(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 60 2004 007 928 T2** 2007.11.22

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 639 805 B1**
(21) Deutsches Aktenzeichen: **60 2004 007 928.6**
(86) PCT-Aktenzeichen: **PCT/EP2004/004794**
(96) Europäisches Aktenzeichen: **04 731 326.7**
(87) PCT-Veröffentlichungs-Nr.: **WO 2005/006736**
(86) PCT-Anmeldetag: **06.05.2004**
(87) Veröffentlichungstag der PCT-Anmeldung: **20.01.2005**
(97) Erstveröffentlichung durch das EPA: **29.03.2006**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **01.08.2007**
(47) Veröffentlichungstag im Patentblatt: **22.11.2007**

(51) Int Cl.[8]: ***H04N 1/41*** (2006.01)

(30) Unionspriorität:
03014970 01.07.2003 EP

(73) Patentinhaber:
THOMSON Licensing, Boulogne, FR

(74) Vertreter:
Rittner, K., Dipl.-Phys. Dr.rer.nat., Pat.-Anw., 30826 Garbsen

(84) Benannte Vertragsstaaten:
AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LI, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR

(72) Erfinder:
ADOLPH, Dirk, 30952 Ronnenberg, DE; HÖRENTRUP, Jobst, 30163 Hannover, DE; KOCHALE, Axel, 31832 Springe, DE; OSTERMANN, Ralf, 30657 Hannover, DE; PETERS, Hartmut, 30890 Barsinghausen, DE

(54) Bezeichnung: **VERFAHREN FÜR LAUFLÄNGENKODIERUNG EINES BITMAPDATENSTROMS**

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

Sachgebiet der Erfindung

[0001] Die Erfindung bezieht sich auf ein Verfahren zum Kodieren eines Datenstroms, insbesondere eines Bitmap-kodierten Untertitel-Datenstroms.

Hintergrund

[0002] Fernsehrundfunksendungen oder schreibgeschützte Aufzeichnungsträger, die Videodaten enthalten, können auch Unterbild-Datenströme umfassen, die Text- oder graphische Informationen enthalten, die benötigt werden, um Untertitel, Glyphe oder Animation für irgendeinen besonderen Zweck, z.B. Menü-Schaltflächen, vorzusehen. Da die Anzeige solcher Informationen üblicherweise wirksam oder unwirksam gemacht werden kann, wird sie dem zugeordneten Videobild als zusätzliche Schicht überlagert und als ein oder mehrere rechteckige Bereiche ausgeführt, die als Regionen bezeichnet werden. Für eine solche Region ist eine Reihe von Attributen spezifiziert, wie zum Beispiel Bereichsgröße, Bereichsposition oder Hintergrundfarbe. Da die Region dem Fernsehbild überlagert wird, ist ihr Hintergrund oft als transparent definiert, so dass das Videobild gesehen werden kann oder mehrere Unterbildschichten überlagert werden können. Ferner kann eine Untertitelregion breiter als das zugeordnete Bild sein, so dass nur ein Teil der Unterbildregion sichtbar ist und der sichtbare Teil der Region zum Beispiel von rechts nach links durch den ganzen Untertitelbereich verschoben wird, was so aussieht, als ob die Untertitel durch die Anzeige geschoben würden. Dieses Verfahren von auf Pixel beruhender Untertitelung ist in der europäischen Patentanmeldung EP 02 025 474.4 beschrieben und wird als Zurechtschneiden (cropping) bezeichnet.

