Grenzpixel in dem Pixel-Array aus der Bildspeichereinrichtung in die Arbeitsspeichereinrichtung in integraler Weise eingelesen werden, und die arithmetische Filteroperation basierend auf den Pixeldaten in der Arbeitsspeichereinrichtung durchgeführt wird.

7. Bildverarbeitungseinrichtung nach einem der Ansprüche 1 bis 6, bei der die arithmetische Filteroperation entweder für ein Pixel-Array in horizontaler Richtung oder ein Pixel-Array in einer vertikalen Richtung des Blockes durchgeführt wird, und anschließend in der jeweils anderen Richtung durchgeführt wird.

8. Bildverarbeitungseinrichtung nach einem der Ansprüche 1 bis 7, bei der die Quantisierung als eine Verarbeitung durchgeführt wird, um eine grobe Quantisierung der hochfrequenten Komponenten der Ergebnisse der Diskreten Kosinustransformation vorzunehmen, und die Filterverarbeitung derart durchgeführt wird, dass die arithmetische Filteroperation für ein Pixel basierend auf dem Wert eines zu berechnenden Zielpixels und den Werten benachbarter Pixel, die zu dem Zielpixel benachbart sind, durchgeführt wird, und bei der arithmetischen Filteroperation, wenn die Differenz zwischen dem jeweiligen Pixel und dem Ziel-

pixel der benachbarten Pixel einen Schwellenwert überschreitet, der Wert des Zielpixels anstelle des Wertes des jeweiligen Pixels verwendet wird.

9. Bildverarbeitungseinrichtung nach Anspruch 8, bei der die Filterverarbeitung unter Verwendung eines konstanten Wertes als Schwellenwert durchgeführt wird.

10. Bildverarbeitungseinrichtung nach Anspruch 9, bei welcher der Schwellenwert gleich " $2^{n-4}$ " ist, wenn das Pixel einen Maximalwert von " $2^n-1$ " hat, mit  $n \geq 4$ .

11. Bildverarbeitungseinrichtung nach einem der Ansprüche 1 bis 10, bei der die Quantisierung als eine Verarbeitung durchgeführt wird, um eine grobe Quantisierung der hochfrequenten Komponenten der Ergebnisse der Diskreten Kosinustransformation vorzunehmen, und die arithmetische Filteroperation für ein Pixel basierend auf dem Wert eines zu berechnenden Zielpixels und den Werten von vier, zwei oder einem benachbarten Pixel, die zu dem Zielpixel benachbart sind, durchgeführt wird.

12. Bildverarbeitungseinrichtung nach einem der Ansprüche 1 bis 11, bei der die Quantisierung als eine Verarbeitung durchgeführt wird, um eine grobe Quantisierung der hochfrequenten Komponenten der Ergebnisse der Diskreten Kosinustransformation vorzunehmen, und die arithmetische Filteroperation für ein Pixel durch Dividieren des Wertes eines zu berechnenden Zielpixels durch einen vorbestimmten Wert, durch Dividieren des Wertes eines benachbarten Pixels, der zu dem Zielpixel benachbart ist, durch den vorbestimmten Wert, und Addieren der Divisionsergebnisse durchgeführt wird.

13. Bildverarbeitungseinrichtung nach einem der vorhergehenden Ansprüche, die eine Arbeitsspeichereinrichtung aufweist, die mit einer Bildspeichereinrichtung zum Speichern des expandierten Bildes verbunden ist und die eine ausreichende Speicherkapazität zum Speichern von Daten aller Pixel im Bereich von dem einen Pixel, das in einem Pixel-Array, das die Grenze zwischen benachbarten Blöcken schneidet, am weitesten weg von der Grenze liegt bis zu einem Grenzpixel, das in demselben Pixel-Array liegt und an der Grenze in einem benachbarten Block positioniert ist, aufweist wobei die Filterverarbeitung derart durchgeführt wird, dass Daten aller Pixel von dem einen Pixel bis zu dem Grenzpixel in dem Pixel-Array aus der Bildspeichereinrichtung in die Arbeitsspeichereinrichtung in integraler Weise eingelesen werden, und die arithmetische Filteroperation basierend auf den Pixeldaten in der Arbeitsspeichereinrichtung durchgeführt wird.

