



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 32 689 T2** 2007.11.08

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 081 649 B1**

(21) Deutsches Aktenzeichen: **600 32 689.6**

(96) Europäisches Aktenzeichen: **00 307 555.3**

(96) Europäischer Anmeldetag: **01.09.2000**

(97) Erstveröffentlichung durch das EPA: **07.03.2001**

(97) Veröffentlichungstag

der Patenterteilung beim EPA: **03.01.2007**

(47) Veröffentlichungstag im Patentblatt: **08.11.2007**

(51) Int Cl.⁸: **G06T 9/00** (2006.01)
H03M 7/30 (2006.01)

(30) Unionspriorität:
387870 01.09.1999 US

(73) Patentinhaber:
ATI International Srl, Christ Church, BB

(74) Vertreter:
Meissner & Meissner, 14199 Berlin

(84) Benannte Vertragsstaaten:
DE, FR, GB

(72) Erfinder:
DeRoo, John, Hopkinton, Massachusetts 01748, US; Favela, Brian, Schrewsbury, Massachusetts 01545, US; Morein, Steven, Cambridge, Massachusetts 02139, US; Wright, Michael T., Marlborough, Massachusetts 01752, US

(54) Bezeichnung: **Verfahren und Vorrichtung zur Bildkompression**

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

[0001] Die Erfindung betrifft ganz allgemein die dreidimensionale (3D) Bildbearbeitung und insbesondere ein Verfahren und eine Vorrichtung zum Komprimieren von Parameterwerten für Pixel in einem 3D-Grafik-Einzelbild.

[0002] Computer werden für viele Anwendungen verwendet. Da sich Computersysteme weiterentwickeln, sind die Anforderungen an die grafische Anzeige mehr und mehr gefragt. Dies trifft insbesondere zu für das Gebiet der dreidimensionalen (3D) Bildbearbeitung. Um 3D-Grafik-bilder zu bearbeiten, muss die Position der Grafikgrundelemente in Bezug zu der Anzeige in allen drei Dimensionen verstanden werden. Dies umfasst die Dimension der Tiefe, häufig als die Z-Dimension bezeichnet. Die Z-Dimension beschreibt die Positionierung eines 3D-Grafik-Grundelements in Bezug auf andere 3D-Grafik-Grundelemente in dem Einzelbild hinsichtlich der Tiefe bzw. des Abstandes vom Betrachter. Dadurch können Objekt in einer überlappenden Weise vor- oder hintereinander gezogen werden.

[0003] Computerdisplays oder andere hochauflösende Anzeigeeinrichtungen, wie HDTV-Fernseher (HDTVs), Projektoren, Drucker und dergleichen, präsentieren dem Betrachter ein Bild als eine Anordnung einzelner Bildelemente oder Pixel. Den einzelnen Pixeln wird eine spezifische Farbe gegeben, die der Farbe der Abbildung an dem Ort des speziellen Pixels entspricht. Die Pixel liegen in einem engen Abstand zueinander, und das optische Wahrnehmungssystem des Betrachters führt eine Filterung der einzelnen Pixelfarben durch, um ein zusammengesetztes Bild zu formen. Wenn das Partitionieren der Abbildung in die einzelnen Pixelelemente richtig durchgeführt wird und die Pixel eng genug beieinander liegen, nimmt der Betrachter die angezeigte Anordnung von Pixel als ein virtuelles zusammengesetztes Bild wahr.

[0004] Um ein gleichmäßiges und kontinuierliches Bild auf dem Display darzustellen, muss die Bearbeitungseinheit beim Bearbeiten der 3D-Grafikbilder eine hohe Pixel-Bearbeitungsrate beibehalten. Um hohe Pixel-Bearbeitungsraten zu erreichen, müssen die in einem Displayspeicher gespeicherten Pixeldaten in einer effizienten Weise zurückgeladen, bearbeitet und dann wieder im Speicher zurückgespeichert werden. Um sehr hohe Pixelbearbeitungsraten zu erhalten, wird eine große Menge Speicherplatz benötigt. Der Grund dafür ist, dass immer dann, wenn neue Pixelfragmente zurückgeladen werden, wenigstens die bereits in einem Display-Teilbild gespeicherte, bestehende Z-Komponente an Pixeln zurückgeladen und mit den Fragmenten verglichen werden muss, um ein aktualisiertes Bild zu erzeugen. Der sich ergebende Informationssatz für jeden geänder-

ten Pixel muss dann in den Speicher zurückgespeichert werden.

[0005] Mit zunehmender Auflösung des Displays oder Pixelzahl in einer bestimmten Fläche, nehmen die Anforderungen an den Speicherplatz zu, um das angezeigte Bild zu erhalten. Höhere Anforderungen an den Speicherplatz übertragen sich in erhöhte Kosten. Der Grund dafür kann sein, die Anforderung nach schnelleren, teureren Speichern, deren Geschwindigkeit den benötigten Platz bereitstellt oder durch mehrere Speicher, die parallel arbeiten kann. Parallele Speicherstrukturen sind unerwünscht, da sie das System zu komplex machen und die Kosten für die Herstellung von 3D-Bildbearbeitungssystemen steigern.

[0006] Deshalb besteht ein Bedürfnis nach einem Verfahren und einer Vorrichtung zum Verringern der Speicherplatzanforderungen in einem 3D-Grafiksystem.

[0007] Die US 5,864,342 offenbart ein Beispiel eines Verfahrens zum Erzeugen grafischer Objekte unter Nutzung eines reduzierten Speicherplatzes. In einem Beispiel werden die geometrischen Grundelemente von Modellen in einer Szene in Portionen oder Blöcke von 32×32 Pixel unterteilt. Diese Blöcke werden noch einmal unterteilt in kleine Unterblöcke von 8×8 Pixel. Die Unterblöcke werden dann unter Verwendung von direkten Cosinus-Transformationstechniken einzeln komprimiert.

[0008] Gemäß einem Aspekt der vorliegenden Erfindung umfasst ein Verfahren zum Komprimieren von Parameterwerten für Pixel eines Pixelblocks: Umwandeln von Parameterwerten für den Pixelblock in einen spaltenweise Differentialdarstellung als Steigung, die die Parameterwerte als eine Mehrzahl von Bezugspunkten, eine Mehrzahl von Steigungen und eine Mehrzahl von abgeleiteten Steigungen darstellt; Umwandeln der spaltenweisen Differentialdarstellung als Steigung in eine planare Darstellung der Steigung, die die planaren Werte unter Verwendung wenigstens eines Bezugswertes, zweier Bezugssteigungen und einer Mehrzahl von abgeleiteten Steigungen erzeugt; und Erzeugen einer Ausgabeformat-Darstellung der Parameterwerte für den Pixelblock aus der planaren Differentialdarstellung als Steigung.

[0009] Gemäß einem zweiten Aspekt der vorliegenden Erfindung umfasst ein Verfahren für eine zweiseitige Kompression von Parameterwerten für Pixel eines Pixelblocks ein Durchführen einer ersten und zweiten Randkompression, wobei jede Randkompression das jeweilige Erzeugen umfasst: einer spaltenweisen Differentialdarstellung als Steigung mit den ersten spaltenweisen Differentialdarstellungen als Steigung und

den planaren Differentialdarstellungen der Steigung gemäß dem ersten Aspekt der Erfindung, und Erzeugen einer Ausgabeformat-Darstellung für den Pixelblock, wobei das Erzeugen der Ausgabeformat-Darstellung umfasst:

wenn ein vorbestimmtes Unterbrechungsmuster in der ersten oder zweiten Mehrzahl von Spalten-Steigungsableitungen erfasst wird, eine Doppelrand-Ausgabeformatdarstellung der Parameterwerte für die ersten und zweiten spaltenweise abgeleiteten Steigungsdarstellungen erzeugt wird, wobei die Doppelrand-Ausgabeformatdarstellung die ersten und zweiten Bezugspixelwerte, die ersten und zweiten Bezugszeilensteigungen, die ersten und zweiten Bezugsspaltensteigungen, Teile jeder ersten und zweiten Mehrzahl von Zeilen-Steigungsableitungen, Teile der ersten und zweiten Mehrzahl von Spalten-Steigungsableitungen und eine Unterbrechungs-Anzeige in eine vorbestimmte doppelseitige Ordnung ordnet, wobei die Teile der ersten und zweiten Mehrzahl von Spalten- und Zeilen-Steigungsableitungen, die in der Doppelseiten-Ausgabeformat-Darstellung enthalten sind, auf der Grundlage einer Stelle des vorbestimmten Unterbrechungsmusters in dem Pixelblock bestimmt wird, wobei die Unterbrechungsanzeige die Teile der ersten und zweiten Mehrzahl von Spalten- und Zeilen-Steigungsableitungen beschreibt; und wenn das vorbestimmte Unterbrechungsmuster nicht erfasst wird, eine Einzelseite-Ausgabeformat-Darstellung der Parameterwerte für den Pixelblock erzeugt wird, wobei die Einzelseite-Ausgabeformat-Darstellung den ersten Bezugspixelwert, die erste Bezugsspaltensteigung, die erste Bezugszeilensteigung, die erste Mehrzahl von Zeilen-Steigungsableitungen und die erste Mehrzahl von Spalten-Steigungsableitungen in einer vorbestimmten Einzelseitenordnung ordnet.

[0010] Gemäß einem dritten Aspekt der vorliegenden Erfindung umfasst ein Pixelparameter-Kompressionsprozessor zum Komprimieren von Parameterwerten für Pixel eines Pixelblocks:

ein Bearbeitungsmodul; und einen Speicher, der betriebsmäßig mit dem Bearbeitungsmodul gekoppelt ist, wobei der Speicher Betriebsbefehle speichert, die bei Ausführung durch das Bearbeitungsmodul bewirken, dass das Bearbeitungsmodul die folgenden Funktionen ausführt.

[0011] Umwandeln von Parameterwerten für den Pixelblock in eine spaltenweise abgeleitete Steigungsdarstellung, die die Parameterwerte als eine Mehrzahl von Bezugspunkten, eine Mehrzahl von Steigungen und eine Mehrzahl von Steigungsableitungen darstellt;

Umwandeln der spaltenweisen abgeleiteten Steigungsdarstellung in eine planare abgeleitete Steigungsdarstellung, die die planaren Werte unter Verwendung wenigstens eines Bezugspixelwertes, zwei Bezugssteigungen und einer Mehrzahl von Stei-

gungsableitungen darstellt; und Erzeugen einer Ausgabeformat-Darstellung der Parameterwerte für den Pixelblock aus der planaren abgeleiteten Steigungsdarstellung.

[0012] Gemäß einem weiteren Aspekt der vorliegenden Erfindung wird der Pixelparameter-Kompressionsprozessor des dritten Aspekts für ein Doppelrand-Kompressionsverfahren verwendet. In diesem Falle

speichert der Speicher Betriebsbefehle, die bei Ausführung durch das Bearbeitungsmodul das Bearbeitungsmodul dazu veranlassen, die Funktionen durchzuführen:

Durchführen einer ersten Randkompression, wobei die erste Randkompression beinhaltet:

Umwandeln von Parameterwerten für den Pixelblock in eine erste spaltenweise abgeleitete Steigungsdarstellung, wobei die erste spaltenweise abgeleitete Steigungsdarstellung eine erste Mehrzahl von Pixelpunktwerten entlang eines ersten Randes des Pixelblocks, eine erste Mehrzahl von Spaltensteigungen und eine erste Mehrzahl von Spalten-Steigungsableitungen umfasst, wobei jede Spaltensteigungsableitung der ersten Mehrzahl von Spaltensteigungsableitungen eine Spaltensteigungsabweichung an einem Pixelpunkt darstellt, der nicht durch die erste Mehrzahl von Pixelpunktwerten und die erste Mehrzahl von Spaltensteigungen dargestellt wird; und

Umwandeln der ersten spaltenweisen abgeleiteten Steigungsdarstellung in eine erste planare abgeleitete Steigungsdarstellung, wobei die erste planare Steigungsdarstellung die erste Mehrzahl von Pixelpunktwerten und die

erste Mehrzahl von Spaltensteigungen auf einen ersten Bezugspixelwert, eine erste Bezugsspaltensteigung, eine erste Bezugszeilensteigung und eine erste Mehrzahl von Zeilen-Steigungsableitungen reduziert;

Durchführen einer zweiten Randkompression, wobei die zweite Randkompression beinhaltet:

Umwandeln wenigstens eines Teils der Parameterwerte für den Pixelblock in eine zweite spaltenweise abgeleitete Spaltendarstellung, wobei die zweite spaltenweise abgeleitete Spaltendarstellung die zweite Mehrzahl von Pixelpunktwerten und die zweite Mehrzahl von Spaltensteigungen auf einen zweiten Bezugspixelwert, eine zweite Bezugsspaltensteigung, eine zweite Bezugszeilensteigung und eine zweite Mehrzahl von Zeilen-Steigungsableitungen reduziert; und

Erzeugen einer Ausgabeformat-Darstellung für den Pixelblock, wobei das Erzeugen der Ausgabeformat-Darstellung umfasst:

wenn ein vorbestimmtes Unterbrechungsmuster in einer der ersten und zweiten Mehrzahl von Spalten-Steigungsableitungen erfasst wird, eine Doppelrand-Ausgabeformatdarstellung für die Parameterwerte für die ersten und zweiten spaltenweise abgeleitete Spaltensteigungsdarstellungen erzeugt wird,

wobei die Doppelrand-Ausgabeformatdarstellung die ersten und zweiten Bezugspixelwerte, die ersten und zweiten Bezugszeilensteigungen, die ersten und zweiten Bezugsspaltensteigungen, Teile jeweils der ersten und zweiten Mehrzahl von Zeilen-Steigungsableitungen, Teile der ersten und zweiten Mehrzahl von Spalten-Steigungsableitungen in ein vorbestimmtes Muster und eine Unterbrechungspunktanzeige ordnet, wobei die enthaltenen Teile, welche die Teile der ersten und zweiten Mehrzahl von Spalten- und Zeilen-Steigungsableitungen beinhalten, die in der Doppelrand-Ausgabeformatdarstellung enthalten sind, basierend auf der Stelle des vorbestimmten Unterbrechungspunktmusters in dem Pixelblock bestimmt werden, wobei die Unterbrechungspunktanzeige enthaltene Teile beschreibt; und wenn das vorbestimmte Knickpunktmuster nicht erfasst wird, eine Einzelrand-Ausgabeformatdarstellung der Parameterwerte für den Pixelblock erzeugt wird, wobei die Einzelrand-Ausgabeformatdarstellung den ersten Bezugspixelwert, die erste Bezugsspaltensteigung, die erste Bezugszeilenspalte, die erste Mehrzahl von Zeilen-Steigungsableitungen und die erste Mehrzahl von Spalten-Steigungsableitungen in eine vorbestimmte Ordnung ordnet.