[0003] Untertitel waren ursprünglich als Unterstützung von behinderten Personen oder zur Einsparung der Kosten für die Übersetzung eines Films in wenig benutzte Sprachen gedacht, und für reinen Untertiteltext würde es daher ausreichen, wenn der Untertitel-Datenstrom zum Beispiel gemäß ASCII kodierte Zeichen enthält. Heute enthalten Untertitel aber auch andere Elemente bis hinauf zu hochauflösenden Bildern, Glyphen oder animierten graphischen Objekten. Die Handhabung solcher Elemente ist leichter, wenn der Untertitelstrom im Bitmap-Format kodiert wird, bei dem die Zeilen eines Bereiches und die Pixel innerhalb einer Zeile aufeinanderfolgend kodiert und dekodiert werden. Dieses Format enthält viel Redundanz, z.B. wenn aufeinanderfolgende Pixel denselben Farbwert haben. Diese Redundanz kann durch verschiedene Kodierverfahren vermindert werden, z.B. durch Lauflängenkodierung (RLE). RLE wird oft verwendet, wenn Datensequenzen denselben Wert haben, und die Grundideen bestehen darin, die Sequenzlänge und den Wert getrennt zu kodieren und die häufigsten Kodeworte so kurz wie möglich zu kodieren. Insbesondere bei der Kodierung der Untertitelschicht für hochauflösende Videosignale (HDTV) mit 1920×1280 Pixeln wird ein Kodier-Algorithmus benötigt, der für diesen Zweck optimiert ist, um die erforderliche Datenmenge zu vermindern.

Zusammenfassung der Erfindung

[0004] Der Erfindung liegt die Aufgabe zugrunde, ein Verfahren zur optimierten Kodierung von Untertitel- oder Unterbildschichten für hochauflösende Videosignale, wie zum Beispiel HDTV, aufzuzeigen, die als Bitmap-formatierte Bereiche dargestellt werden, die viel breiter als das sichtbare Videobild sein können.

[0005] Die Erfindung ist in den angefügten Ansprüchen dargelegt.

[0006] Gemäß der Erfindung wird zu diesem Zweck eine vierstufige Lauflängen-Kodierung (RLE) verwendet, bei der die kürzesten Kodeworde für einzelne Pixel verwendet werden, die von Transparenz abweichende individuelle Farbwerte haben, bei der die zweitkürzesten Kodeworte für kürzere Sequenzen von transparenten Pixeln verwendet werden, bei der die drittkürzesten Kodeworte für längere Sequenzen von transparenten Pixeln und kürzere Sequenzen von Pixeln mit gleicher, von transparent abweichender Farbe verwendet werden, und bei der die viertkürzesten Kodeworte für längere Sequenzen von Pixeln mit gleicher, von Transparenz abweichender Farbe verwendet werden. Üblicherweise sind die meisten Pixel in der Untertitelschicht transparent. Anders als bei konventioneller RLE, bei der die häufigsten Daten die kürzesten Kodeworte verwenden, umfasst dieses Verfahren die Verwendung der zweitkürzesten Kodeworte für kurze Sequenzen mit der häufigsten Farbe und der drittkürzesten Kodeworte für längere Sequenzen der häufigsten Farbe und auch kurzer Sequenzen mit anderen Farben. Kürzeste Kodeworte werden für einzelne Pixel mit einer anderen als der häufigsten Farbe reserviert. Dies ist vorteilhaft, wenn Pixel mit der häufigsten Farbe fast immer in Sequenzen erscheinen, was der Fall für transparente Pixel in der Untertitelschicht ist, während einzelne Pixel mit individueller Farbe mit größerer Wahrscheinlichkeit nicht transparent sind.

[0007] Vorteilhafterweise enthält ein Kode gemäß dem erfindungsgemäßen Verfahren nur wenige redundante Kodeworte, die definiert sind, unter den längeren Kodeworten zu sein. Zum Beispiel wird ein einzelnes Pixel mit irgendeiner von transparent abweichenden Farbe im Idealfall mit einem Kodewort der kürzesten Art kodiert, aber es kann ebenso ein Kodewort der drittkürzesten Art verwendet werden, wobei die Sequenzlänge gleich eins ist. Obwohl die letztere

Möglichkeit üblicherweise für diesen Zweck nicht benutzt wird, können diese nicht benutzten Kodeworte oder Lücken in dem Kodewortraum für den Transport anderer Informationen verwendet werden. Ein Beispiel ist die Zeilenend-Information, die zur Resynchronisation verwendet werden kann. Gemäß der Erfindung wird das kürzeste redundante Kodewort zum Kodieren dieser Informationen verwendet.