14. Bildverarbeitungsverfahren zur Durchführung einer Filterverarbeitung bei einem expandierten Bild, nachdem das Bild in Einheiten von vorbestimmten Blöcken expandiert wurde, die auf komprimierten Bilddaten basieren, die durch eine Bildkompression komprimiert wurden, bei der eine Diskrete Kosinustransformation und eine Quantisierung in den Einheiten von Blöcken durchgeführt wurden, dadurch gekennzeichnet, dass der Filterprozess derart durchgeführt wird, dass für ein Pixel-Array, das eine Grenze zwischen benachbarten Blöcken schneidet, wenn Pixelwerte auf der einen Seite der Grenze als  $v_4$ ,  $v_3$  und  $v_2$  in der Reihenfolge ihrer Nähe zur Grenze bezeichnet sind, und Pixelwerte auf der gegenüberliegenden Seite der Grenze als  $v_5$ ,  $v_6$  und  $v_7$  in der Reihenfolge ihrer Nähe zur Grenze bezeichnet sind, Pixelwerte  $v_2'$  bis  $v_7'$  nach dem Filtern durch eine arithmetische Filteroperation berechnet werden, und zwar durch Anwenden folgender arithmetischer Gleichungen:

$$v_2' = (v_2 + v_5)/2 + (v_2 - v_5)/4,$$

$$v_3' = (v_3 + v_5)/2 + (v_3 - v_5)/8,$$

$$v_4' = (v_4 + v_5)/2,$$

$$v_5' = (v_5 + v_4)/2 + (v_5 - v_4)/8,$$

$$v_6' = (v_6 + v_4)/2 + (v_6 - v_4)/4, \text{ und}$$

$$v_7' = (v_7 + v_4)/8.$$

15. Verfahren nach Anspruch 14, bei dem die arithmetische Filteroperation für ein Pixel basierend auf dem Wert eines zu berechnenden Zielpixels und den Werten benachbarter Pixel, die zu dem Zielpixel benachbart sind, durchgeführt wird, und bei der arithmetischen Filteroperation, wenn die Differenz zwischen dem jeweiligen Pixel und dem Zielpixel der benachbarten Pixel einen Schwellenwert überschreitet, der Wert des Zielpixels anstelle des Wertes des jeweiligen Pixels verwendet wird.

16. Verfahren nach Anspruch 14, bei dem die arithmetische Filteroperation für ein jeweiliges Pixel basierend auf dem Wert eines zu berechnenden Zielpixels und den Werten von vier, zwei oder einem benachbarten Pixel, die zu dem Zielpixel benachbart sind, durchgeführt wird.

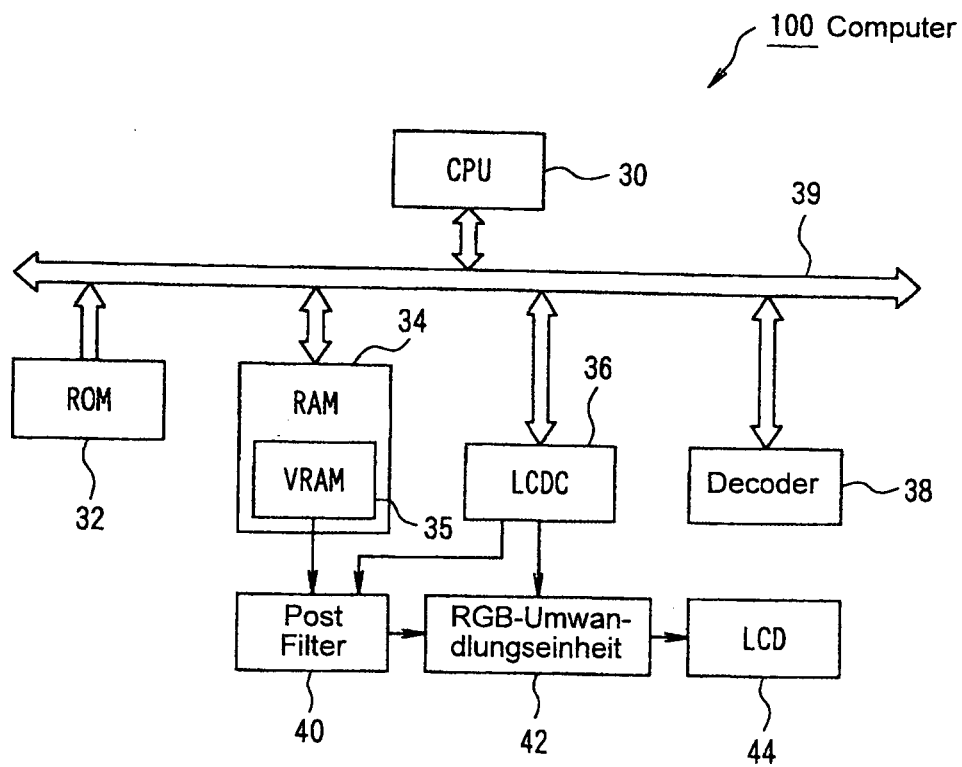
17. Verfahren nach Anspruch 14, bei dem die arithmetische Filteroperation für ein jeweiliges Pixel durch Dividieren des Wertes eines zu berechnenden Zielpixels durch einen vorbestimmten Wert, durch Dividieren des Wertes eines benachbarten Pixels, der zu dem Zielpixel benachbart ist, durch den vorbestimmten Wert, und Addieren der Divisionsergebnisse durchgeführt wird.