[0013] Ein Beispiel in Übereinstimmung mit der vorliegenden Erfindung wird mit Bezug auf die beigefügten Zeichnungen beschrieben, in welchen:

[0014] [Fig. 1](#) zeigt ein Blockdiagramm eines Einzelbildes, das in eine Mehrzahl von Pixelblöcke unterteilt ist;

[0015] [Fig. 2](#) zeigt ein Blockdiagramm eines Grundelements und verschiedene Pixelblöcke, die eine dem Grundelement entsprechende Pixelinformation enthalten;

[0016] [Fig. 3](#) zeigt ein Blockdiagramm eines Pixelparameter-Kompressionsprozessors in Übereinstimmung mit der vorliegenden Erfindung;

[0017] [Fig. 5](#) zeigt einen Pixelblock, der eine Mehrzahl von Pixel mit einer Mehrzahl von Parameterwerten enthält;

[0018] [Fig. 6](#) zeigt ein Blockdiagramm einer spaltenweisen abgeleiteten Steigungsdarstellung der aus den Pixelblockparametern in [Fig. 5](#);

[0019] [Fig. 7](#) zeigt eine planare abgeleitete Steigungsdarstellung aus den Pixelblockparametern in [Fig. 5](#);

[0020] [Fig. 8](#) zeigt ein Blockdiagramm einer Ausgabeformat-Darstellung der Parameterwerte für einen Pixelblock;

[0021] [Fig. 9](#) zeigt ein Blockdiagramm einer ande-

ren Ausgabeformat-Darstellung der Parameterwerte eines Pixelblocks;

[0022] [Fig. 10](#) zeigt ein Blockdiagramm eines Pixelblocks und der entsprechenden Pixelparameterwerte;

[0023] [Fig. 11](#) zeigt eine spaltenweise abgeleitete Steigungsdarstellung aus den Parameterwerten des Pixelblocks aus [Fig. 10](#);

[0024] [Fig. 12](#) zeigt ein Blockdiagramm einer planaren abgeleiteten Steigungsdarstellung aus den Parameterwerten des Pixelblocks in [Fig. 10](#);

[0025] [Fig. 13](#) zeigt eine Schnittansicht eines Pixelblocks, der zwei überschneidende Grundelemente aufweist;

[0026] [Fig. 14](#) zeigt ein Flussdiagramm eines anderen Verfahrens zum Komprimieren von Parameterwerten eines Pixelblocks;

[0027] [Fig. 15](#) zeigt ein Blockdiagramm einer ersten spaltenweisen abgeleiteten Steigungsdarstellung eines Pixelblocks, die mit dem in [Fig. 14](#) dargestellten Verfahren korrespondiert;

[0028] [Fig. 16](#) zeigt eine zweite spaltenweise abgeleitete Steigungsdarstellung, die auch mit dem in [Fig. 14](#) dargestellten Verfahren korrespondiert; und

[0029] [Fig. 17](#) zeigt einen Teil einer doppelseitigen planaren abgeleiteten Steigungsdarstellung eines Pixelblocks in Übereinstimmung mit dem in [Fig. 14](#) dargestellten Verfahren.

[0030] Ganz allgemein liefert die vorliegende Erfindung ein Verfahren und eine Vorrichtung zum Komprimieren von Parameterwerten für Pixel in einem Frame, wobei die Parameterwerte die Werte des Z-Parameters für jeden Pixel in dem Frame enthalten können. Dies wird durch ein Gruppieren der Pixel in dem Einzelbild in eine Mehrzahl von Pixelblöcke erreicht, wobei jeder Pixelblock eine Mehrzahl von Pixel enthält. Für wenigstens einen der Pixelblöcke werden die Parameterwerte für den Pixelblock in eine spaltenweise abgeleitete Steigungsdarstellung umgewandelt, welche die Parameterwerte als eine Mehrzahl von Bezugspunkte, eine Mehrzahl von Steigungen und eine Mehrzahl von Steigungsableitungen darstellt. Die spaltenweise abgeleitete Steigungsdarstellung wird dann in eine planare abgeleitete Steigungsdarstellung umgeformt, welche die Darstellung der Mehrzahl von Bezugspunkten und der Mehrzahl von Steigungen auf einen einzigen Bezugspixelwert, zwei Bezugsteigungen und eine Mehrzahl von Steigungsableitungen reduziert. Eine Ausgabeformat-Darstellung der planaren abgeleiteten Steigungsdarstellung wird dann erzeugt, wobei

ein Kodieren der Steigungsableitungen ermöglicht, die Parameterwerte für den Pixelblock in den meisten stark zu komprimieren. Diese komprimierte Formatdarstellung der Parameterwerte kann dann in einer Weise in einem Speicher gespeichert und von diesem zurückgeladen werden, welche die Speicherplatzanforderungen zum Durchführen einer dreidimensionalen 3D-Grafikverarbeitung stark verringert.

[0031] Die Erfindung kann mit Bezug auf die [Fig. 1](#) bis [Fig. 17](#) besser verstanden werden. [Fig. 1](#) zeigt ein Einzelbild **10**, das in eine Mehrzahl von Pixelblöcke **20** zergliedert ist, wobei die Mehrzahl von Pixelblöcken **20** einen Pixelblock **22** umfasst. Die Pixel in dem Einzelbild **10** sind in Pixelblöcke gruppiert, um eine Parameterkompression zu erleichtern. In dem Einzelbild **10** sind Anzeigeobjekte durch eine Anzahl von 3D-Grafik-Grundelemente darstellt. Vorzugsweise sind diese 3D-Grafik-Grundelemente polygonale Grundelemente und können Dreieck-Grundelemente sein, welche gewöhnlich in 3D-Grafiksystemen verwendet werden. Die Größe der Pixelblöcke **20** wird basierend auf der typischen Größe eines für die Erzeugung eines Objekts in dem Einzelbild **10** verwendeten Grundelements gewählt. Vorzugsweise wird die Größe der Pixelblöcke **20** derart gewählt, dass der Hauptteil der Pixelblöcke vollständig in ein spezielles Grafik-Grundelement in dem Einzelbild **10** fällt. In einer Ausführungsform ist jeder der Pixelblöcke **20** ein 8 Pixel mal 8 Pixel Block. In einer solchen Ausführungsform würde jeder Pixelblock **64** Pixel umfassen. Obwohl die in jedem der Blöcke enthaltene Anzahl an Pixel variieren kann, wird das 8 mal 8 Beispiel in dem Rest der Beschreibung verwendet, um die Veranschaulichung der Erfindung zu unterstützen. Es sollte angemerkt werden, dass die verwendete Blockkonfiguration nicht nach einem quadratischen Block verlangt und zum Beispiel ein 8 mal 16 rechtwinkliger Block verwendet werden könnte.

[0032] [Fig. 2](#) zeigt ein Dreieck-Grundelement **40**. Dreiverschiedene Typen von Pixelblöcken sind in Verbindung mit dem Grundelement **40** dargestellt. Der Feldblock **46** ist ein Pixelblock, der vollständig in das Grundelement **40** fällt. Der Randblock **44** ist ein Pixelblock, der entlang eines Randes des Grundelements **40** liegt und vorzugsweise auch entlang des Randes eines weiteren Grundelements liegt, derart, dass der Randblock **44** die Grenze von zwei Grundelementen entlang einer gemeinsamen Teilungslinie darstellt. Der Eckpunktblock **42** umfasst einen der Eckpunkte des Grundelements **40** und schneidet ebenso mehrere Grundelemente. Wie dem Fachmann klar ist, können mehrere Eckpunkte oder alle der Eckpunkte des Grundelements in dem gleichen Block enthalten sein, wenn das Grundelement im Vergleich zu der verwendeten Blockgröße klein genug ist.

[0033] Die hier beschriebene Erfindung nutzt die

Tatsache, dass Feldblöcke, wie der Block **46**, den Hauptteil der Pixelblöcke in einem speziellen Einzelbild umfassen. Alle Pixel in einem Feldblock haben Z-Koordinatenwerte, die in der gleichen Ebene liegen, die mit der Ebene korrespondiert, entlang welcher die Grundelemente liegen. Das Ausnutzen dieser planaren Zugehörigkeit zwischen den verschiedenen Z-Koordinatenwerten der Feldblöcke in dem Frame ermöglicht, dass der Hauptteil der Parameterwerte für einen Feldblock mit einer stark reduzierten Anzahl von Bits dargestellt wird, aber erforderlich macht, dass jedes Pixel einzeln dargestellt wird. Die Komprimierung kann in einer verlustfreien Weise durchgeführt werden, die ermöglicht, dass alle speziellen Z-Koordinatenwerte für jeden der Pixel in einem Pixelblock vollständig wieder gewonnen werden, wenn der komprimierte Datensatz für den Pixelblock entkomprimiert wird.

[0034] [Fig. 3](#) zeigt einen Kompressionsprozessor, der in Verbindung mit den in [Fig. 4](#) und [Fig. 14](#) dargestellten Verfahren verwendet werden kann. Es sei darauf hingewiesen, dass die Verfahren der [Fig. 4](#) und [Fig. 14](#) ohne die Verwendung eines dargestellten Kompressionsprozessors durchgeführt werden könnten. Der Pixelparameter-Kompressionsprozessor **50** aus [Fig. 3](#) umfasst ein Bearbeitungsmodul **52** und einen Speicher **54**. Das Bearbeitungsmodul kann eine einzige Bearbeitungseinheit oder eine Mehrzahl von Bearbeitungseinheiten umfassen. Eine solche Bearbeitungseinheit kann ein Mikroprozessor, ein Mikrokontroller, ein Digital-Signalprozessor, ein Zustandsautomat, eine logische Schaltung und/oder eine beliebige Einrichtung sein, die eine auf Betriebs- und/oder Programmbefehle basierende Information bearbeitet.

[0035] Der Speicher **54** kann eine einzelne Speichereinrichtung oder eine Mehrzahl von Speichereinrichtungen sein. Eine solche Speichereinrichtung kann eine Nullese-Speichereinrichtung, eine direkte Zugriffsspeichereinrichtung, eine Floppydisc, ein Festplattenspeicher und/oder eine Einrichtung, die eine digitale Information speichert. Es sei darauf hingewiesen, dass, wenn das Bearbeitungsmodul **52** ein oder mehrere seiner Funktionen mit einem Zustandsautomaten und/oder einer logischen Schaltung durchführt, der die entsprechenden Betriebsbefehle enthaltende Speicher in den Zustandsautomaten und/oder die logische Schaltung eingebettet ist. Der Speicher **54** speichert Programm- und/oder Betriebsbefehle, die, wenn sie ausgeführt werden, dem Bearbeitungsmodul **52** ermöglichen, wenigstens eines der in den [Fig. 4](#) und [Fig. 14](#) dargestellten Verfahren durchzuführen. Es sei angemerkt, dass der Prozessor **50** einige der Funktionen dieser Verfahren durch in dem Speicher **54** gespeicherte Software durchführen kann, während andere Teile unter Verwendung von Hardware oder einem in dem Prozessor **50** enthaltenen Schaltkreis durchgeführt werden können.

So können in einigen Ausführungsformen eine Mischung aus Hardware und Software dazu verwendet werden, die Verfahren der [Fig. 4](#) und [Fig. 14](#) durchzuführen.

[0036] [Fig. 4](#) zeigt ein Flussdiagramm eines Verfahrens zum Komprimieren der Parameterwerte für Pixel in einem Einzelbild, derart, dass die Speicherplatzanforderungen verringert werden können. Vorzugsweise sind die komprimierten Parameterwerte die Z-Koordinaten-Parameterwerte für die Pixel in dem Einzelbild. Es sollte jedoch für den Fachmann klar sein, dass andere Parameterwerte für die Pixel in einer ähnlichen Weise komprimiert werden können. Damit andere Parameterwerte komprimiert werden können, sollte die Beziehung zwischen den Parameterwerten in einem Grundelement eine relativ planare Beziehung sein, wie diejenige, die für die Z-Werte für verschiedene Pixel in einem Grundelement existiert.

[0037] Das Verfahren aus [Fig. 4](#) beginnt am Schritt **62**, an welchem Pixel für das Einzelbild in eine Mehrzahl von Pixelblöcken gruppiert sind. Vorzugsweise wird diese Pixelgruppierung derart durchgeführt, dass der Hauptteil der Pixelblöcke in dem Einzelbild Feldblöcke sind, derart, dass sie in einem einzelnen Grundelement enthalten sind. Nachdem die Gruppierung am Schritt **62** durchgeführt worden ist, werden die Schritte **64** bis **68** für wenigstens einen Pixelblock durchgeführt und vorzugsweise für alle Pixelblöcke in dem Einzelbild. Die Schritte **64** bis **78** entsprechen der Kompressionspeicherung und der Dekomprimierung eines einzelnen Pixelblocks und können für mehrere Pixelblöcke in dem Frame wiederholt werden.

[0038] [Fig. 5](#) zeigt einen speziellen Pixelblock **22**, der eine Mehrzahl von Pixel **80** enthält, von denen jeder einen Pixelparameterwert **81** hat. So zeigt jeder der „P“ Zeichen einen Pixelparameterwert an der speziellen Pixelstelle. In einer Ausführungsform, in welcher die Parameterwerte Z-Koordinaten-Parameterwerte für die Pixel sind, können die Z-Koordinaten-Parameterwerte durch zwei 16 Bit Wörter dargestellt werden, derart, dass 32 Bits benötigt werden, um jeden der Parameterwerte **81** in dem Block **22** darzustellen. In einer solchen Ausführungsform würden 2048 Bits benötigt, um die Z-Werte für alle Pixel in dem Block **22** zu speichern.

[0039] Die Komprimierung beginnt am Schritt **64**, an welchem die Parameterwerte für die in dem Pixelblock enthaltenen Pixel in eine spaltenweise abgeleitete Steigungsdarstellung übertragen werden. [Fig. 6](#) stellt die spaltenweise abgeleitete Steigungsdarstellung **88** der Parameterwerte für den Pixelblock **22** dar. Die spaltenweise abgeleitete Steigungsdarstellung **88** umfasst eine Mehrzahl von Pixelpunktswerten **82**, eine Mehrzahl von Spaltensteigungen **84** und eine Mehrzahl von Spalten-Steigungsableitungen **86**.

Die Pixelpunktswerte **82** entsprechen den Parameterwerten für die Pixel, die vorzugsweise entlang eines Randes des Pixelblocks angeordnet sind. Es sei darauf hingewiesen, dass die Ausrichtung des Pixelblocks derart verändert werden kann, dass die Zeile von Pixelpunktswerten, welche als die obere Zeile des Pixelblocks in [Fig. 6](#) dargestellt ist, entlang der unteren Zeile oder entlang der äußerst linken oder äußerst rechten Spalten positioniert sein kann.

[0040] Die Spaltensteigungen **84** repräsentieren die Steigung des Parameterwertes entlang jeder der Spalten. Innerhalb jeder Spalte kann die Spaltensteigung durch ein einfaches Subtrahieren des Parameterwertes in der zweiten Zeile von dem Parameterwert in der ersten Zeile berechnet werden. Fall somit die erste Zeile einen Parameterwert P_0 hat und der Parameterwert in der zweiten Zeile einen Wert P_1 hat, ist die Spaltensteigung in der Spalte gleich P_1 minus P_0 .

[0041] Jede der Spalten-Steigungsableitungen **86** wird so berechnet, dass diese die Abweichung von der gegenwärtigen Steigung des Parameters am Punkt entlang der Spalte angibt. Auf diese Weise wird die abgeleitete Steigung (dS_2) die Differenz zwischen der Steigung des Parameterwertes zwischen den Pixelpunkten P_0 und P_1 und die Steigung zwischen den Pixelpunkten P_1 und P_2 sein. Die Gleichung zum Berechnen des Wertes dS_2 ist:

$$dS_2 = (P_2 - P_1) - (P_1 - P_0)$$

[0042] Ebenso können die abgeleiteten Steigungen für die verbleibenden Pixelstellen in dem Pixelblock unter Verwendung der Formel berechnet werden:

$$dS_{(N)} = (P_{(N)} - P_{(N-1)}) - (P_{(N-1)} - P_{(N-2)}).$$

[0043] Es sollte angemerkt werden, dass diese Formel entlang jeder Spalte in dem Block verwendet wird. Die Werte für jede Spalte in der spaltenweisen abgeleiteten Steigungsdarstellung **88** werden unabhängig von den Werten in anderen Spalten berechnet. Es sollte angemerkt werden, dass anstatt des Berechnens einer spaltenweisen abgeleiteten Steigungsdarstellung das gleiche Prinzip auf eine zeilenweise Basis angewendet werden kann anstatt auf eine spaltenweise Basis, um ähnliche Ergebnisse mit einer unterschiedlichen Ausrichtung zu erhalten.