[0008] Ein weiterer Vorteil besteht darin, dass das offenbarte Verfahren die Menge der erforderlichen Daten vermindert und somit den Untertitel-Datenstrom komprimiert, wobei der Kompressionsfaktor von den Inhalten des Datenstroms abhängt. Besonders hohe Kompressionsfaktoren werden für Datenkombinationen erreicht, die sehr oft in typischen Untertitelströmen erscheinen. Das sind Sequenzen mit einer kürzeren Länge als z.B. 64 Pixel, die denselben Farbwert haben, aber auch Sequenzen aus transparenten Pixeln, die irgendeine Länge haben, und einzelne Pixel, die individuelle Farbwerte haben. Die ersten dieser Gruppen werden oft in Zeichen oder Glyphen verwendet, die zweiten dieser Gruppen werden vor, zwischen und nach den angezeigten Elementen des Untertitelstroms verwendet, und die dritte dieser Gruppen wird in Bildern oder Bereichen mit sich leicht ändernder Farbe verwendet. Da transparente Pixel kaum jemals in sehr kurzen Sequenzen erscheinen, z.B. weniger als drei Pixel, reicht es aus, diese nicht mit den kürzesten, sondern nur mit den zweitkürzesten Kodeworten zu kodieren.

[0009] Gleichzeitig kann das erfindungsgemäße Verfahren wirksam Sequenzen handhaben, die länger als 1920 Pixel sind und zum Beispiel bis zu 16.383 Pixel lang sein können, so dass sehr breite Untertitelbereiche ermöglicht werden.

[0010] Ferner erzeugt das Kodierverfahren einen eindeutigen Wert, der das Ende einer Zeile darstellt, und daher ist es für den Fall eines Synchronisationsverlustes möglich, jede Zeile zu resynchronisieren.

[0011] Vorteilhafterweise ist das erfindungsgemäße Verfahren für das Kodieren dieser Kombination einer Anzahl von für Untertitelströme typischen Merkmalen optimiert.

[0012] Daher kann die Menge der für den Untertitelstrom benötigten Daten vermindert werden, was zur besseren Nutzung der Übertragungs-Bandbreite im Fall einer Sendung führt, oder zu einer verminderten Abtaster-Sprungfrequenz im Fall von Speichermedien, bei denen ein einziger Abtaster mehrere Datenströme liest, wie zum Beispiel bei der Blu-Ray Disc(BD)-Technologie. Ferner wird, je besser das Untertitel-Bitmap komprimiert wird, umso höher die Kapazität in Form von Bitrate für Audio- und Videoströme belassen, wodurch die Bild- und Audioqualität erhöht wird.

[0013] Vorteilhafte Ausführungsformen der Erfindung sind in den Unteransprüchen, der folgenden Beschreibung und den Zeichnungen offenbart.

Kurze Beschreibung der Zeichnungen

[0014] Ausführungsbeispiele der Erfindung werden unter Bezugnahme auf die beigefügten Zeichnungen beschrieben. In den Zeichnungen stellen dar:

[0015] Fig. 1 Zuschneiden eines Untertitelbereiches in einem Video-Vollbild;

[0016] Fig. 2 eine Pixelsequenz in einem Untertitelbereich;

[0017] Fig. 3 eine Kodiertabelle für Untertitelung einschließlich Text und Graphiken; und

[0018] Fig. 4 eine Tabelle als Beispiel für eine Syntax eines erweiterten Objekt-Datensegments für die Blu-Ray-Prerecorded Norm.

Ausführliche Beschreibung der Erfindung

[0019] Während Untertitelung in vorproduziertem audiovisuellen (AV)-Material für Rundfunk oder Film-Discs primär zur Darstellung von einfacher statischer Textinformation, z.B. Untertitel, Teletext oder DVB-Untertitel optimiert wird, erfordert der Fortschritt in der Multimedia-Entwicklung für die Darstellung und Animation von Text- und graphischen Informationen, die angemessen für neue HDTV-Formate sind, eine fortschrittliche Anpassung an Bitmap-Kodierung. Fig. 1 zeigt ein Video-Vollbildsignal TV und einen Untertitelbereich SUB, der Text und graphische Elemente G enthält, wobei der Untertitelbereich SUB bitmap-kodiert ist. Die Größe des Untertitelbereichs SUB kann die Video-Vollbild-Abmessungen überschreiten, z.B. sind für das Blu-Ray-Prerecorded Disc (BDP)-Format Untertitel-Bitmaps für eine Dimension erlaubt, die größer als das Video-Vollbild ist. Dann werden vor dem Abspielen Zeilen zugeschnitten, d.h. ein an die entsprechende Vollbilddimension angepasster Teil wird aus der virtuellen Zeile herausgeschnitten und dem Vollbild überlagert angezeigt. In Fig. 1 wird der Untertitelbereich SUB mit der Breite B_{SUB} so zurechtgeschnitten, dass nur ein Teil mit der Breite B_{TV} sichtbar ist. Für Norm-HDTV, das zum Beispiel für BDP verwendet wird, beträgt B_{TV} 1920 Pixel, während B_{SUB} weitaus mehr Pixel enthält.