18. Maschinenlesbares Medium, bei dem das

Medium ein Programm aus durch die Maschine ausführbaren Anweisungen trägt, um das Verfahren nach einem der Ansprüche 14 bis 17 auszuführen.

Es folgen 9 Blatt Zeichnungen

FIG. 1



*FIG. 2*

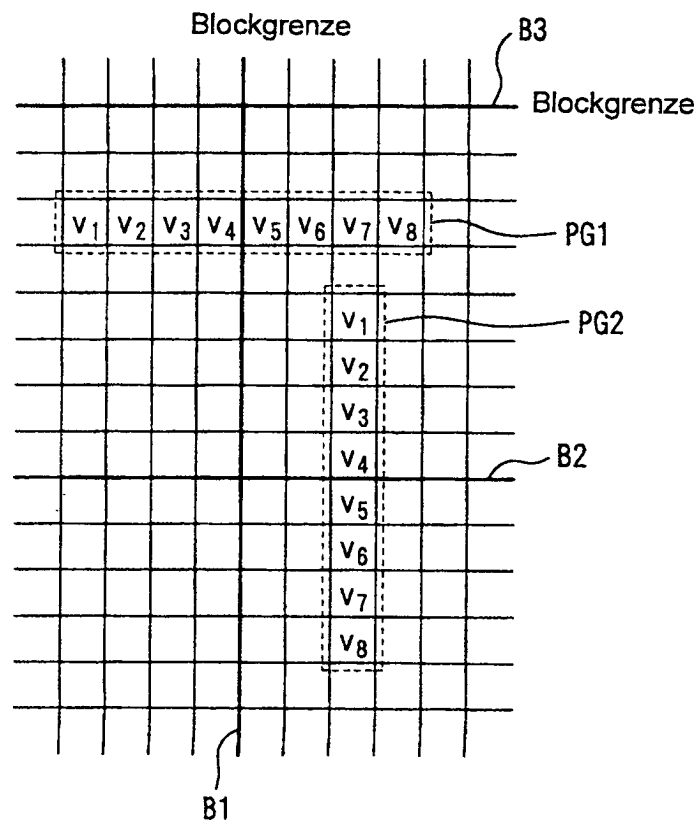


FIG. 3 (A)