[0044] Weiter mit dem Flussdiagramm in [Fig. 4](#) wird am Schritt **66** die spaltenweise abgeleitete Steigungsdarstellung in eine planare abgeleitete Steigungsdarstellung übertragen. Die planare abgeleitete Steigungsdarstellung **99**, die sich aus der Übertragung der spaltenweisen abgeleiteten Steigungsdarstellung **88** ergibt, ist in [Fig. 7](#) dargestellt. Zu sehen ist, dass die planare abgeleitete Steigungsdarstellung **99** die Mehrzahl von Pixelpunktswerten **82** und

die Mehrzahl von Spaltensteigungen **84** der spaltenweisen abgeleiteten Steigungsdarstellung **88** auf einen Bezugspixelwert **92**, eine Bezugszeilensteigung **94**, eine Bezugsspaltensteigung **96** und eine Mehrzahl von Zeilen-Steigungsableitungen **98** reduziert.

[0045] Die planare abgeleitete Steigungsdarstellung zieht einen weiteren Vorteil aus der Korrelation zwischen den Parameterwerten in dem Pixelblock und vereinfacht weiter die Darstellung der Pixelpunktweite **82** und der Spaltensteigungen **84** der spaltenweisen abgeleiteten Steigungsdarstellung **88**. Im Wesentlichen wird das gleiche Prinzip, das auf die Spalten angewendet wird, um die spaltenweise abgeleitete Spaltendarstellung **88** zu erhalten, auf die ersten zwei Zeilen spaltenweisen abgeleiteten Steigungsdarstellung **88** angewendet, um die planare abgeleitete Steigungsdarstellung **99** zu erzeugen. Es sei darauf hingewiesen, dass die Spalten-Steigungsableitungen **86**, die zum Einsetzen in die spaltenweise abgeleitete Steigungsdarstellung **88** berechnet werden, in der planaren abgeleiteten Steigungsdarstellung **99** unverändert bleiben. Die Bezugszeilensteigung **94** ermöglicht, dass die Pixelpunktweite **82** in eine Reihe von Zeilen-Steigungsableitungen komprimiert werden.

[0046] Die Regelmäßigkeit der Parameterwerte in dem Pixelblock wird auch dazu verwendet, die Spaltensteigungen **84** in der zweiten Zeile auf eine einzelne Bezugsspaltensteigung **96** und die verbleibenden Zeilen-Steigungsableitungen zu reduzieren, die basierend auf die Bezugsspaltensteigung **96** erzeugt werden. Die abgeleitete Steigung, die in der zweiten Zeile und der zweiten Spalte liegt, wird durch Subtrahieren der Steigung an der Pixelstelle von der Bezugsspaltensteigung **96** berechnet. Die abgeleitete Steigung in der zweiten Zeile und der dritten Spalte wird berechnet durch Subtrahieren der Steigung an der Pixelstelle von der Steigung an der zweiten Zeile und der zweiten Spalte und so weiter.

[0047] Weiter mit dem Verfahren in [Fig. 4](#) am Schritt **68** wird eine Ausgabeformat-Darstellung aus der planaren abgeleiteten Steigungsdarstellung erzeugt. Die Ausgabeformat-Darstellung **100**, die in einer Ausführungsform verwendet wird, ist in [Fig. 8](#) dargestellt. Unter Verwendung des Beispiels, in welchem jeder Parameterwert an jeder Pixelstelle unter Verwendung von 32 Bits dargestellt wird, kann der Wert an dem Bezugspixel **102** mit 32 Bits dargestellt werden. Die Bezugszeilensteigung **104** und die Bezugsspaltensteigung **106** benötigen 33 Bits, da diese Steigungen mit Vorzeichen versehene Steigungen sind, derart, dass ein zusätzliches Bit benötigt wird, um eine positive oder negative Steigung anzugeben.

[0048] Die Komprimierung wird beim Kodieren der abgeleiteten Steigung in der Ausgabeformat-Darstellung **100** erreicht. Wegen der hohen Korrelation zwi-

schen Pixelparameterwerten (Z-Werten in einer Ausführungsform) in den Pixelblöcken, werden die abgeleiteten Steigungen typischerweise sehr kleine Werte haben. In den meisten Fällen werden die abgeleiteten Steigungen einen ganzzahligen Wert haben, der von -1 bis 1 reicht. Die abgeleiteten Steigungen ergeben sich aus der Diskretisierung der Z-Werte. Ein Kodierschema kann somit verwendet werden, bei welchem ein -1 durch einen Zwei-Bitwert dargestellt wird, eine 0 durch einen Zwei-Bitwert dargestellt wird und eine 1 als ein dritter Zwei-Bitwert angegeben wird. Der letzte verfügbare Zwei-Bitwert in einem Zwei-Bitkodierungssatz kann dann dazu verwendet werden, ein Escape zu kodieren. Ein Escape gibt an dass der Wert der abgeleiteten Steigung an einer speziellen Stelle außerhalb des Bereichs der Kodierungsbits liegt. Auf diese Weise wird jede abgeleitete Steigung, die größer als 1 oder kleiner als -1 ist, mit einem Escape-Wert kodiert. In einem einfachen Beispiel könnte eine 0 kodiert werden als „00“, eine -1 könnte kodiert werden als X10“, eine 2 könnte kodiert werden als „01“ und schließlich könnte ein Escape-Wert kodiert werden als „11“.

[0049] Es sollte angemerkt werden, dass die Anzahl von Kodierungsbits, die verwendet werden, um die Spalten- und Zeilen-Steigungsableitungen in der Ausgabeformat-Darstellung zu kodieren, mehr oder weniger als zwei Bits sein könnten. In einem Ein-Bit-Beispiel könnte ein Wert verwendet werden, um eine 0 Ableitungssteigung zu kodieren, wohingegen der andere Wert ein Escape angibt. Ebenso ermöglicht die Verwendung von drei Kodierbits einen größeren Kodierbereich, so dass möglicherweise weniger Escapes notwendig werden. In der Annahme, dass zwei Bits verwendet werden, um jede abgeleitete Steigung zu kodieren, würde die kodierte abgeleitete Steigungsanordnung **108** in der Ausgabeformat-Darstellung **100** 122 Bits benötigen. Es sei darauf hingewiesen, dass das Ordnen der abgeleiteten Steigungen in der kodierte abgeleiteten Steigungsanordnung vorzugsweise derart vorbestimmt ist, dass ein Umkehren des Kompressionsprozesses unkompliziert ist.

[0050] Alle benötigten Escape-Werte, um die Werte der abgeleiteten Steigungen in dem Pixelblock zu kodieren, müssen auch in der Ausgabeformat-Darstellung **100** enthalten sein. Weil jeder Escape-Wert aus einer Berechnung resultiert, die vier 32-Bitmengen enthält, können 34 Bits benötigt werden, um einen Escape-Wert darzustellen. Auf diese Weise werden die Escape-Werte **110**, die in der Ausgabeformat-Darstellung **100** enthalten sind, 34 Bits mal der Anzahl von Escapes benötigen, die in dem Pixelblock erforderlich sind, mit welchem die Ausgabeformat-Darstellung **100** korrespondiert. Das Ordnen der Escape-Werte **110** in der Ausgabeformat-Darstellung **100** sollte in gewisser Weise mit dem Ordnen der abgeleiteten Steigungen in der kodierte abgeleiteten

Steigungsanordnung **108** korrespondieren, derart, dass eine Dekomprimierung der Ausgabeformat-Darstellung **100** vereinfacht wird. Es sollte angemerkt werden, dass das Ordnen der verschiedenen Teile der Ausgabeformat-Darstellung **100** aufgrund einer spezifischen Implementierung variieren kann.

[0051] Im günstigsten Fall, in welchem keine Escape-Werte für einen speziellen Pixelblock benötigt werden, kann die Ausgabeformat-Darstellung **100** die 2048 Bits, die in dem Beispiel zum Speichern der unkomprimierten Parameterwerte für alle Pixel in dem Pixelblock benötigt werden, auf die 220 Bits reduzieren lassen, die benötigt werden, um die komprimierte Version zu speichern. Das Durchführen einer solchen Kompression der Parameterwerte kann ein System ermöglichen, um diese Werte zu einer Speicherstruktur zu übertragen, und zwar mit weit weniger Platz, als in einem unkomprimierten Format erforderlich wäre. Zudem kann die Komprimierung auch den Speicherplatz reduzieren, der benötigt wird, um die Parameterwerte zu speichern. Weil die Komprimierung verlustfrei ist, sind die Parameterwerte vollständig aus dem komprimierten Format wiedergewinnbar.

[0052] [Fig. 9](#) zeigt eine Ausgabeformat-Darstellung **120**, die in einer anderen Ausführungsform der Erfindung verwendet werden kann, um die Anzahl von Bits weiter zu reduzieren, die benötigt wird, um die Parameterwerte für ein Pixelblock zu speichern. Wie im Falle der Ausgabeformat-Darstellung **100**, speichert die Ausgabeformat-Darstellung **120** den Bezugspixel **102** und die kodierte abgeleitete Steigungsanordnung **108**. Eine Differenz zwischen der Ausgabeformat-Darstellung **120** und der Ausgabeformat-Darstellung **100** besteht darin, dass die Escape-Werte **130** unter Verwendung von weniger als 34 Bits gespeichert werden können. Vorzugsweise bestimmt die maximale Anzahl von Bits, die benötigt wird, um den größten oder den meisten Bit verbrauchenden Escape-Wert zu speichern, die Anzahl von Bits, die verwendet werden, um jeden der Escape-Werte **130** zu speichern. So M-Bits verwendet werden, um jeden der Escape-Werte zu speichern, wobei M-Bits die minimale Anzahl von Bits ist, die benötigt wird, um jeden der Escape-Werte in einer verlustfreien Weise vollständig zu kodieren. Damit eine Dekomprimierung der Ausgabeformat-Darstellung **120** auftreten kann, muss die Anzahl von Bits, die verwendet wird, um jeden der Escape-Werte zu speichern, in der Ausgabeformat-Darstellung **120** angegeben werden. So werden die Konfigurationsbits **128**, die die Anzahl von Bits-pro-Escape angeben, in der Ausgabeformat-Darstellung **120** kodiert. Weil jeder Escape-Wert ein Minimum von zwei Bits benötigt, sind fünf Bits richtig, um einen Bereich zwischen 2 und 24 Bits zu kodieren.

[0053] Ein anderer Unterschied zwischen der Aus-

gabeformat-Darstellung **120** und der Ausgabeformat-Darstellung **100** ist, dass die Bezugs-Reihensteigung **124** und die Bezugs-Spaltensteigung **126** mit weniger als 33 Bits dargestellt werden kann. In vielen Fällen werden die Bezugs-Reihensteigung und die Bezugs-Spaltensteigung relativ klein sein. So könnten N-Bits verwendet werden, um sowohl die Bezugs-Reihensteigung **124** als auch die Bezugs-Spaltensteigung **126** darzustellen, wobei N die minimale Zahl an Bits ist, die benötigt wird, um jeden der Bezugs-Reihensteigung und der Bezugs-Spaltensteigung in einer verlustfreien Weise zu kodieren. Noch einmal, die Konfigurationsbits **122**, die die Anzahl von Bits-pro-Steigung angeben, müssen in der Ausgabeformat-Darstellung enthalten sein, um eine Dekomprimierung zu erlauben. Wie im Fall bei den Escape-Werten, sind fünf Bits richtig, um die Anzahl von Bits, N, zu kodieren, die verwendet werden, um die Bezugs-Steigung zu kodieren.

[0054] Es ist sollte angemerkt werden, dass die Optimierungen hinsichtlich einer Reduktion der Anzahl von verwendeten Bits, um Escape-Werte oder Bezugs-Steigungen zu speichern, in verschiedenen Ausführungen der Erfindung enthalten sein können oder nicht. So kann in einer Ausführungsform nur die Optimierung für die Escape-Werte verwendet werden, in einer anderen Ausführungsform kann nur die Optimierung für Zeilen- oder Spaltenbezugssteigungen verwendet werden und in einer noch anderen Ausführungsform können beide Optimierungen verwendet werden.

[0055] Zurück zu [Fig. 4](#) wird am Schritt **70** die Ausgabeformat-Darstellung im Speicher gespeichert. Weil die Anzahl von Bits, die erforderlich ist, um die Parameterwerte für alle Pixel in einen Pixelblock zu kodieren, stark reduziert worden ist, ist die benötigte Speichergröße, um die Ausgabeformat-Darstellung zu speichern, sehr viel geringer als es erforderlich wäre, um einen unkomprimierten Satz von Parameterwerten zu speichern. Zudem ist der benötigte Speicherplatz, um die Parameterwerte zu speichern, reduziert. Eine ähnliche Reduktion in der Speichergröße wird realisiert, wenn die Ausgabeformat-Darstellung von dem Speicher am **72** erneuert wird. Sobald die Ausgabeformat-Darstellung erneuert worden ist, werden die Parameterwerte für jeden der Pixel aus der Ausgabeformat-Darstellung zurück gewonnen.

[0056] Die Zurückgewinnung der Parameterwerte beginnt am Schritt **74**, an welchem die planare abgeleitete Steigungsdarstellung aus der Ausgabeformat-Darstellung rekonstruiert wird. Weil die Ordnung der Teile der Ausgabeformat-Darstellung bekannt ist, wird die Rekonstruktion der planaren abgeleiteten Steigungsdarstellung erleichtert. Am Schritt **76** wird die spaltenweise abgeleitete Steigungsdarstellung basierend auf der planaren abgeleiteten Steigungs-

darstellung rekonstruiert. Nachdem dies erreicht worden ist, können die Parameterwerte für den Pixelblock durch Berechnen der Parameterwerte basierend auf den Pixelpunktswerten, Spaltensteigungen und Spaltensteigungs-Ableitungen am Schritt **78** zurück gewonnen werden.

[0057] Die [Fig. 10](#) bis [Fig. 12](#) zeigen ein Beispiel unter Verwendung tatsächlicher Pixelparameterwerte, die das Verständnis von Teilen des Verfahrens in [Fig. 4](#) erleichtern können. [Fig. 10](#) zeigt einen Pixelblock **140**, in welchem die Zahlen an jeder Pixelstelle den Z-Wert darstellen oder irgendeinen anderen Pixelparameterwert entsprechend jedem Pixel in dem Pixelblock **140**. Der Pixelblock **140** soll ein Feldpixelblock sein, derart, dass der gesamte Pixelblock **140** in einem einzelnen Grafik-Grundelement enthalten ist. Es gibt eine starke Korrelation unter den Parameterwerten über den gesamten Pixelblock.

[0058] [Fig. 11](#) zeigt eine spaltenweise abgeleitete Steigungsdarstellung **150**, die sich aus der Umwandlung des Pixelblocks **140** ergibt. Die spaltenweise abgeleitete Spaltendarstellung **150** umfasst eine Mehrzahl von Pixelpunktswerten **146**, welche die Parameterwerte für die in der oberen Reihe liegenden Pixel sind. Wie vorher beschrieben, wird jede der Spaltensteigungen **148** durch Subtrahieren des Parameterwerts des Pixels in der zweiten Zeile von dem Parameterwert des Pixels in der ersten Zeile berechnet. Es ist zu sehen, dass die Steigung des Parameters entlang jeder Spalte des beispielhaften Pixelblocks **1** ist. Dies sollte aus der Darstellung in [Fig. 10](#) deutlich werden.

[0059] Jede der Spalten-Steigungsableitungen **144** gibt an, ob einer der Z-Werte an diesen Stellen von der Steigung entlang ihrer jeweiligen Spalte abweicht oder nicht. In dem dargestellten einfachen Fall sind alle diese Spalten-Steigungsableitungen **144** Null. Dies gibt an, dass die Auflösung, mit welcher die Steigung für jede Spalte berechnet wurde, richtig ist, und dass keine Abweichung von der Steigung in dem Pixelblock vorliegt.