[0020] Aufgrund der rechteckigen Form des Untertitelbereichs SUB sind die meisten Pixel in diesem Bereich transparent. Dies ist in Fig. 2 in vereinfachter Weise in vergrößertem Maßstab dargestellt, da üblicherweise eine Zeile SL1, SL2 auf einem HDTV-Bildschirm TV mehrere Pixel breit sein muss, um deutlich sichtbar zu sein. Eine Zeile wird hierbei als eine horizontale Struktur verstanden. Jede Zeile von Unterti-

teldaten enthält üblicherweise eine oder mehr Pixelsequenzen mit gleicher Farbe. Fig. 2 zeigt einen Teil einer Untertitelzeile SL1, die transparente Sequenzen PS1, PS5, aber auch einzelne sichtbare Pixel PS4, kürzere sichtbare Zeilen PS2 und längere sichtbare Zeilen PS3 enthält. Die meisten Pixel innerhalb einer Zeile sind transparent. Dies ist der Fall zwischen Zeichen, aber auch am Beginn und am Ende von Untertitelzeilen. Da in jedem Fall die Zeilen mit transparenten Abschnitten beginnen und enden, enthält jede Zeile einen transparenten Abschnitt mehr als farbige Abschnitte. Aber transparente Abschnitte PS1, PS5 sind üblicherweise länger, weil für von transparent abweichende Pixelsequenzen, die z.B. für Zeichen verwendet werden, der häufigste Fall eine Sequenzlänge von 64 oder weniger ist. Dies kann durch eine grobe Abschätzung erkannt werden, wobei angenommen wird, dass wenigstens 25 Zeichen gleichzeitig angezeigt werden, und dass der Raum zwischen Zeichen etwa 1/4 der Breite eines Zeichens hat, so dass ein einzelnes Zeichen nicht mehr als 1920/25·(8/10)=62 Pixel innerhalb einer Zeile benutzen kann. Oft enthält eine Zeile SL2 nur sehr wenige sichtbare Pixel und daher nur wenige transparente Sequenzen, die sehr lang sind.

[0021] Ein Kode, der ein bevorzugtes Ausführungsbeispiel der Erfindung bildet, ist in Fig. 3 aufgelistet. Es ist ein Lauflängen-Kode, der Kodeworte mit einer Länge umfasst, die zwischen 1 Byte bis hinauf zu 4 Bytes mit 8 Bits pro Byte liegen. Er kann 256 verschiedene Farben mit einer bevorzugten Farbe kodieren. Die bevorzugte Farbe ist bei diesem Beispiel ‚transparent', aber sie kann jede andere angemessene Farbe sein. Eine Farb-Nachschlagtabelle (CLUT) kann die dekodierten Farbwerte in die aktuelle Anzeigefarbe transformieren. Ferner können Pixelsequenzen mit gleicher Farbe in zwei Bereichen kodiert werden, wobei der kürzere Bereich bis hinauf zu 63 Pixel und der längere Bereich bis hinauf zu 16.383 Pixel beträgt.

[0022] Die kürzesten Kodeworte mit einer Länge von 1 Byte werden zum Kodieren eines einzelnen Pixels benutzt, das eine individuelle, von der bevorzugten Farbe abweichende Farbe aufweist, die hier transparent ist. Der Farbwert CCCCCCCC kann zwischen 1 bis hinauf zu 255 betragen und kann eine Farbe direkt oder indirekt darstellen. Z.B. kann er einen Eintrag in einer Farb-Nachschlagtabelle (CLUT) darstellen, der den aktuellen Farb-Kode enthält. Einer der 8-Bit-Werte, der nur Nullen (00000000) enthält, dient als Escape-Sequenz, wodurch angezeigt wird, dass die folgenden Bits als Teil desselben Kodewortes zu betrachten sind. In diesem Fall hat der Kodewort-Baum vier mögliche Zweige, die durch die zwei folgenden Bits markiert sind.