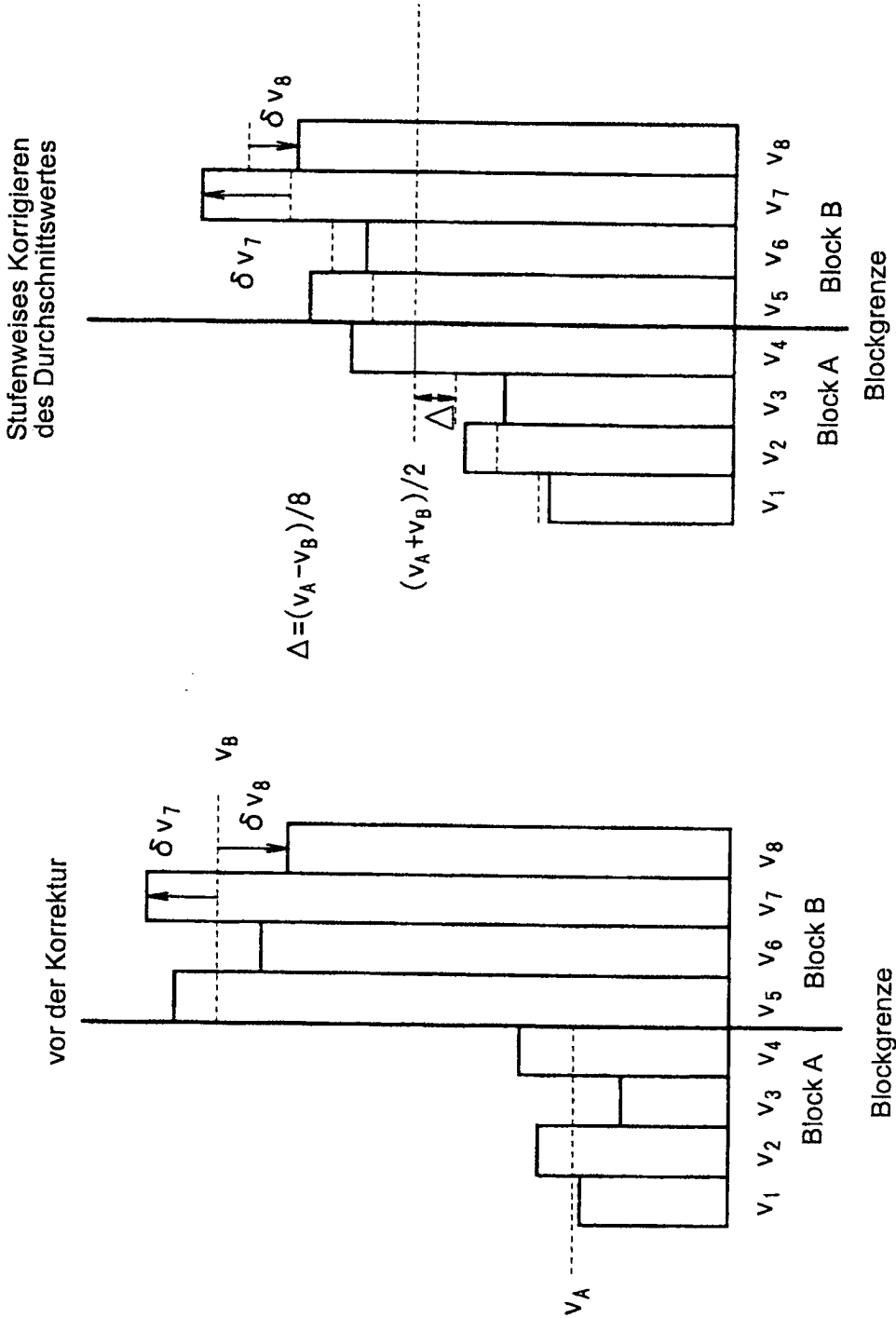


FIG. 3 (B)