[0060] Die spaltenweise abgeleitete Steigungsdarstellung **150** in [Fig. 11](#) wird dann in die planare abgeleitete Steigungsdarstellung **160** umgeformt, die in [Fig. 12](#) dargestellt ist. Wie zu sehen ist, bleiben die Spalten-Steigungsableitungen **144** unverändert. Die Zeile von Pixelpunktswerten **146** in [Fig. 11](#) wird in einen einzelnen Bezugspixelwert **162**, eine Bezugs-Zeilensteigung **166** und die obere Zeile der Zeilen-Steigungsableitungen **168** umgeformt. Die Bezugs-Zeilensteigung **166** wird durch Subtrahieren des Pixelpunktswertes in der zweiten Spalte von dem Pixelpunktswert in der ersten Spalte berechnet. Die Steigung in diesem Fall wird als gleich 5 berechnet. So wird die Steigung von 5 über den gesamten Satz an Spalten beibehalten und somit sind die Zei-

len-Steigungsableitungen, die in der ersten Zeile enthalten sind, als 0 dargestellt.

[0061] Die Zeile von Spaltensteigungen **148** in [Fig. 11](#) kann in einer ähnlichen Weise reduziert werden. Die in der zweiten Spalte enthaltene Steigung wird von der Steigung in der ersten Spalte subtrahiert, um den Wert der abgeleiteten Steigung zu bestimmen. Weil alle der Spaltensteigungen **148** als gleich 1 dargestellt sind, sind die Zeilen-Steigungsableitungen in der zweiten Zeile als 0 dargestellt.

[0062] Das Zusammenstellen der in der planaren abgeleiteten Steigungsdarstellung **160** in [Fig. 12](#) enthaltenen Werte in einer Ausgabeformat-Darstellung ist dann eine einfache Aufgabe, die die Anzahl von Bits weitestgehend reduziert, die benötigt wird, um die Z-Werte für den Pixelblock **140** in einer verlustfreien Weise zu kodieren. Ein Komprimieren der Z-Werte und beliebiger anderer ähnlicher Werte entsprechend der Pixelblöcke in einem 3D-Grafiksystem kann dem 3D-Grafiksystem ermöglichen, unter viel weniger Speichergröße betriebsfähig zu sein. Dies wiederum kann ein viel preiswerteres System zulassen, das noch in der Lage ist, hohe Pixel-Bearbeitungsraten beizubehalten.

[0063] Das in den [Fig. 10](#) bis [Fig. 12](#) dargestellte Beispiel war ein einfaches Beispiel einer Komprimierung eines Feldblocks, der vollständig in einem einzelnen Grundelement enthalten ist. Das Beispiel ist weiter vereinfacht, da es keine Steigungsabweichung (-1 und 1 Werte enthält), die typischerweise in tatsächlichen Feldblöcken vorhanden sind, wie ihnen in der Praxis begegnet wird. Ein schwierigeres und komplexeres Beispiel existiert für einen Randblock, welcher, wie vorher beschrieben, entlang der Grenzlinie von zwei Grundelementen liegt. [Fig. 13](#) zeigt einen Pixelblock-Querschnitt **171**, an welchem der dargestellte Pixelblock zwei Grundelemente hat, die in etwa in der Mitte des Pixelblocks aneinander angrenzen. Die gelieferte Ansicht zeigt die Grundelemente als Linien, an welchen ein Grundelement eine erste Steigung **174** und das andere Grundelement eine zweite Steigung **176** hat. Falls der Linie mit der ersten Steigung **174** ausgehend von der linken Seite gefolgt wird, wird die Steigung bis zu dem Schnittpunkt beibehalten, an welchem die Steigung sich verändert. Die neue Steigung wird dann beibehalten, bis der äußerste rechte Bereich des Pixelblocks erreicht ist. Auf pixelweisen Basis wird die abgeleitete Steigung Null sein, bis zum vierten Pixel von links. An diesem Punkt wird eine dritte Steigung **178** bestimmt. Diese dritte Steigung ist eine Anomalie, die der Steigung zwischen dem Pixelpunkten **182** und **180** entspricht.

[0064] Die Veränderungen in der Steigung innerhalb des Pixelblocks würde zu zwei Escape-Werten führen, die entlang jeder Zeile oder Spalte (in Abhän-

gigkeit von der Ausrichtung) der Pixelausgabedarstellung des Pixelblocks enthalten sind. Falls jedoch der Pixelblock mit Bezugspixeln an beiden Enden des Pixelblocks angenähert ist, dann werden die Steigungsdifferenzen entlang jeder der ersten und zweiten Steigungen **174** und **176** bis zum Schnittpunkt Null sein. Dies deshalb, weil die Steigung entlang jeder Linie bis zu dem Schnittpunkt konstant ist. Das Ausnutzen dieser Zusammenfassung kann die Anzahl von Bits reduzieren, die in der Ausgabeformat-Darstellung benötigt wird, im Vergleich zu der Anzahl von Bits, die mit der Einfügung von zwei Escape-Werten pro Zeile oder Spalte benötigt wird. Eine weitere übliche Situation, in welcher diese Technik sich als nützlich erweist, ergibt sich bei zwei Grundelementen, die tatsächlich nicht aneinander grenzen, die aber in verschiedenen Ebenen liegen, wobei eine Ebene vollständig vor der anderen liegt. Eine solche Situation findet sich häufig am Rand von Objekten, die in der Anzeige enthalten sind.

[0065] [Fig. 17](#) zeigt einen Bereich einer planaren abgeleiteten Doppelrand-Steigungsdarstellung **210**. Der Unterbrechungspunkt **220** gibt den Punkt an, an welchem das Bezugspixel, an welchem die abgeleitete Steigung Bezug nimmt zwischen dem in der oberen linken Ecke, P_u **212** und dem in der unteren linken Ecke P_u **222** umschaltet. Diese Steigung **214** und die abgeleiteten Steigungen **216** und **218** werden unter Verwendung des Bezugspixels **212** berechnet, während die Steigung **224** und die abgeleiteten **226** und **228** unter Verwendung des Bezugspixels **222** berechnet werden. Um die planare abgeleitete Doppelrand-Steigungsdarstellung **210** zu dekomprimieren, muss der Unterbrechungspunkt mit Bezug auf jede Spalte in der Ausgabeformat-Darstellung enthalten sein. Obwohl dies einigen zusätzlichen Mehraufwand mit sich bringt, kann dies in einigen Fällen sehr günstig für das Reduzieren der Anzahl von Escape-Werten sein, die in der Ausgabedarstellung benötigt werden. Die Reduktion der Escape-Werte ist typischerweise von genügendem Vorteil, um die Einfügung eines zusätzlichen Bezugs-Pixelwertes und zwei zusätzlicher Bezugs-Steigungen, die mit dem zusätzlichen Bezugs-Pixelwert korrespondieren, rechtfertigt.

[0066] [Fig. 14](#) zeigt ein Verfahren, das verwendet werden kann, um die Doppelrand-Komprimierungstechnik bei der Komprimierung eines speziellen Pixelblocks zu nutzen. Das Verfahren beginnt am Schritt **190**, an welchem eine erste Randkomprimierung durchgeführt wird. Die erste Randkomprimierung wird vorzugsweise so durchgeführt, wie dies mit Bezug auf die Schritte **64** und **66** in [Fig. 4](#) beschrieben wurde, derart, dass eine spaltenweise und eine planare abgeleitete Steigungsdarstellung basierend auf einem Bezugspixel erzeugt werden. Am Schritt **192** wird eine zweite Randkomprimierung durchgeführt, wobei die zweite Randkomprimierung einen Bezugspixel nutzt, der in einer Ecke orthogonal ge-

genüber dem Bezugspixel der ersten Randkomprimierung liegt. Die [Fig. 15](#) und [Fig. 16](#) zeigen spaltenweise abgeleitete Steigungsdarstellungen **202** und **204** entsprechend der ersten und zweiten Randkomprimierung. Der Bezugspixel für die zweite spaltenweise abgeleitete Steigungsdarstellung **204** ist vorzugsweise in der unteren linken Ecke. So ist die zweite Randkomprimierung ähnlich der ersten Randkomprimierung, mit dem einzigen Unterschied, dass diese einen anderen Bezugspixel in dem Pixelblock nutzt. Zusätzlich dazu kann die zweite Randkomprimierung eine partielle Randkomprimierung sein. Dies deshalb, weil dann, wenn eine Doppelrand-Ausgabedarstellung verwendet wird, die ersten zwei Zeilen basierend auf dem Bezugspixel der ersten Randkomprimierung beschrieben werden und deshalb die abgeleiteten Steigungen für diese Zeilen nicht für die Erfüllung der zweiten Randkomprimierung berechnet werden müssen.

[0067] Am Schritt **194** wird bestimmt, ob ein vorbestimmtes Unterbrechungspunktmuster innerhalb einer der ersten und zweiten Mehrzahl von Spalten-Steigungsableitungen entsprechend der ersten und zweiten Randkomprimierungen vorliegt oder nicht. Das Unterbrechungspunktmuster wird als ein Punkt innerhalb der abgeleiteten Steigungen erkannt, an welchem eine Steigungsdiskontinuität erfolgt. So gibt der Unterbrechungspunkt **220** in [Fig. 17](#) den Punkt an, welchem die Steigung entlang der Spalte seine Richtung signifikant verändert. Falls kein Unterbrechungspunktmuster am Schritt **194** erfasst wird, geht das Verfahren weiter zum Schritt **196**, an welchem eine Einzelrand-Ausgabeformatdarstellung der Parameterwerte für den Pixelblock aus den Ergebnissen der ersten Randkomprimierung erzeugt wird. Die Einzelrand-Ausgabeformatdarstellung wird den ersten Bezugspixel, die erste Bezugs-Spaltensteigung, die erste Bezugs-Zeilensteigung, die erste Mehrzahl von Zeilen-Steigungsableitungen und die erste Mehrzahl von Spalten-Steigungsableitungen entsprechend der ersten Randkomprimierung in einer vorbestimmten Einzelrandordnung enthalten. Die Darstellung ist vorzugsweise so, wie sie mit Bezug auf die [Fig. 8](#) und [Fig. 9](#) beschrieben wurde.

[0068] Falls am Schritt **194** bestimmt wird, dass ein Unterbrechungsmuster vorliegt, geht das Verfahren weiter zum Schritt **198**, an welchem eine Doppelrand-Ausgabeformatdarstellung für die Parameterwerte des Pixelblocks basierend auf den Ergebnissen der ersten und zweiten Randkomprimierung erzeugt wird. Die Doppelrand-Ausgabeformatdarstellung ordnet die ersten und zweiten Bezugs-Pixelwerte, die ersten und zweiten Bezugs-Zeilensteigungen, die ersten und zweiten Bezugs-Spaltensteigungen, Teile jeder der ersten und zweiten Mehrzahl von Zeilen-Steigungsableitungen, Teile der ersten und zweiten Mehrzahl von Spalten-Steigungsableitungen und eine Unterbrechungspunktangabe für jede Spalte

(oder Zeile in Abhängigkeit von der Wahl der Bezugspixel) in einer vorbestimmten Doppelrandordnung. Die Teile der ersten und zweiten Mehrzahl von Spalten- und Steigungsableitungen, die in der Doppelrand-Ausgabeformatdarstellung enthalten sind, basieren auf der Stelle des vorbestimmten Unterbrechungsmusters in dem Pixelblock. Die Unterbrechungspunktangabe beschreibt die Teile der ersten und zweiten Mehrzahl von Spalten- und Steigungsableitungen, die in der Doppelrand-Ausgabedarstellung enthalten sind.

[0069] Vorzugsweise umfassen auch die Einzel- und Doppelrand-Ausgabeformatdarstellungen eine Angabe, ob die Komprimierung für den Pixelblock eine Einzelrandkomprimierung oder eine Doppelrandkomprimierung ist. Ein einzelnes Bit könnte enthalten sein, um diese Bestimmung zu machen. Eine Darstellung der verschiedenen Ableitungssteigungen und Escape-Werte wird vorzugsweise mit der Einzel- und Doppelrand-Ausgabeformatdarstellungen in der gleichen Weise herbeigeführt, wie dies mit Bezug auf die [Fig. 8](#) und [Fig. 9](#) beschrieben wurde. So kann das Zwei-Bit-Kodiersystem, das für die abgeleiteten Steigungen in einer Einzelrandkomprimierungstechnik beschrieben ist, leicht auf die Doppelrand-Komprimierungstechnik zusammen mit dem Escape-Wert und den besprochenen Zeilen- und Spalten-Steigungswertoptimierungen angewendet werden.

[0070] Das Verfahren in [Fig. 14](#) kann modifiziert werden, um jegliche unnötige Bearbeitung zu vermeiden. Vorzugsweise werden die erste und zweite Randkomprimierung parallel durchgeführt, derart, dass die Bestimmung bezüglich der Existenz eines Unterbrechungsmusters entweder auf der ersten oder zweiten Randkomprimierung basieren kann, da beide das vorbestimmte Unterbrechungspunktmuster innerhalb ihrer abgeleiteten Steigungsfelder angeben werden. Es kann jedoch vorgezogen werden, nur einen Teil der ersten Randkomprimierung durchzuführen, derart, dass, wenn die erste spaltenweise abgeleitete Steigungsdarstellung erzeugt wird, diese dann untersucht werden kann, um zu bestimmen, ob eine Einzel- oder Doppelrandkomprimierung für den speziellen Pixelblock geeignet ist. Wenn ein solches Verfahren verwendet wird, kann die Erfassung des Unterbrechungspunktmusters in dem Verfahren früher auftreten und wenn eine Einzelrandkomprimierung verwendet wird, um einen Pixelblock zu komprimieren, kann die Durchführung eines Teils der zweiten Randkomprimierung vermieden werden.

[0071] Die Komprimierungstechnik für die Parameterwerte der Pixelblöcke innerhalb eines Einzelbildes, wie sie hier beschrieben wurde, kann die Größenanforderungen zum Speichern und Zurückladen der Parameterwerte aus dem Speicher stark verringern. Es wird darauf hingewiesen, dass die Komprimierung auch den Speicherplatz reduziert, der benö-

tigt wird, um die Parameterwerte für einen speziellen Pixelblock zu speichern, wenn die Komprimierung erfolgreich ist.

[0072] Es sei angemerkt, dass in einigen Fällen eine Komprimierung eines speziellen Pixelblocks dadurch nicht vorteilhaft sein kann, dass die sich daraus ergebende Ausgabeformat-Darstellung in der Tat größer hinsichtlich der Anzahl von Bits ist, als dies erforderlich wäre, um die Parameterwerte in einem unkomprimierten Zustand zu speichern. Aufgrund dessen machen Systeme, welche die hier beschriebene Komprimierungstechnik verwenden, vorzugsweise eine Feststellung dahingehend, ob eine Komprimierung tatsächlich im Hinblick auf einen Pixelblock möglich ist oder ob tatsächlich eine Expansion das Ergebnis wäre. Solche Systeme können dann das Speichern einer expandierten Version der Pixelblöcke in dem Speicher zusammen mit der Angabe, dass der Block unkomprimiert ist, vermeiden. Zudem kann festgestellt werden, ob ein Speicher-Block-Fetching eine Komprimierung weniger günstig macht. Falls der minimale Speicher-Block-Fetch 64 Bits beträgt und der Komprimierungsalgorithmus die Parameterdaten von 92 Bits auf 70 Bits komprimiert, werden keinen realen Größensparungen realisiert, da zwei Block-Fetches nach wie vor erforderlich sein werden. So wird die Entscheidung, ob das Komprimieren eines Parameterdatensatzes erfolgen soll oder nicht auch auf der Basis einer Speicher-Zugriffsgranularität erfolgen muss.