[0023] Im ersten Zweig, der durch die folgenden Bits 00 angezeigt wird, haben gültige Kodeworte zwei Bytes, und eine kürzere Sequenz bis hinauf zu 63 Pixeln wird mit der bevorzugten Farbe kodiert, z.B. transparent. Das einzig ungültige Kodewort in diesem Zweig ist dasjenige, das nur Nullen umfasst, da 0 keine gültige Sequenzlänge darstellt. Dieses Kodewort ‚00000000 00000000' kann für andere Zwecke verwendet werden. Gemäß der Erfindung wird es verwendet, um das Ende einer Zeile anzuzeigen, da es das kürzeste redundante Kodewort ist.

[0024] In dem zweiten Zweig, der durch die folgenden Bits 01_b angezeigt wird, umfasst das Kodewort ein anderes Byte, und die vierzehn L-Bits dienen zur Kodierung der Länge einer Pixelsequenz mit der bevorzugten Farbe, z.B. transparent. Somit kann die Sequenzlänge bis hinauf zu $2^{14} – 1 = 16.383$ sein. Die Kodeworte, bei denen die L-Bits einen Wert unter 64 haben, sind redundant und können für andere Zwecke verwendet werden.

[0025] In dem dritten Zweig, der durch die folgenden Bits 10_b angezeigt wird, umfassen die Kodeworte ein zusätzliches Byte, und die sechs L-Bits des zweiten Bytes stellen die Länge einer kürzeren Sequenz bis hinauf zu 63 Pixel dar, die eine von der bevorzugten Farbe abweichende Farbe haben. Die aktuelle Farbe wird direkt oder indirekt durch den CCCCCCCC-Wert des dritten Bytes dargestellt. Die Kodeworte mit einer Sequenzlänge LLLLLL unterhalb drei sind redundant, da eine Sequenz von ein oder zwei Pixeln mit dieser Farbe billiger kodiert werden kann unter Verwendung eines Bytes pro Pixel, wie oben beschrieben, und eine Sequenzlänge von 0 ist ungültig. Diese Kodeworte können für andere Zwecke verwendet werden.

[0026] In dem vierten Zweig, der durch die folgenden Bits 11_b angezeigt wird, umfassen die Kodeworte zwei zusätzliche Bytes, wobei die verbleibenden sechs Bits des zweiten Bytes und das dritte Byte die Länge einer längeren Sequenz von 64 bis hinauf zu 16.383 Pixeln ergeben, und der Farbwert CCCCCCCC des vierten Bytes gibt die Farbe direkt oder indirekt an, die nicht die bevorzugte Farbe ist. Die Kodeworte mit einer Sequenzlänge unter 64 sind redundant, da diese Sequenzen billiger unter Verwendung des dritten Zweiges kodiert werden können. Diese Kodeworte können für andere Zwecke verwendet werden.

[0027] Die oben erwähnten redundanten Kodeworte können zur Erweiterung des Kodes dienen, z.B. zum Hinzufügen interner Prüfsummen oder anderer Informationen.

[0028] Die in Fig. 3 dargestellte und oben beschriebene erweiterte Lauflängen-Kodiertabelle bietet hauptsächlich zwei Vorteile. Erstens erlaubt sie die kompakteste Kodierung von üblichen Untertitelströmen mit transparenten Bereichen, kleinen graphischen Objekten und normalem Untertiteltext. Einzel-