FIG. 4

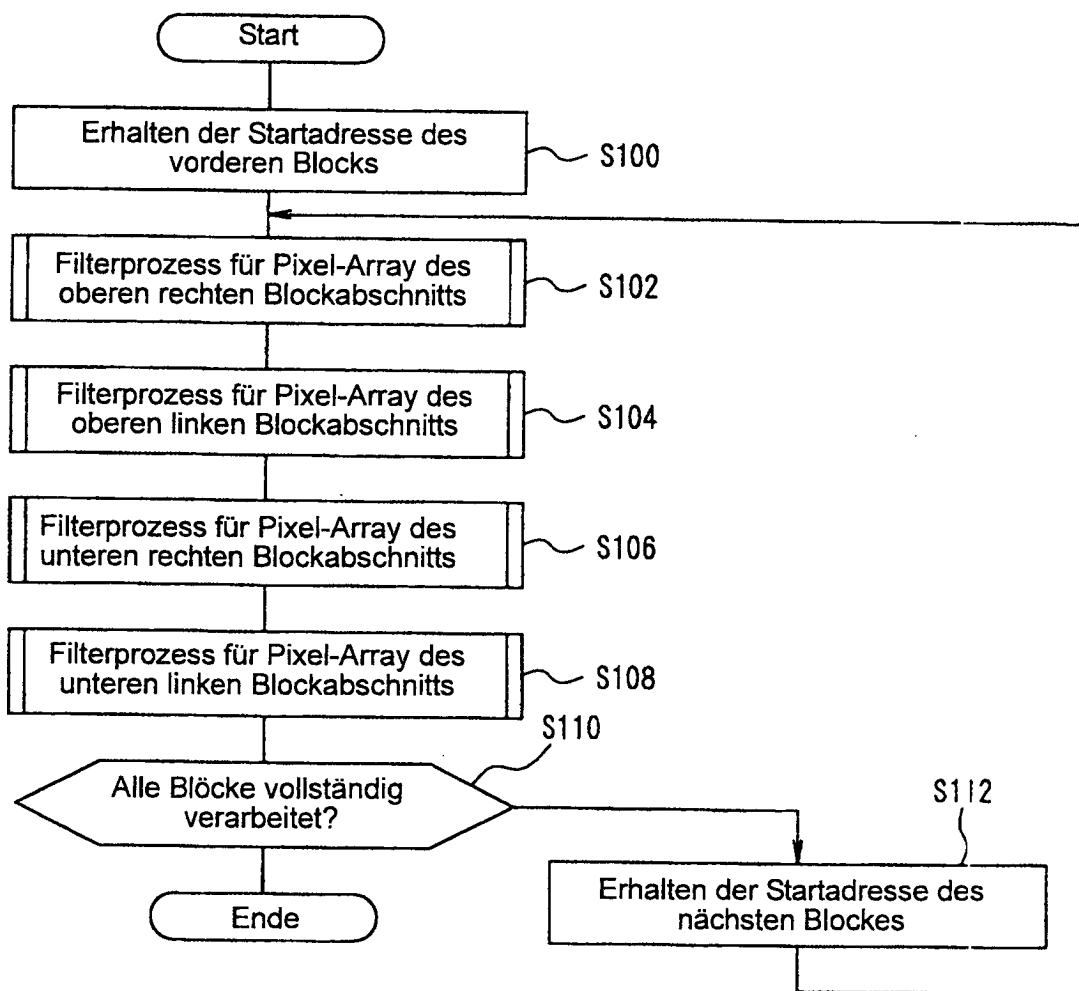