[0073] Wie bereits ausgeführt wurde, ist die räumliche Ausrichtung des Pixelblocks in Bezug zu einer „Spalten“-Steigungsberechnung frei wählbar. So können Bezugspixel in jeder der vier Ecken gewählt werden, und die erste Komprimierungsstufe in eine spaltenweise abgeleitete Steigungsdarstellung kann tatsächlich eine Transformation in eine zeilenweise abgeleitete Spaltendarstellung sein. Es sollte auch für den Fachmann deutlich sein, dass andere Typen von Kodierschemen, wie eine Huffman-Kodierung und dergleichen verwendet werden könnte, um einen begrenzten Satz von Werten für jede der abgeleiteten Steigungen in der Ausgabeformat-Darstellung zu kodieren. Die hier beschriebene Zwei-Bit-Kodierung ist nur ein Beispiel eines solchen Kodierschemas. Ferner kann es für große Blöcke vorteilhaft sein, die vier Eckenpixel als Bezugspixel zu verwenden und sich in Richtung des Zentrums des Blocks vorzuarbeiten, in einer Weise, die eine Extrapolation der Verwendung von zwei Bezugspixeln und das Vorarbeiten in Richtung einer Linie zwischen den zwei Bezugspixeln reflektiert.

Figurenbeschreibung

[Fig. 1](#)

Figure = Figur

Block **22** = Block **22**
 Pixel Blocks **20** = Pixelblöcke **20**
 Frame **10** = Einzelbild **10**

[Fig. 2](#)

Figure = Figur
 Field Block **46** = Feldblock **46**
 Vertex Block **42** = Eckblock **42**
 Edge Block **44** = Randblock **44**
 Primitive **40** = Grundelement **40**

[Fig. 3](#)

Figure = Figur
 Processing Module **52** = Bearbeitungsmodul **52**
 Memory **54** = Speicher **54**
 Processor **50** = Prozessor **50**

[Fig. 5](#)

Figure = Figur
 Pixels **80** = Pixel **80**
 Parameter Values **81** = Parameterwerte **81**
 Block **22** = Block **22**

[Fig. 4](#)

Figure = Figur
 Group pixels of the display frame into a plurality of pixel blocks **62** = Gruppiere Pixels des Einzelbildes in eine Mehrzahl von Pixelblöcken **62**
 For at least one pixel block: = Für wenigstens einen Pixelblock:
 Translate parameter values into a column-wise differential slope representation **64** = Wandle Parameterwerte in eine spaltenweise abgeleitete Steigungsdarstellung um **64**
 Translate the column-wise differential slope representation into a planar differential slope representation **66** = Forme die spaltenweise abgeleitete Steigungsdarstellung in eine planare abgeleitete Steigungsdarstellung um **66**
 Generate an output format representation from the planar differential slope representation **68** = Erzeuge eine Ausgabeformat-Darstellung von der planaren abgeleiteten Steigungsdarstellung **68**
 Store the output format representation in memory **70** = Speichere die Ausgabeformat-Darstellung im Speicher **70**
 Retrieve the output format representation from the memory **72** = Lade die Ausgabeformat-Darstellung aus dem Speicher zurück **72**
 Reconstruct the planar differential slope representation from the output format representation **74** = Rekonstruiere die planare abgeleitete Steigungsdarstellung aus der Ausgabeformat-Darstellung **74**
 Reconstruct the column-wise differential slope representation from the planar differential slope representation **76** = Rekonstruiere die spaltenweise abgeleitete

te Steigungsdarstellung aus der planaren abgeleiteten Steigungsdarstellung **76**
 Recover the parameter values for the pixel block from the column-wise differential slope representation **78** = Gewinne die Parameterwerte für den Pixelblock aus der spaltenweisen abgeleiteten Steigungsdarstellung zurück

[Fig. 6](#)

Figure = Figur
 Column Slopes **84** = Spaltensteigungen **84**
 Column-wise Differential Slope Representation **88** = Spaltenweise abgeleitete Steigungsdarstellung **88**
 Pixel Point Values **82** = Pixelpunktswerte **82**
 Column Slope Differentials **86** = Spalten-Steigungsableitungen **86**

[Fig. 7](#)

Figure = Figur
 Reference Row Slope **94** = Bezugs-Zeilensteigung **94**
 Reference Pixel Value **92** = Bezugspixelwert **92**
 Reference Column Slope **96** = Bezugsspaltensteigung **96**
 Planar Differential Slope Representation **99** = Planare abgeleitete Steigungsdarstellung **99**
 Column Slope Differentials **86** = Spalten-Steigungsableitungen **86**
 Row Slope Differentials **98** = Zeilen-Steigungsableitungen **98**

[Fig. 8](#)

Figure = Figur
 Output Format Representation **100** = Ausgabeformat-Darstellung **100**
 Reference Pixel **32** bits 102 = Bezugspixel **32** Bits 102
 Reference Row Slope **33** bits 104 = Bezugs-Zeilensteigung **33** Bits 104
 Reference Column Slope **33** bits 106 = Bezugs-Spaltensteigung **33** Bits 106
 Encoded Differential Slope Array **122** Bits 108 = Kodierte abgeleitete Steigungsanordnung **122** Bits
 Escape Values **34** bits * (Number of Escapes) 110 = Escape-Werte **34** Bits * (Anzahl von Escapes 110)

[Fig. 9](#)

Figure = Figur
 Output Format Representation **120** = Ausgabeformat-Darstellung **120**
 Reference Pixel **32** bits 102 = Bezugspixel **32** Bits 102
 Bits/slope 5 bits 122 = Bits/Steigung 5 Bits 122
 Reference Row Slope N bits 124 = Bezugs-Zeilensteigung N-Bits 124
 Reference Column Slope N bits 126 = Bezugs-Spal-

tensteigung N-Bits 126
 Encoded Differential Slope Array **122** bits 108 = Ko-
 dierte abgeleitete Steigungsanordnung **122** Bits 108
 Bits/escape 5 bits 128 = Bits/Escapes 5 Bits 128
 Escape Values M bits * (Number of Escapes) =
 Escpae-Werte M Bits * (Anzahl von Escapes)

[Fig. 10](#)

Figure = Figur
 Pixel Block **140** = Pixelblock **140**

[Fig. 11](#)

Figure = Figur
 Column Slopes **148** = Spaltensteigungen **148**
 Column-wise Differential Slope Representation **150** =
 Spaltenweise abgeleitete Steigungsdarstellung **150**
 Pixel Point Values **146** = Pixelpunktwerte **146**
 Column Slope Differentials **144** = Spalten-Stei-
 gungsableitungen **144**

[Fig. 12](#)

Figure = Figur
 Reference Row Slope **166** = Bezugs-Zeilensteigung
166
 Reference Pixel Value **162** = Bezugspixelwert **162**
 Reference Column Slope **164** = Bezugsspaltenstei-
 gung **164**
 Planar Differential Slope Representation **160** = Pla-
 nare abgeleitete Steigungsdarstellung **160**
 Column Slope Differentials **144** = Spalten-Stei-
 gungsableitungen **144**
 Row Slope Differentials **168** = Zeilen-Steigungsablei-
 tungen **168**

[Fig. 13](#)

Figure = Figur
 First Slope **174** = Erste Steigung **174**
 Pixel Point **182** = Pixelpunkt **182**
 Third Slope **178** = Dritte Steigung **178**
 Pixel Point **180** = Pixel Punkt **180**
 Second Slope **176** = Zweite Steigung **176**
 Pixel Divixions **172** = Pixelunterteilungen **172**
 Pixel Block Cross-section **171** = Pixelblock-Quer-
 schnitt **171**

[Fig. 14](#)

Figure = Figur
 Perform a first edge compression **190** = Führe eine
 erste Randkomprimierung durch **190**
 Perform a second edge Compression **192** = Führe
 eine zweite Randkomprimierung durch **192**
 Is there a breakpoint pattern? **194** = Gibt ein Unter-
 brechungspunktmuster? **194**
 No = Nein
 Yes = Ja

Generate a single edge output format representati-
 on for the parameter values of the pixel block from the
 results if the first edge compression **196** = Erzeuge
 eine Einzelrand-Ausgabeformatdarstellung für die
 Parameterwerte des Pixelblocks aus den Ergebnis-
 sen der ersten Randkomprimierung **196**

Generate a double edge output format representation
 for the parameter values of the pixel block from the
 results of the first and second edge compressions
198 = Erzeuge eine Doppelrand-Ausgabeformatdar-
 stellung für die Parameter= werte des Pixelblocks
 aus den Ergebnissen der ersten und zweiten Rand-
 komprimierung **198**

[Fig. 15](#)

Figure = Figur
 First Column-wise Differential Slope Representation
202 = Erste spaltenweisen abgeleitete Steigungs-
 darstellung **202**

[Fig. 16](#)

Figure = Figur
 Second Column-wise Differential Slope Representa-
 tion **204** = Zweite spaltenweisen abgeleitete Stei-
 gungsdarstellung **204**

[Fig. 17](#)

Figure = Figur
 Breakpoint **220** = Unterbrechungspunkt **220**
 Double Edge Planar Differential Slope Representati-
 on **210** = Planare abgeleitete Doppelrand-Steigungs-
 darstellung **210**

Patentansprüche

1. Verfahren zum Komprimieren von Parameter-
 werten (**81**) für Pixel (**80**) eines Pixelblocks (**20**), das
 die folgenden Schritte umfasst:
 Umsetzen (**64**) von Parameterwerten (**81**) für den Pi-
 xelblock (**20**) in eine spaltenweise Differentialnei-
 gungsdarstellung, die die Parameterwerte als eine
 Mehrzahl von Referenzpunkten, eine Mehrzahl von
 Neigungen und eine Mehrzahl von Neigungsdifferen-
 tialen darstellt;
 Umsetzen (**66**) der spaltenweisen Differentialnei-
 gungsdarstellung in eine planare Differentialnei-
 gungsdarstellung, die die planaren Werte mittels we-
 nigstens eines Referenzpixelwertes, zweier Refer-
 enzneigungen und einer Mehrzahl von Neigungsdif-
 ferentialen darstellt; und
 Erzeugen (**68**) einer Ausgangsformatdarstellung der
 Parameterwerte (**81**) für den Pixelblock (**20**) von der
 planaren Differentialneigungsdarstellung.

2. Verfahren für eine Doppelrandkompression von
 Parameterweren (**81**) für Pixel (**80**) eines Pixel-
 blocks (**20**), das die folgenden Schritte umfasst:

Ausführen einer ersten Randkompression, wobei die erste Randkompression Folgendes beinhaltet:

Umsetzen (64) von Parameterwerten (81) für den Pixelblock (20) in eine erste spaltenweise Differentialneigungsdarstellung, wobei die erste spaltenweise Differentialneigungsdarstellung eine erste Mehrzahl von Pixelpunktwerten entlang einem ersten Rand des Pixelblocks, eine erste Mehrzahl von Spaltenneigungen und eine erste Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential aus der ersten Mehrzahl von Spaltenneigungsdifferentialen eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die erste Mehrzahl von Pixelpunktwerten und die erste Mehrzahl von Spaltenneigungen repräsentiert wird; und

Umsetzen (66) der ersten spaltenweisen Differentialneigungsdarstellung in eine erste planare Differentialneigungsdarstellung, wobei die erste planare Neigungsdarstellung die erste Mehrzahl von Pixelpunktwerten und die erste Mehrzahl von Spaltenneigungen zu einem ersten Referenzpixelwert, einer ersten Referenzspaltenneigung, einer ersten Referenzreihenneigung und einer ersten Mehrzahl von Reihenneigungsdifferentialen reduziert;

Ausführen einer zweiten Randkompression, wobei die zweite Randkompression Folgendes beinhaltet:

Umsetzen (64) wenigstens eines Teils der Parameterwerte (81) für den Pixelblock (20) in eine zweite spaltenweise Differentialneigungsdarstellung, wobei die zweite spaltenweise Differentialneigungsdarstellung eine zweite Mehrzahl von Pixelpunktwerten entlang einem zweiten Rand des Pixelblocks, eine zweite Mehrzahl von Spaltenneigungen und eine zweite Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential der zweiten Mehrzahl von Spaltenneigungsdifferentialen eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die zweite Mehrzahl von Pixelpunktwerten und die zweite Mehrzahl von Spaltenneigungen repräsentiert wird; und

Umsetzen (66) der zweiten spaltenweisen Differentialneigungsdarstellung zu einer zweiten planaren Differentialneigungsdarstellung, wobei die zweite planare Neigungsdarstellung die zweite Mehrzahl von Pixelpunktwerten und die zweite Mehrzahl von Spaltenneigungen zu einem zweiten Referenzpixelwert, einer zweiten Referenzspaltenneigung, einer zweiten Referenzreihenneigung und einer zweiten Mehrzahl von Reihenneigungsdifferentialen reduziert; und

Erzeugen (66) einer Ausgangsformatdarstellung für den Pixelblock (20), wobei das Erzeugen der Ausgangsformatdarstellung Folgendes beinhaltet:

wenn ein vorbestimmtes Knickpunktmuster in der ersten oder der zweiten Mehrzahl von Spaltenneigungsdifferentialen erfasst wird, Erzeugen einer Doppelrand-Ausgangsformatdarstellung der Parameterwerte für die erste und die zweite spaltenweise Differentialspaltenneigungsdarstellung, wobei die Doppelrand-Ausgangsformatdarstellung die ersten

und zweiten Referenzpixelwerte, die ersten und zweiten Referenzreihenneigungen, die ersten und zweiten Referenzspaltenneigungen, Teile jedes aus der ersten und zweiten Mehrzahl von Reihenneigungsdifferentialen, Teile der ersten und zweiten Mehrzahl von Spaltenneigungsdifferentialen und eine Knickpunktanzeige in einer vorbestimmten Doppelrandreihenfolge ordnet, wobei die Teile der ersten und zweiten Mehrzahl von Spalten- und Reihenneigungsdifferentialen, die in der Doppelrand-Ausgangsformatdarstellung enthalten sind, auf der Basis der Stelle des vorbestimmten Knickpunktmusters im Pixelblock ermittelt werden, wobei die Knickpunktanzeige die Teile der ersten und zweiten Mehrzahl von Spalten- und Reihenneigungsdifferentialen beschreibt; und

wenn das vorbestimmte Knickpunktmuster nicht erfasst wird, Erzeugen einer Einzelrand-Ausgangsformatdarstellung der Parameterwerte für den Pixelblock, wobei die Einzelrand-Ausgangsformatdarstellung den ersten Referenzpixelwert, die erste Referenzspaltenneigung, die erste Referenzreihenneigung, die erste Mehrzahl von Reihenneigungsdifferentialen und die erste Mehrzahl von Spaltenneigungsdifferentialen in einer vorbestimmten Einzelrandreihenfolge ordnet.

3. Verfahren nach Anspruch 2, wobei das Erzeugen einer Ausgangsformatdarstellung für den Pixelblock ferner das Einbeziehen einer Anzeige beinhaltet, dass die Kompression für den Pixelblock entweder eine Einzelrandkompression oder eine Doppelrandkompression ist.

4. Verfahren nach Anspruch 2 oder Anspruch 3, wobei das Erfassen des vorbestimmten Knickpunktmusters ferner das Erfassen einer Neigungsdiskontinuität in der ersten oder der zweiten Mehrzahl von Spaltenneigungsdifferentialen umfasst.

5. Verfahren nach Anspruch 1 zum Komprimieren von Parameterwerten für Pixel in einem Anzeigerahmen, wobei jedes Pixel im Anzeigerahmen einen entsprechenden Parameterwert hat, umfassend das Gruppieren von Pixeln des Anzeigerahmens zu einer Mehrzahl von Pixelblöcken und das Komprimieren der Parameterwerte für wenigstens einen Pixelblock aus der Mehrzahl von Pixelblöcken, wobei:

jedes Spaltenneigungsdifferential eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen repräsentiert wird; und

die Ausgangsformatdarstellung den Referenzpixelwert, die Referenzspaltenneigung, die Referenzreihenneigung, die Mehrzahl von Reihenneigungsdifferentialen und die Mehrzahl von Spaltenneigungsdifferentialen zu einem vorbestimmten Muster ordnet, indem eines der vier Eckpixel als erstes Referenzpixel für den ersten Pixelblock ausgewählt wird.