ne Pixel jeder Farbe, die für kleine farbige Graphiken verwendet werden, werden mit einem einzigen Byte kodiert. Die dominante Farbe, z.B. Transparenz für BDP-Untertitelung, wird immer zusammen mit einer Lauflänge kodiert. Lauflängen-Kodes sind in zwei verschiedenen Größen oder zwei Pixelquantitäten verfügbar. Bei einem ersten Schritt sind Lauflängen bis hinauf zu 63 Pixeln als 2-Byte-Kodeworte für die dominante Farbe verfügbar, und als 3-Byte-Kodeworte für die anderen Farben. Bei einem zweiten Schritt sind Lauflängen bis hinauf zu 16.383 Pixeln als 3-Byte-Kodeworte für die dominante Farbe verfügbar, und als 4-Byte-Kodeworte für die anderen Farben. Der Kode für das Ende der Pixelfolge oder der Kode für das Zeilenende ist ein eindeutiges 2-Byte-Kodewort, das für die Resynchronisation verwendet werden kann. Zweitens bedeutet die Verfügbarkeit von längeren Sequenzen für den Untertitelbereich bis hinauf zu 16.383 Pixel pro Kodewort eine Verminderung von Redundanz und daher der Datenmenge. Dies bedeutet, dass bei Anwendungen mit getrennten Datenströmen, die sich einen Kanal teilen, zum Beispiel mehrere Datenströme auf einem optischen Aufzeichnungsträger, die sich denselben Abtaster teilen, größere Teile des Untertitelstroms mit derselben Datenmenge geladen werden können, wodurch die Zugriffsfrequenz für den Untertitelstrom vermindert wird.

[0029] Ein anderer Aspekt der Erfindung ist eine weitere Optimierung des Datenstroms für den Transport unter Verwendung von Transportpaketen, z.B. in einem paketorientierten elementaren Strom (PES). Wegen der großen Dateigröße von Bitmaps ist die Paketisierung solcher Daten, z.B. in Objektdaten-Segmenten (ODS) ein Problem. Oft ist die maximale Größe eines ODS durch andere Faktoren begrenzt, z.B. durch die PES-Paketgröße. Damit große Bitmaps in solche Pakete passen, wäre es erforderlich, Bitmaps in kleine Bitmap-Stücke vor dem Kodieren zu zerschneiden, was die Kompressionseffizienz vermindert. Um diese Bitmap-Aufspaltung zu überwinden, wird ein neues erweitertes Objekt-Datensegment (ExODS) für BDP oder vergleichbare Anwendungen offenbart, was in Fig. 4 gezeigt ist. ExODS ist eine Datenstruktur, die alle Fragmente darstellt, in die ein ODS zerschnitten ist, um in eine Sequenz von Segmenten mit begrenzter Größe und PES-Paketen zu passen. Das vollständige ODS kann durch Verketten der Sequenz aus individuellen Stücken der aufeinanderfolgenden ExODSs rekonstruiert werden.

[0030] Der Beginn und das Ende einer Sequenz von ExODS wird durch getrennte Kennzeichen (flags) angezeigt, „first_in_sequence" und „last_in_sequence". Wenn das Kennzeichen „first_in_sequence" gleich 1 ist, beginnt eine neue Sequenz. Ein ExODS, das das Kennzeichen „first_in_sequence" auf 1 gesetzt hat, zeigt auch die Größe des dekomprimierten Bitmaps dadurch an, dass es seine Dimension, Objektbreite und Objekthöhe enthält. Der Vorteil einer Anzeige der Bitmap-Dimension ist die Unterstützung der Zielspeicher-Zuordnung, bevor die Dekompression beginnt. Ein anderer Vorteil besteht darin, dass die angezeigten Bitmap-Dimensionen auch während der Dekodierung zur Gegenprüfung der Bitmap-Dimensionen verwendet werden können. Wenn das Kennzeichen „last_in_sequence" auf 1 gesetzt wird, wird das letzte ExODS eines vollständigen ODS angezeigt. Es kann ExODS geben, die weder das Flag „first_in_sequence" noch „last_in_sequence" gesetzt haben. Dies sind ExODS-Stücke in der Mitte einer Sequenz. Auch der Fall, dass sowohl das Flag „first_in_sequence" als auch das Flag „last_in_sequence" gesetzt ist, ist möglich, wenn das ODS in einem einzigen ExODS geführt wird. Um diese Begrenzung in der Größe, die für ein einziges ODS verfügbar ist, durch die PES-Paketgröße innerhalb der Untertitelung zu überwinden, kann der beschriebene ExODS-Typ als Behälter (container) für Stücke eines ODS eingeführt werden, z.B. zum Paketisieren von großen ODS für HDTV-Anwendung. Außer den ODS-Stücken führt das ExODS auch Kennzeichen, die anzeigen, ob es das erste Stück, das letzte Stück, ein mittleres Stück oder das eine, aber vollständige Stück einer ExODS-Sequenz führt. Wenn weiterhin das erste Stück in der Sequenz des ExODS übertragen wird, sind die Dimensionen des resultierenden ODS, d.h. Höhe und Breite des kodierten Bitmap, in dem Segment enthalten. Die angezeigten Bitmap-Dimensionen können auch für eine Dekodier-Gegenprüfung verwendet werden.