FIG. 5

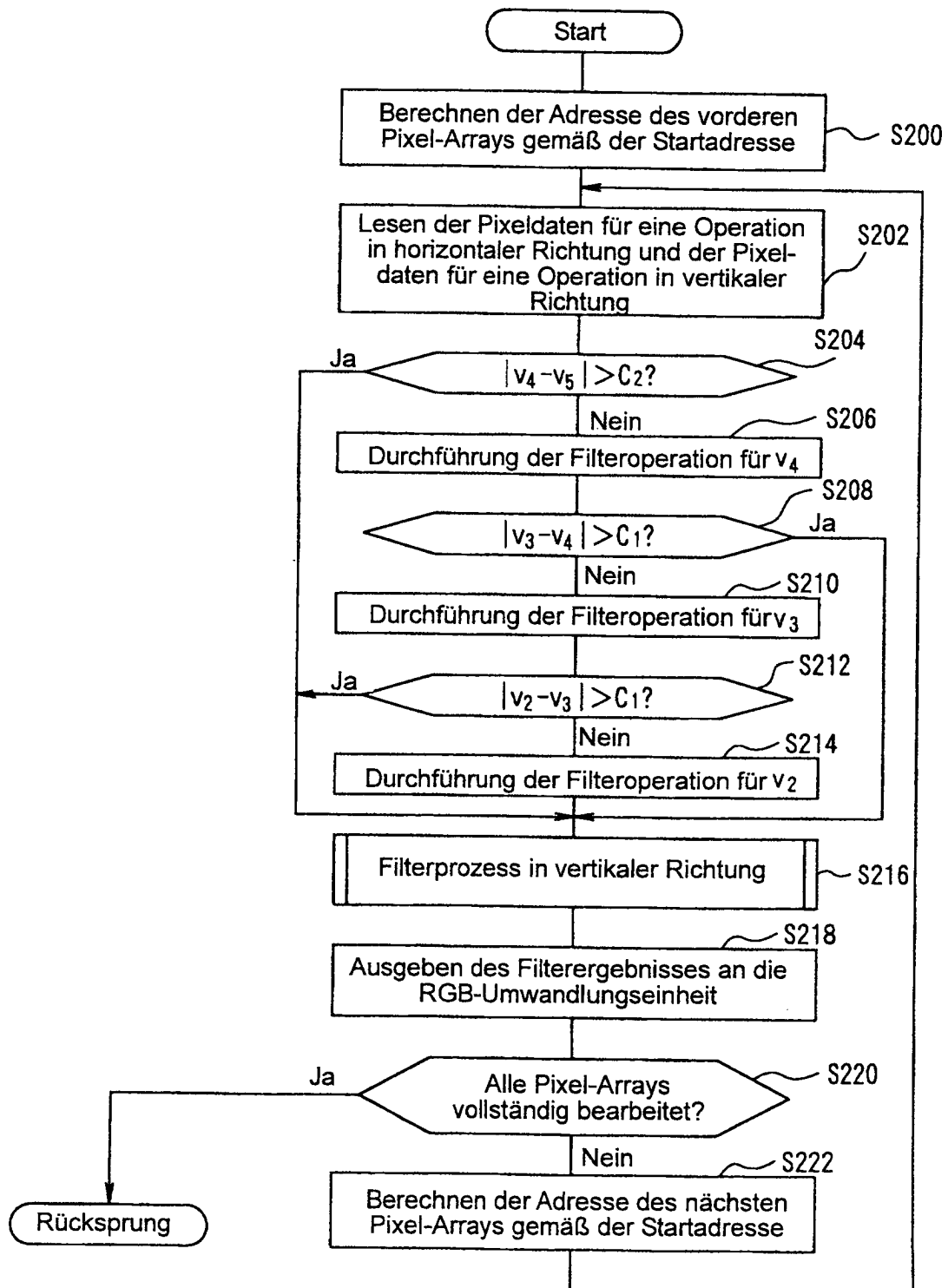