6. Verfahren nach Anspruch 5, das ferner das Speichern der Ausgangsformatdarstellung des wenigstens einen Pixelblocks in einem Speicher umfasst.

7. Verfahren nach Anspruch 6, das ferner Folgendes beinhaltet:

Abrufen der Ausgangsformatdarstellung des wenigstens einen Pixelblocks aus dem Speicher;
Rekonstruieren der planaren Differentialneigungsdarstellung für den Pixelblock von der Ausgangsformatdarstellung;
Rekonstruieren der spaltenweisen Differentialneigungsdarstellung für den Pixelblock von der planaren Differentialneigungsdarstellung; und
Wiederherstellen der Parameterwerte für die Pixel in dem Pixelblock von der spaltenweisen Differentialneigungsdarstellung.

8. Verfahren nach einem der vorherigen Ansprüche, wobei die Parameterwerte Z-Werte für die Pixel des Pixelblocks sind.

9. Verfahren nach einem der vorherigen Ansprüche, das ferner das Darstellen von Spaltenneigungsdifferentialen und Reihenneigungsdifferentialen in der Ausgangsformatdarstellung mittels einer ersten Anzahl von Codierungsbits umfasst, wobei dann, wenn die erste Zahl von Codierungsbits nicht zum Darstellen eines der Differentiale ausreicht, ein Codierungswert, der einem Escape entspricht, innerhalb einer Position in der Ausgangsformatdarstellung verwendet wird, die dem Differential entspricht, wobei das Escape anzeigt, dass ein Escape-Wert für das Differential in der Ausgangsformatdarstellung enthalten ist, wobei der Escape-Wert durch eine zweite Anzahl von Bits codiert wird, die ausreicht, um das Differential vollständig zu codieren.

10. Verfahren nach Anspruch 9, wobei die zweite Zahl für den Pixelblock auf der Basis einer Anzahl von Bits bestimmt wird, die ausreicht, um ein größtes Differential völlig zu codieren, für das eine Escape-Wertcodierung innerhalb des Pixelblocks erforderlich ist, wobei ein erster Satz von Konfigurationsbits in der Ausgangsformatdarstellung die zweite Zahl anzeigt.

11. Verfahren nach Anspruch 9 oder 10, wobei die erste Anzahl zwei ist, so dass die erste Anzahl von Codierungsbits vier Zustände codiert, wobei die vier Zustände ein Nulldifferential, ein Positive-Eins-Differential, ein Negative-Eins-Differential und den dem Escape entsprechenden Codierungswert beinhalten.

12. Verfahren nach einem der Ansprüche 9 bis 11, das ferner das Darstellen der Referenzspaltenneigung und der Referenzreihenneigung mit einer dritten Anzahl von Bits repräsentiert, wobei die dritte

Anzahl auf der Basis einer Zahl von Bits bestimmt wird, die ausreicht, um eine größte Neigung der Referenzspaltenneigung und der Referenzreihenneigung vollständig zu codieren, wobei ein zweiter Satz von Konfigurationsbits in der Ausgangsformatdarstellung die dritte Anzahl anzeigt.

13. Verfahren nach Anspruch 1, wobei die spaltenweise Differentialneigungsdarstellung eine Mehrzahl von Pixelpunktwerten, eine Mehrzahl von Spaltenneigungen und eine Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen repräsentiert wird; wobei die planare Neigungsdarstellung die Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen auf einen Referenzpixelwert, eine Referenzspaltenneigung, eine Referenzreihenneigung und eine Mehrzahl von Reihenneigungsdifferentialen reduziert; und wobei die Ausgangsformatdarstellung den Referenzpixelwert, die Referenzspaltenneigung, die Referenzreihenneigung, die Mehrzahl von Reihenneigungsdifferentialen und die Mehrzahl von Spaltenneigungsdifferentialen zu einem vorbestimmten Muster ordnet.

14. Pixelparameterkompressionsprozessor (50) zum Komprimieren von Parameterwerten (81) für Pixel eines Pixelblocks (20), der Folgendes umfasst: ein Verarbeitungsmodul (52); und einen Speicher (54), der betriebsmäßig mit dem Verarbeitungsmodul (52) gekoppelt ist, wobei der Speicher (54) Betriebsbefehle speichert, die bei Ausführung durch das Verarbeitungsmodul (52) bewirken, dass das Verarbeitungsmodul (52) die folgenden Funktionen ausführt:

Umsetzen von Parameterwerten (81) für den Pixelblock (20) in eine spaltenweise Differentialneigungsdarstellung, die die Parameterwerte (81) als eine Mehrzahl von Referenzpunkten, eine Mehrzahl von Neigungen und eine Mehrzahl von Neigungsdifferentialen darstellt;

Umsetzen der spaltenweisen Differentialneigungsdarstellung zu einer planaren Differentialneigungsdarstellung, die die planaren Werte mittels wenigstens eines Referenzpixelwertes, zweier Referenzneigungen und einer Mehrzahl von Neigungsdifferentialen repräsentiert; und

Erzeugen einer Ausgangsformatdarstellung der Parameterwerte (81) für den Pixelblock (20) von der planaren Differentialneigungsdarstellung.

15. Pixelparameterkompressionsprozessor (50) zum Komprimieren von Parameterwerten (81) für Pixel eines Blocks (20) von Pixeln (80), der Folgendes umfasst: ein Verarbeitungsmodul (52); und einen Speicher (54), der betriebsmäßig mit dem Ver-

arbeitungsmodul (52) gekoppelt ist, wobei der Speicher (54) Betriebsbefehle speichert, die bei Ausführung durch das Verarbeitungsmodul (52) bewirken, dass das Verarbeitungsmodul (52) die folgenden Funktionen ausführt:

Ausführen einer ersten Randkompression, wobei die erste Randkompression Folgendes beinhaltet:

Umsetzen von Parameterwerten (81) für den Pixelblock (20) in eine erste spaltenweise Differentialneigungsdarstellung, wobei die erste spaltenweise Differentialneigungsdarstellung eine erste Mehrzahl von Pixelpunktwerten entlang einem ersten Rand des Pixelblocks, eine erste Mehrzahl von Spaltenneigungen und eine erste Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential aus der ersten Mehrzahl von Spaltenneigungsdifferentialen eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die erste Mehrzahl von Pixelpunktwerten und die erste Mehrzahl von Spaltenneigungen repräsentiert wird; und

Umsetzen der ersten spaltenweisen Differentialneigungsdarstellung in eine erste planare Differentialneigungsdarstellung, wobei die erste planare Neigungsdarstellung die erste Mehrzahl von Pixelpunktwerten und die erste Mehrzahl von Spaltenneigungen zu einem ersten Referenzpixelwert, einer ersten Referenzspaltenneigung, einer ersten Referenzreihenneigung und einer ersten Mehrzahl von Reihenneigungsdifferentialen reduziert;

Ausführen einer zweiten Randkompression, wobei die zweite Randkompression Folgendes beinhaltet:

Umsetzen wenigstens eines Teils der Parameterwerte (81) für den Pixelblock (20) in eine zweite spaltenweise Differentialneigungsdarstellung, wobei die zweite spaltenweise Differentialneigungsdarstellung eine zweite Mehrzahl von Pixelpunktwerten entlang einem zweiten Rand des Pixelblocks, eine zweite Mehrzahl von Spaltenneigungen und eine zweite Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential der zweiten Mehrzahl von Spaltenneigungsdifferentialen eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die zweite Mehrzahl von Pixelpunktwerten und die zweite Mehrzahl von Spaltenneigungen repräsentiert wird; und

Umsetzen der zweiten spaltenweisen Differentialneigungsdarstellung zu einer zweiten planaren Differentialneigungsdarstellung, wobei die zweite planare Neigungsdarstellung die zweite Mehrzahl von Pixelpunktwerten und die zweite Mehrzahl von Spaltenneigungen zu einem zweiten Referenzpixelwert, einer zweiten Referenzspaltenneigung, einer zweiten Referenzreihenneigung und einer zweiten Mehrzahl von Reihenneigungsdifferentialen reduziert; und

Erzeugen einer Ausgangsformatdarstellung für den Pixelblock, wobei das Erzeugen der Ausgangsformatdarstellung Folgendes beinhaltet:

wenn ein vorbestimmtes Knickpunktmuster in der ersten oder der zweiten Mehrzahl von Spaltennei-

gungsdifferentialen erfasst wird, Erzeugen einer Doppelrand-Ausgangsformatdarstellung der Parameterwerte für die erste und die zweite spaltenweise Differentialspaltenneigungsdarstellung, wobei die Doppelrand-Ausgangsformatdarstellung die ersten und zweiten Referenzpixelwerte, die ersten und zweiten Referenzreihenneigungen, die ersten und zweiten Referenzspaltenneigungen, Teile jedes aus der ersten und zweiten Mehrzahl von Reihenneigungsdifferentialen, Teile der ersten und zweiten Mehrzahl von Spaltenneigungsdifferentialen in einem vorbestimmten Muster und einer Knickpunktanzeige ordnet, wobei die enthaltenen Teile, die die Teile der ersten und zweiten Mehrzahl von Spalten- und Reihenneigungsdifferentialen, die in der Doppelrand-Ausgangsformatdarstellung enthalten sind, auf der Basis der Stelle des vorbestimmten Knickpunktmusters im Pixelblock ermittelt werden, wobei die Knickpunktanzeige enthaltene Teile beschreibt; und wenn das vorbestimmte Knickpunktmuster nicht erfasst wird, Erzeugen einer Einzelrand-Ausgangsformatdarstellung der Parameterwerte für den Pixelblock, wobei die Einzelrand-Ausgangsformatdarstellung den ersten Referenzpixelwert, die erste Referenzspaltenneigung, die erste Referenzreihenneigung, die erste Mehrzahl von Reihenneigungsdifferentialen und die erste Mehrzahl von Spaltenneigungsdifferentialen in einer vorbestimmten Reihenfolge ordnet.

16. Prozessor (50) nach Anspruch 15, wobei der Speicher (54) Befehle beinhaltet, so dass das Verarbeitungsmodul (52) die Ausgangsformatdarstellung für den Pixelblock (20) erzeugt, so dass die Ausgangsformatdarstellung eine Anzeige beinhaltet, die anzeigt, dass die Kompression für den Pixelblock (20) eine Einzelrandkompression oder eine Doppelrandkompression ist.

17. Prozessor (50) nach Anspruch 15 oder Anspruch 16, wobei der Speicher (54) Befehle beinhaltet, so dass der Prozessor (52) das vorbestimmte Knickpunktmuster durch Ermitteln einer Neigungsdiskontinuität entlang der ersten oder der zweiten Mehrzahl von Spaltenneigungsdifferentialen erfasst.

18. Pixelparameterkompressionsprozessor (50) nach Anspruch 14 zum Komprimieren von Parameterwerten (81) für Pixel (80) in einem Anzeigerahmen, wobei jedes Pixel (80) in dem Anzeigerahmen einen entsprechenden Parameterwert (81) hat, in dem der Speicher betriebsmäßig mit dem Verarbeitungsmodul gekoppelt ist, wobei der Speicher Betriebsbefehle speichert, die bei Ausführung durch das Verarbeitungsmodul bewirken, dass das Verarbeitungsmodul zusätzlich die Funktion des Gruppierens von Pixeln des Anzeigerahmens zu einer Mehrzahl von Pixelblöcken ausführt, wobei jedes Spaltenneigungsdifferential eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, der nicht durch die

Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen repräsentiert wird, und die Ausgangsformatdarstellung den Referenzpixelwert, die Referenzspaltenneigung, die Referenzreihenneigung, die Mehrzahl von Reihenneigungsdifferentialen und die Mehrzahl von Spaltenneigungsdifferentialen in einem vorbestimmten Muster ordnet, indem sie die vier Eckpixel als erstes Referenzpixel für den ersten Pixelblock auswählt.

19. Prozessor (50) nach Anspruch 18, wobei der Speicher (54) Befehle beinhaltet, so dass das Verarbeitungsmodul die Ausgangsformatdarstellung des wenigstens einen Pixelblocks (20) in einem Pixel-speicher speichert.

20. Prozessor (50) nach Anspruch 18 oder Anspruch 19, wobei der Speicher (54) Befehle beinhaltet, so dass das Verarbeitungsmodul (52) unter anderem die folgenden zusätzlichen Funktionen ausführt: Abrufen der Ausgangsformatdarstellung des wenigstens einen Pixelblocks aus dem Speicher (54); Rekonstruieren der planaren Differentialneigungsdarstellung für den Pixelblock von der Ausgangsformatdarstellung; Rekonstruieren der spaltenweisen Differentialneigungsdarstellung für den Pixelblock von der planaren Differentialneigungsdarstellung; und Wiederherstellen der Parameterwerte für die Pixel in dem Pixelblock von der spaltenweisen Differentialneigungsdarstellung.

21. Prozessor (50) nach einem der Ansprüche 14 bis 20, wobei die Parameterwerte Z-Werte für die Pixel des Pixelblocks sind.

22. Prozessor (50) nach einem der Ansprüche 14 bis 21, wobei der Speicher (54) Betriebsbefehle beinhaltet, so dass das Verarbeitungsmodul (52) Spaltenneigungsdifferentiale und Reihenneigungsdifferentiale in der Ausgangsformatdarstellung mittels einer ersten Anzahl von Codierungsbits darstellt, wobei dann, wenn die erste Anzahl von Codierungsbits nicht zum Darstellen eines der Differentiale ausreicht, ein Escape entsprechender Codierungswert in einer Position in der Ausgangsformatdarstellung benutzt wird, die dem Differential entspricht, wobei das Escape anzeigt, dass ein Escape-Wert für das Differential in der Ausgangsformatdarstellung enthalten ist, wobei der Escape-Wert durch eine zweite Anzahl von Bits codiert wird, die zum vollständigen Codieren des Differentials ausreicht.

23. Prozessor (50) nach Anspruch 22, wobei die zweite Anzahl auf der Basis einer Anzahl von Bits bestimmt wird, die zum vollständigen Codieren eines größten Differentials ausreicht, für das eine Escape-Wertcodierung innerhalb des Pixelblocks erforderlich ist, wobei ein erster Satz von Konfigurationsbits in der Ausgangsformatdarstellung die zweite An-

zahl anzeigt.

24. Prozessor (50) nach Anspruch 22 oder 23, wobei die erste Anzahl zwei ist, so dass die erste Anzahl von Codierungsbits vier Zustände codiert, wobei die vier Zustände ein Nulldifferential, ein Positive-Eins-Differential, ein Negative-Eins-Differential und den dem Escape entsprechenden Codierungswert beinhalten.

25. Prozessor (50) nach einem der Ansprüche 22 bis 24, wobei der Speicher (54) Betriebsbefehle beinhaltet, so dass das Verarbeitungsmodul (52) die Referenzspaltenneigung und die Referenzreihenneigung mittels einer dritten Anzahl von Bits repräsentiert, wobei die dritte Anzahl auf der Basis einer Zahl von Bits bestimmt wird, die zum vollständigen Codieren einer größten Neigung der Referenzspaltenneigung und der Referenzreihenneigung ausreicht, wobei ein zweiter Satz von Konfigurationsbits in der Ausgangsformatdarstellung die dritte Anzahl anzeigt.

26. Prozessor (50) nach Anspruch 25, wobei die spaltenweise Differentialneigungsdarstellung eine Mehrzahl von Pixelpunktwerten, eine Mehrzahl von Spaltenneigungen und eine Mehrzahl von Spaltenneigungsdifferentialen beinhaltet, wobei jedes Spaltenneigungsdifferential eine Spaltenneigungsabweichung an einem Pixelpunkt repräsentiert, die nicht durch die Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen repräsentiert wird; wobei die planare Neigungsdarstellung die Mehrzahl von Pixelpunktwerten und die Mehrzahl von Spaltenneigungen zu einem Referenzpixelwert, einer Referenzspaltenneigung, einer Referenzreihenneigung und einer Mehrzahl von Reihenneigungsdifferentialen reduziert; und wobei die Ausgangsformatdarstellung den Referenzpixelwert, die Referenzspaltenneigung, die Referenzreihenneigung, die Mehrzahl von Reihenneigungsdifferentialen und die Mehrzahl von Spaltenneigungsdifferentialen zu einem vorbestimmten Muster ordnet.

Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

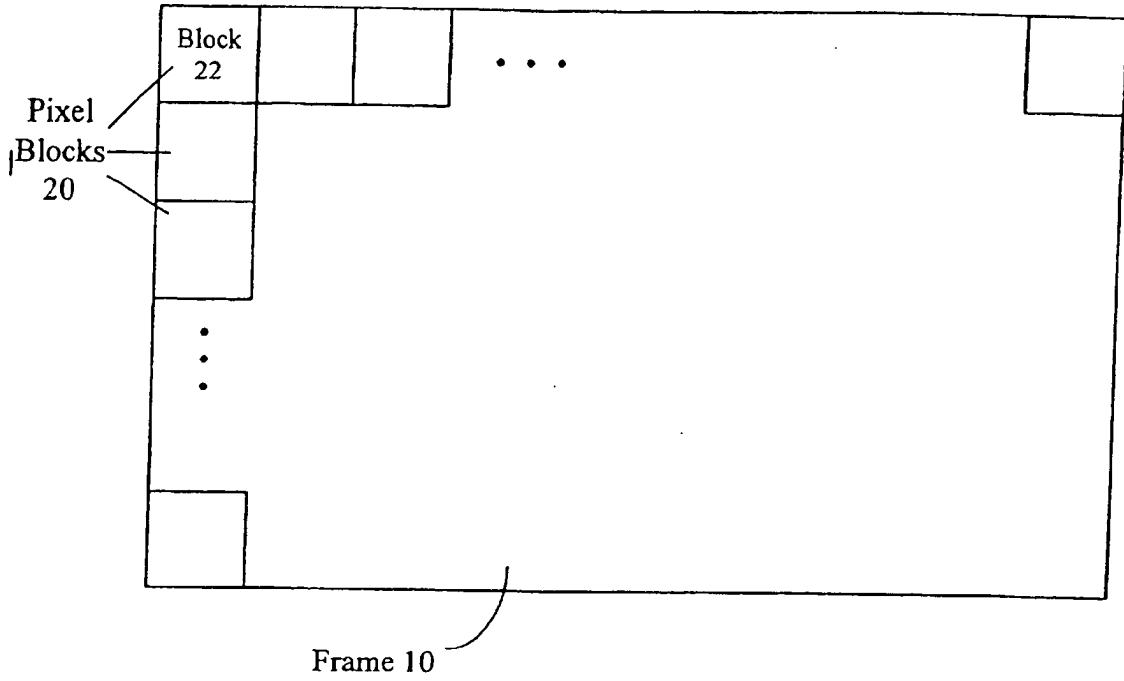


Figure 1.

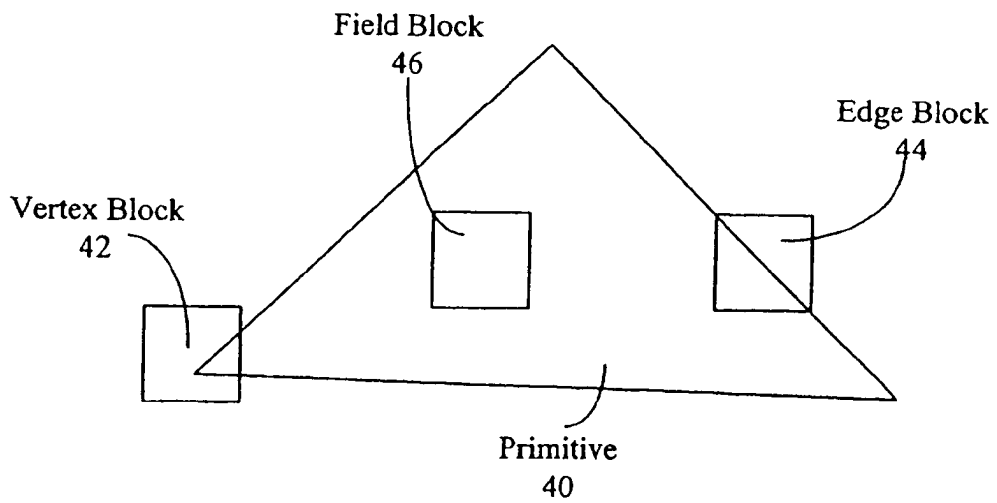


Figure 2.

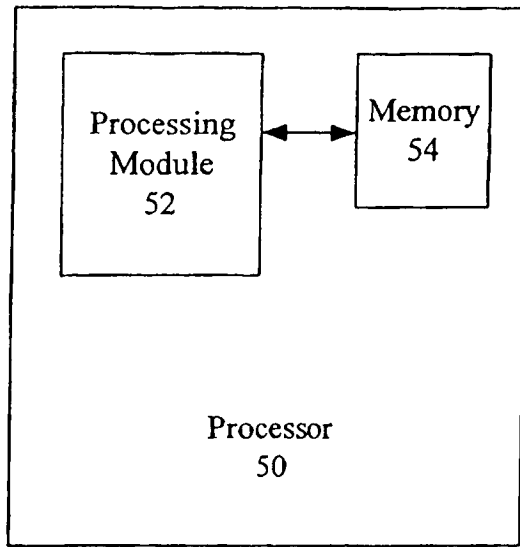


Figure 3.

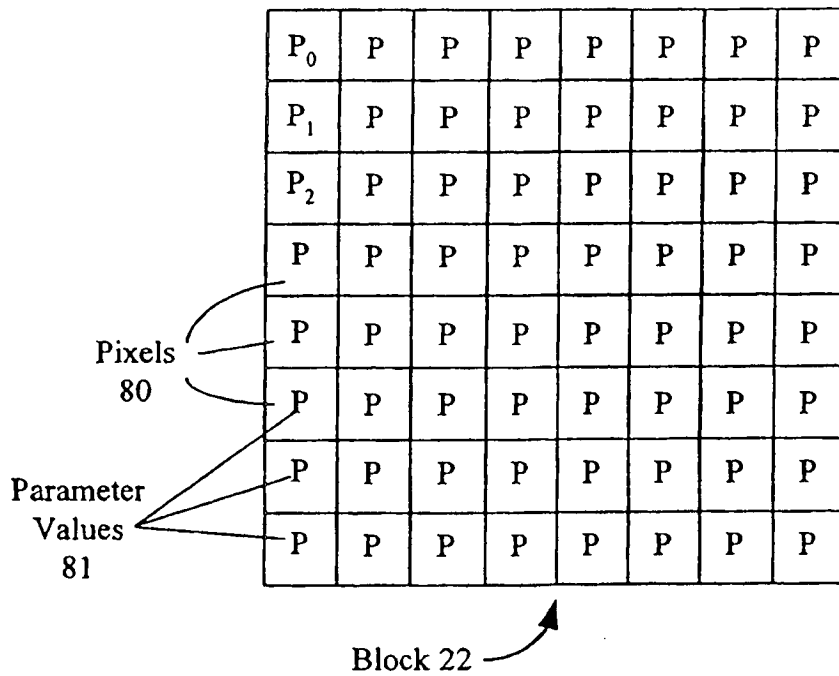


Figure 5.

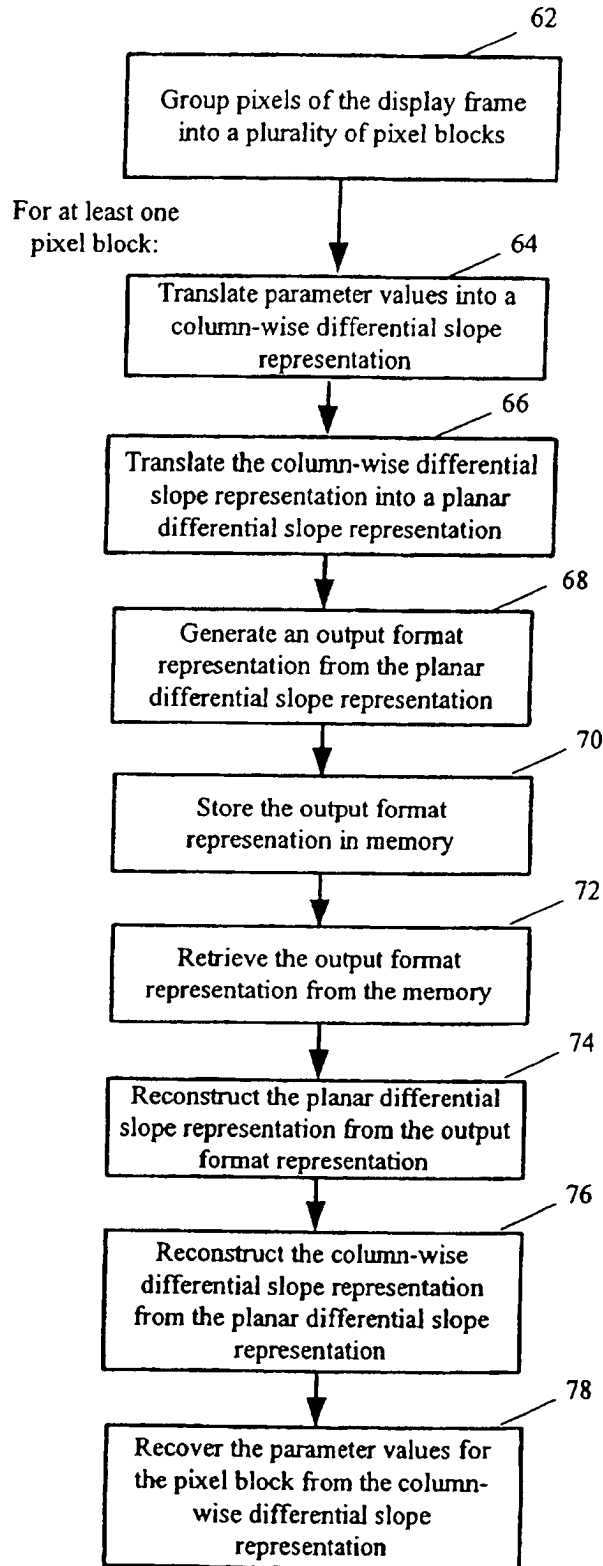


Figure 4.

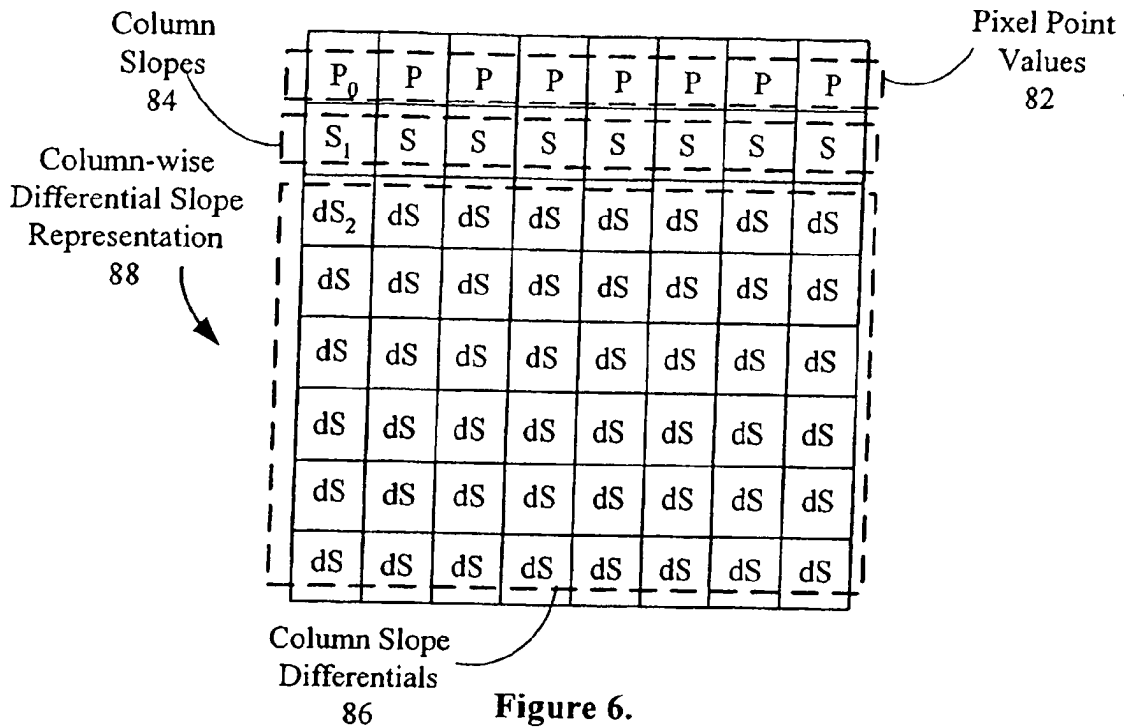


Figure 6.

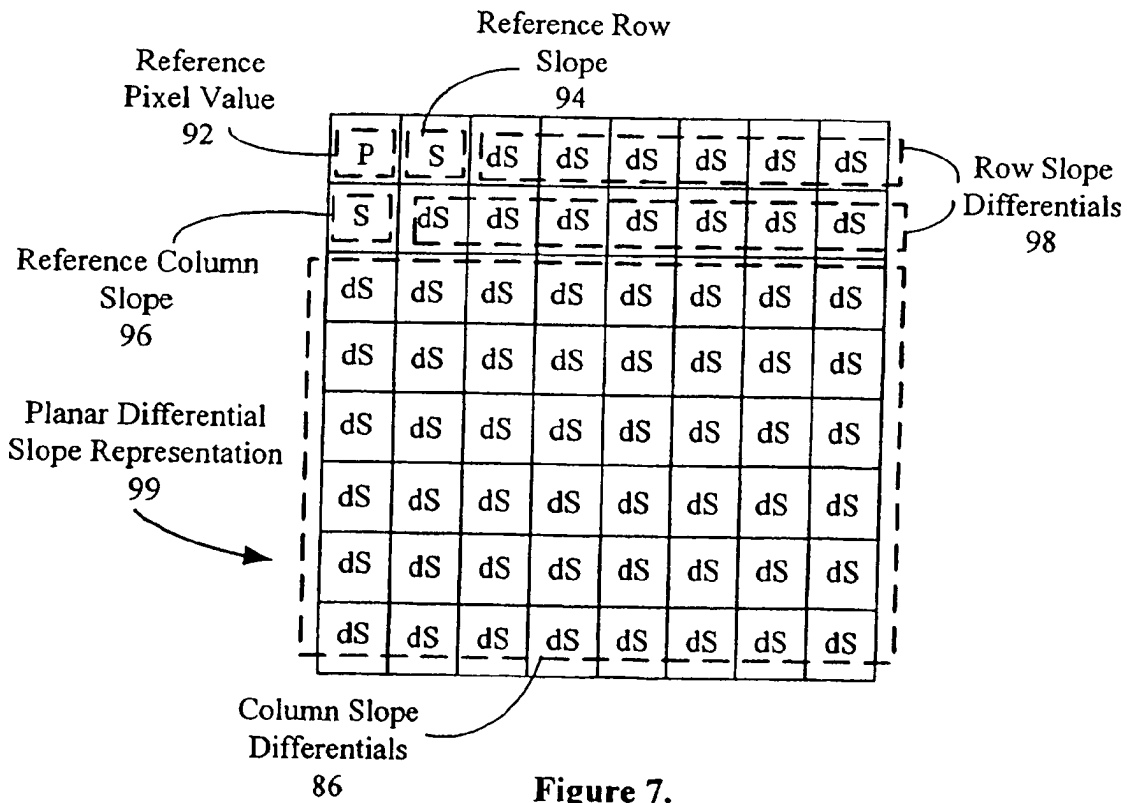


Figure 7.