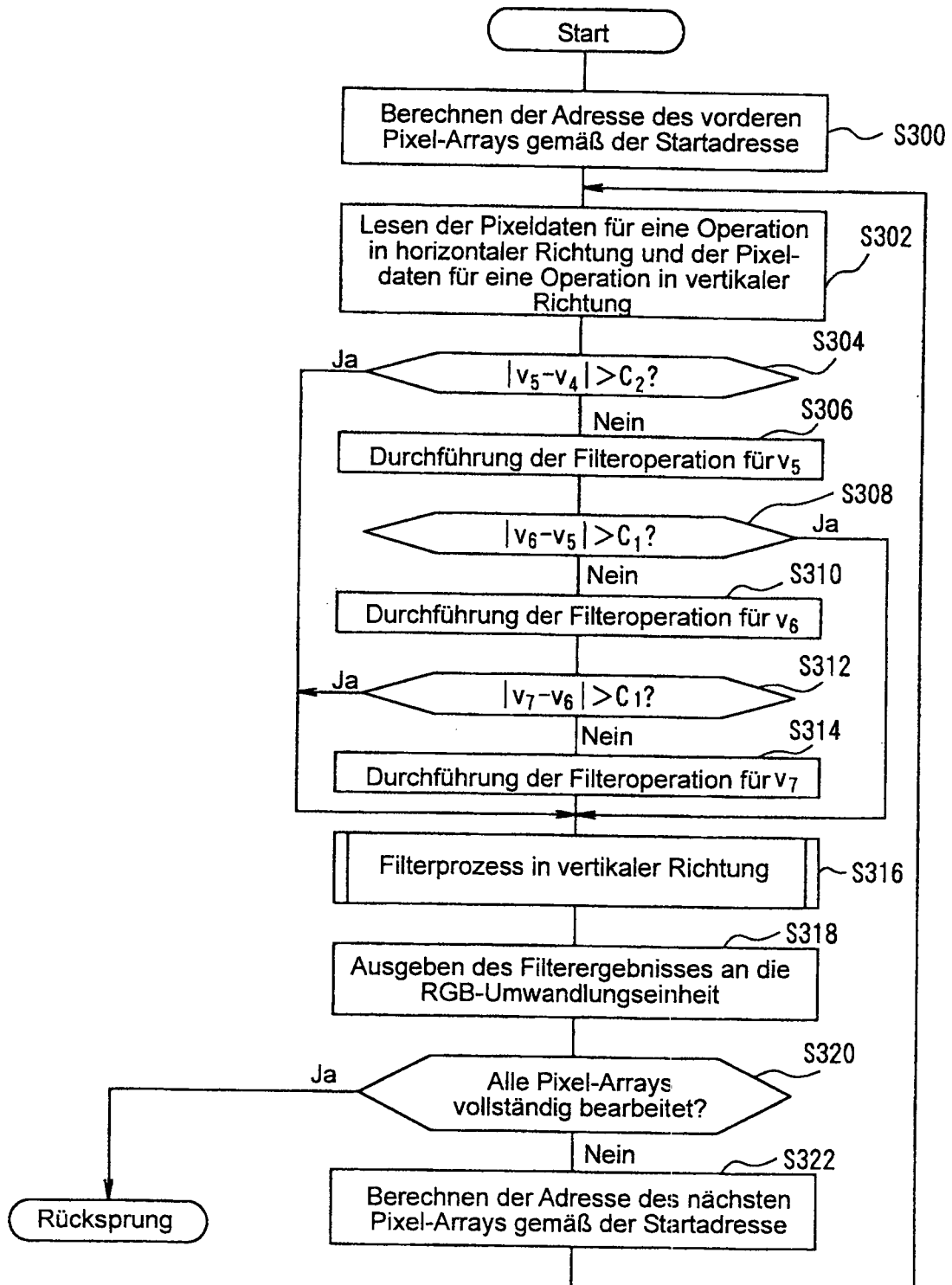