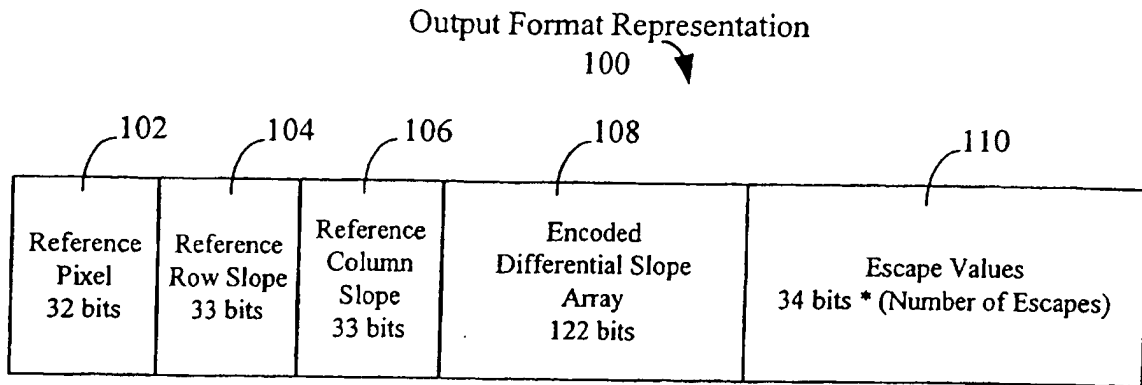


Figure 8.

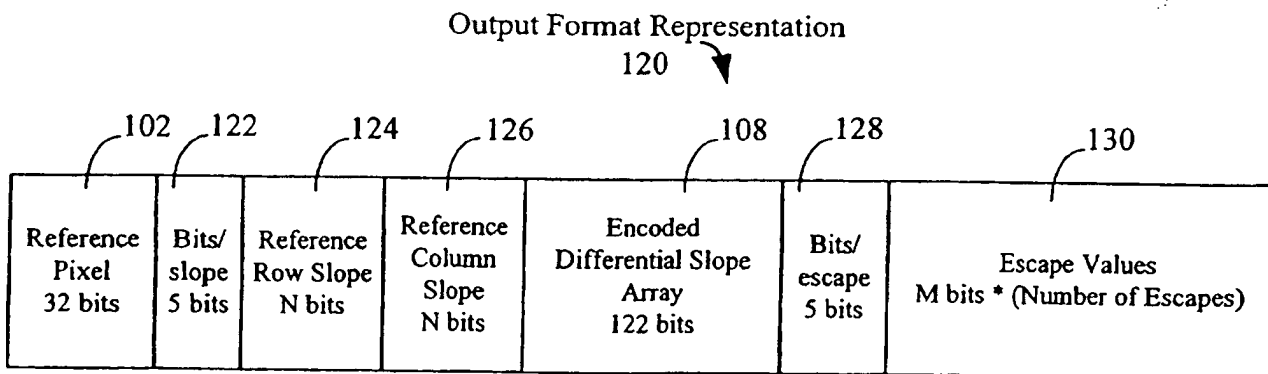


Figure 9.

Pixel Block 140

3	8	13	18	23	28	33	38
4	9	14	19	24	29	34	39
5	10	15	20	25	30	35	40
6	11	16	21	26	31	36	41
7	12	17	22	27	32	37	42
8	13	18	23	28	33	38	43
9	14	19	24	29	34	39	44
10	15	20	25	30	35	40	45

Figure 10.

Column Slopes 148	3	8	13	18	23	28	33	38	Pixel Point Values 146
Column-wise Differential Slope Representation 150	1	1	1	1	1	1	1	1	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
Column Slope Differentials 144									

Figure 11.

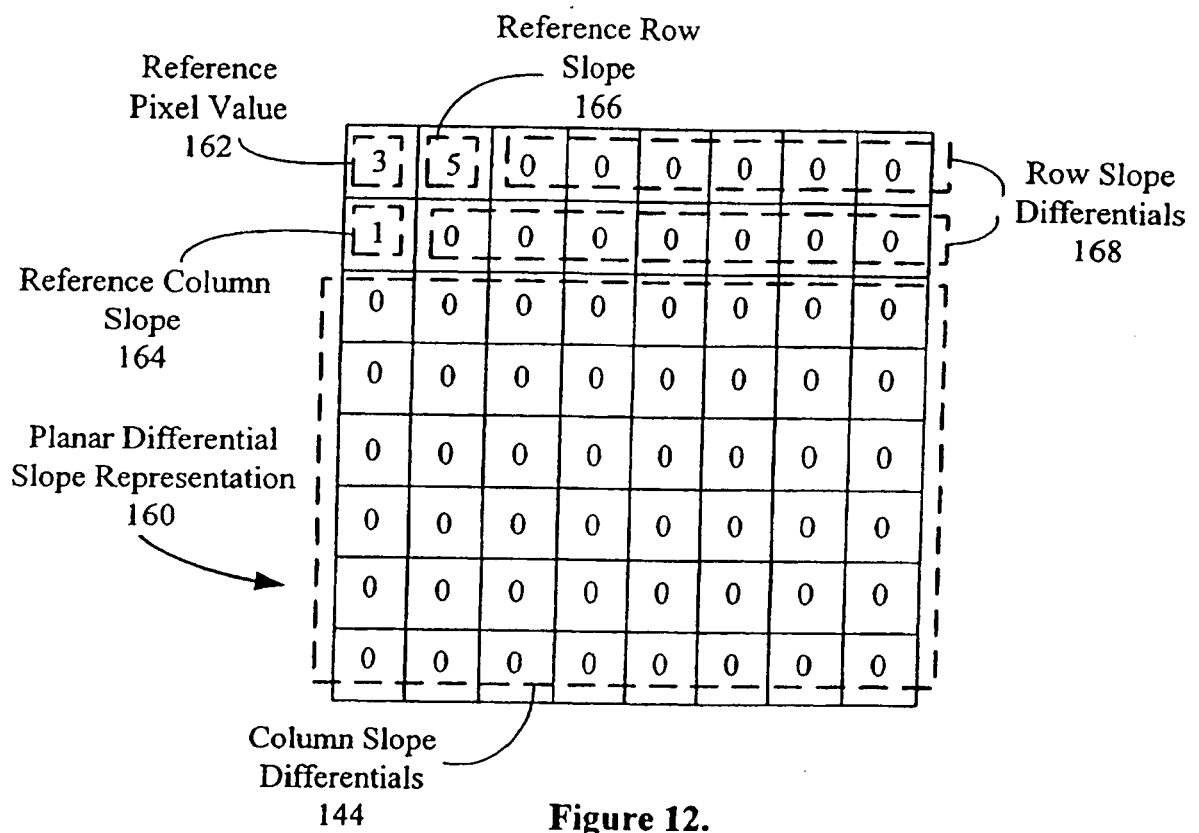


Figure 12.

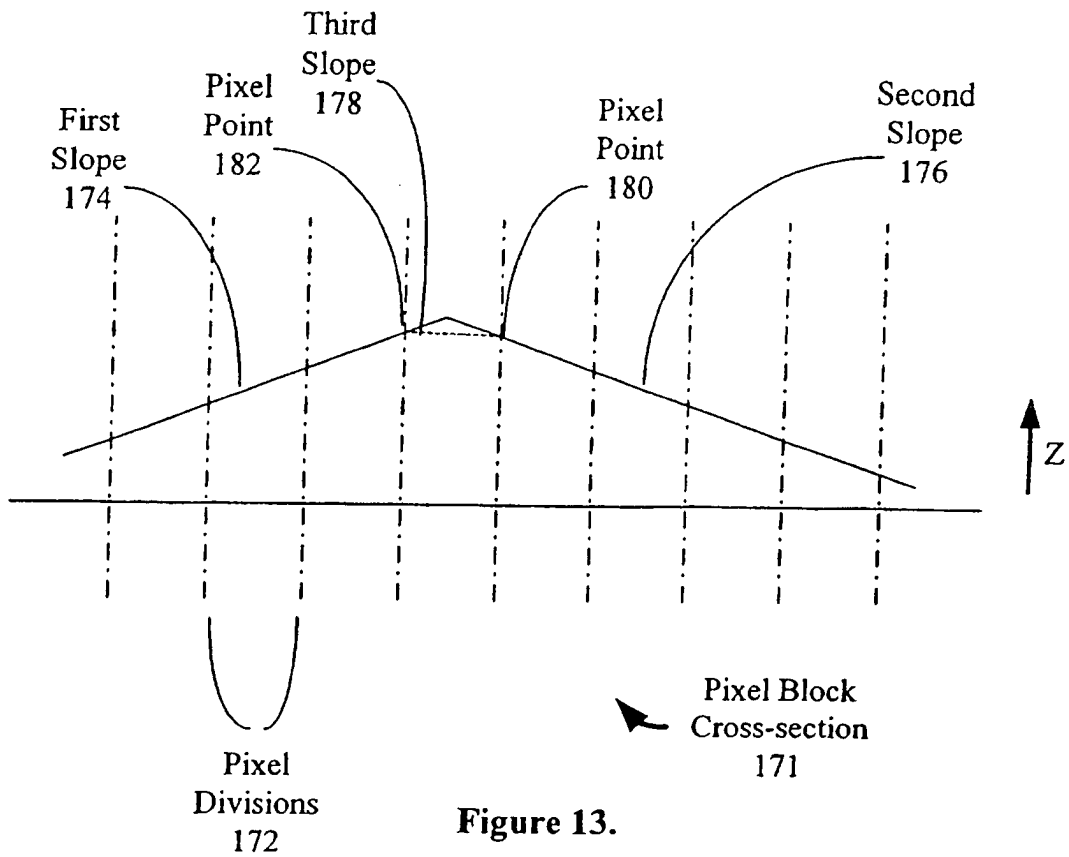


Figure 13.

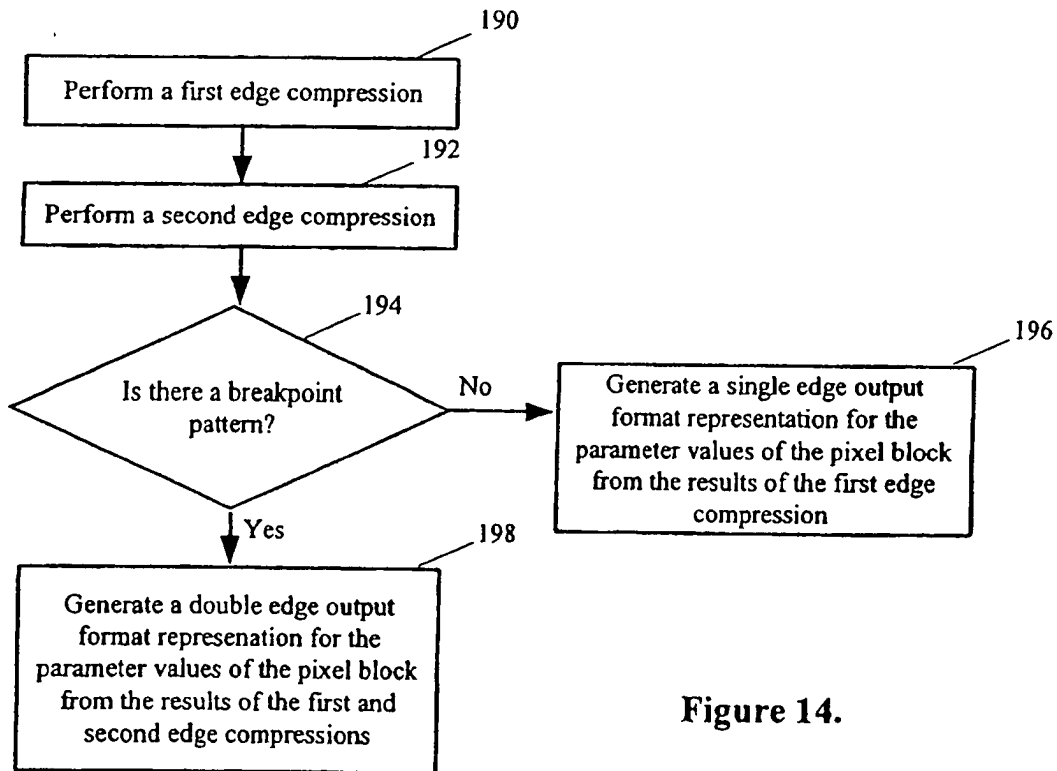


Figure 14.

First Column-wise
Differential Slope
Representation
202

P	P	P	P	P	P	P	P
S	S	S	S	S	S	S	S
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS

Second Column-wise
Differential Slope
Representation
204

dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
dS	dS	dS	dS	dS	dS	dS	dS
S	S	S	S	S	S	S	S
P	P	P	P	P	P	P	P

Figure 15.

Figure 16.

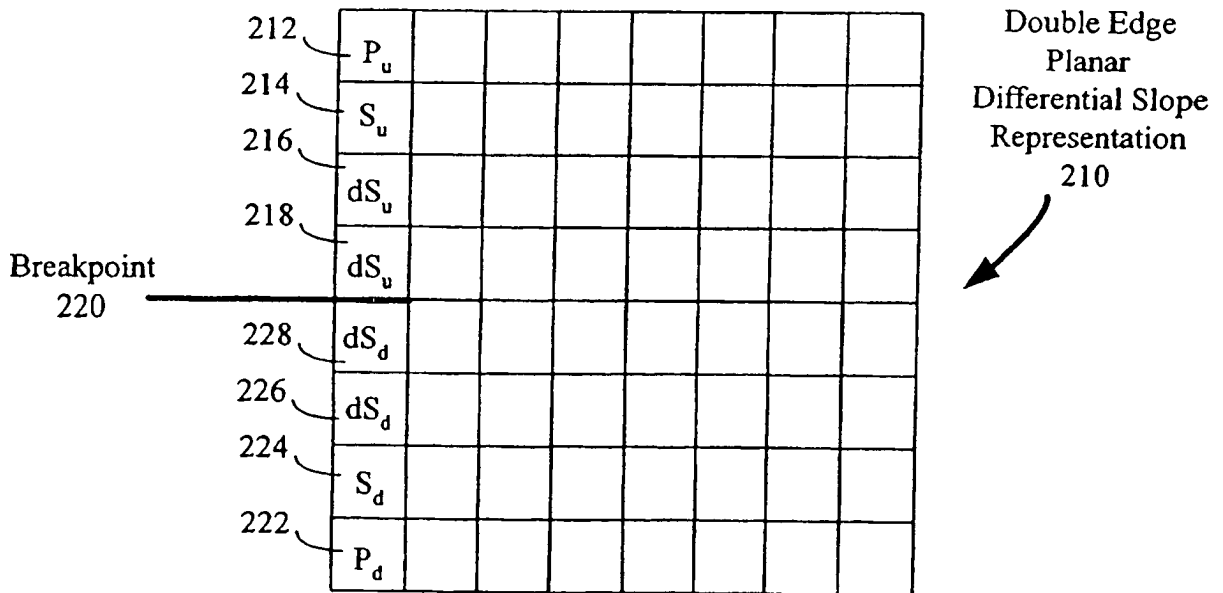


Figure 17.