FIG. 6



*FIG. 7*

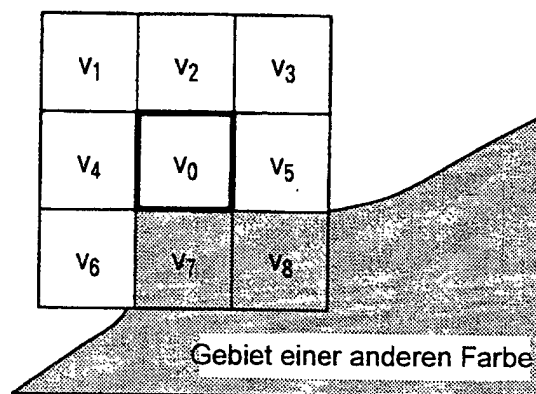


FIG. 8

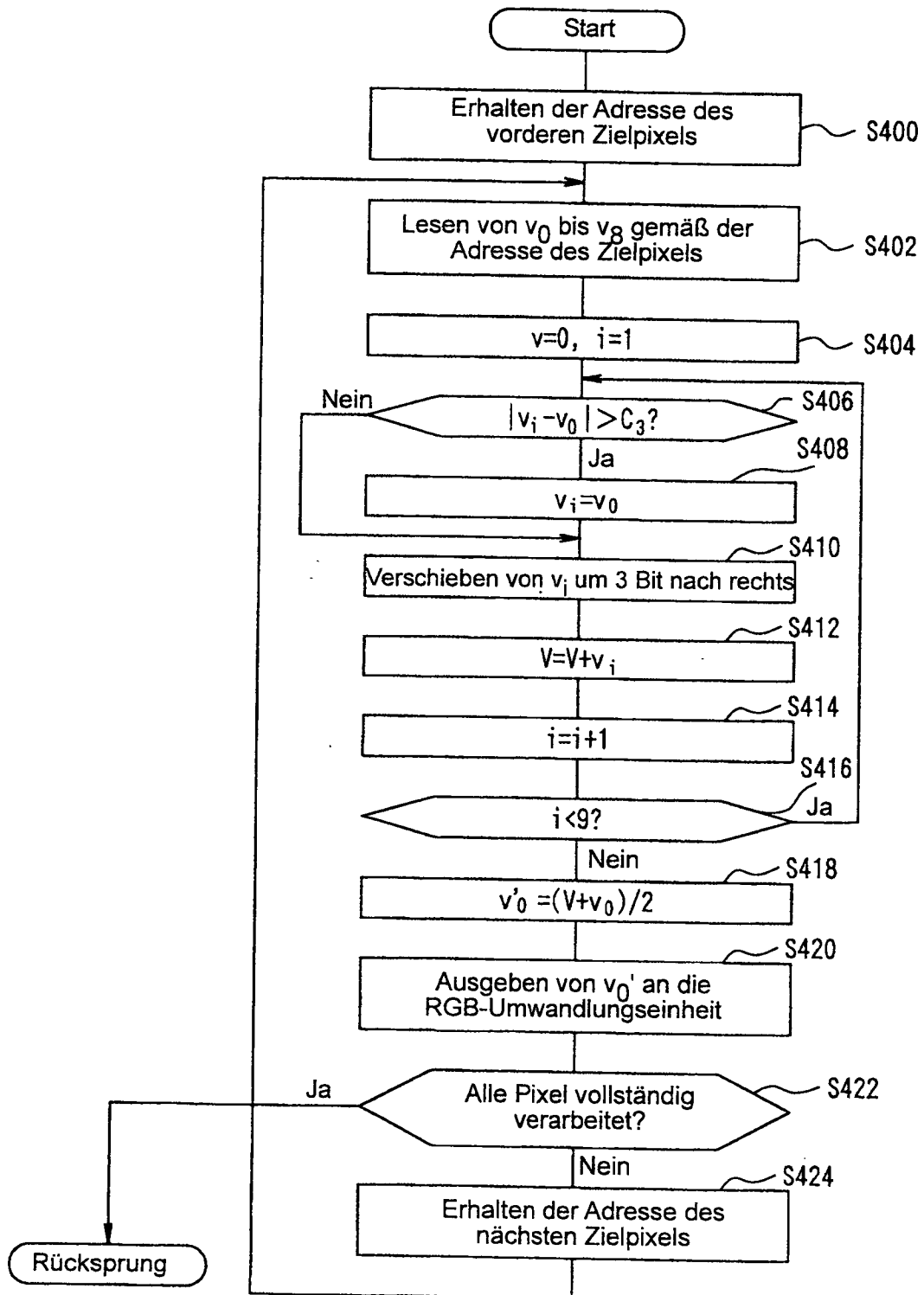


FIG. 9

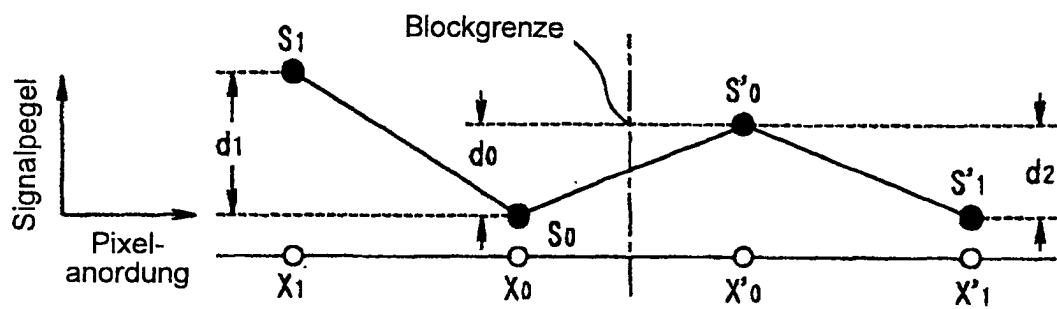


FIG. 10