[0031] Das erfindungsgemäße Verfahren kann zur Kompression von Bitmap-Datenströmen, die zum Beispiel Text, Bilder oder graphische Daten für Animation, Menüs, Navigation, Logos, Reklame, Nachrichten oder sonstiges enthalten, in Anwendungen, zum Beispiel vorbespielten Blu-Ray-(BDP)-Discs, oder allgemein in Aufzeichnungen oder Sendungen von hochauflösenden Videosignalen (HDTV) verwendet werden.

Patentansprüche

1. Verfahren zur Lauflängenkodierung eines Datenstroms, wobei der Datenstrom Bitmap-formatierte Untertiteloder Menüdaten für Video-Darstellung auf einer Anzeige umfasst, wobei die Untertitel- oder Menüdaten Graphiken (G) oder Text (TEXT) oder beides enthalten, und wobei Kodeworte eine Anzeige ihrer Kodewort-Länge enthalten, umfassend die Schritte:
– Definieren einer bevorzugten Farbe (Farbe 0);
– Kodieren von Pixeln mit der bevorzugten Farbe unter Verwendung erster Kodeworte mit zwei oder drei Bytes, wobei die ersten Kodeworte einen Lauflängenwert umfassen, und wobei der in den ersten Kodeworten, die drei Bytes haben, enthaltene Lauflängenwert, die Breite der Anzeige (B_{TV}) überschreiten

kann;
– Kodieren von Pixeln mit einer anderen als der bevorzugten Farbe unter Verwendung zweiter Kodeworte mit einem, drei oder vier Bytes, wobei die zweiten Kodeworte einen Farbwert (C...C) umfassen, und wobei die zweiten Kodeworte, die drei oder vier Bytes haben, einen Lauflängenwert umfassen, und wobei der in den zweiten Kodeworten, die vier Bytes haben, enthaltene Lauflängenwert die Breite der Anzeige (B_{TV}) überschreiten kann.

2. Verfahren nach Anspruch 1, bei dem Kodeworte mit drei Bytes innerhalb der Anzeige ihrer Kodewort-Länge eine Anzeige enthalten, die anzeigt, ob sie zu den ersten oder den zweiten Kodeworten gehören.

3. Verfahren nach Anspruch 1 oder 2, bei dem die Farbwerte und die bevorzugte Farbe mit einer Nachschlagetabelle auf die Anzeigefarben gemappt werden.

4. Verfahren nach Anspruch 1, 2 oder 3, bei dem das kürzeste redundante Kodewort zur Zeilensynchronisation verwendet wird.

5. Verfahren zur Lauflängendekodierung eines kodierten Datenstroms für eine Video-Darstellung auf einer Anzeige, umfassend die Schritte:
– Bestimmen des ersten Bytes eines Kodewortes;
– wenn das erste Byte nicht einen definierten ersten Wert (00000000) hat, Dekodieren des ersten Bytes zu einem einzelnen Pixel mit einer individuellen Farbe, die durch den Wert (C...C) des ersten Bytes definiert ist, wobei die Farbe anders als eine definierte erste Farbe (Farbe 0) ist;
– wenn das erste Byte den definierten ersten Wert (00000000) hat, Bestimmen des ersten und zweiten Bits des folgenden Bytes, das das zweite Byte ist;
– wenn das erste und zweite Bit des zweiten Bytes einen ersten Wert (00) haben, Dekodieren der restlichen Bits des zweiten Bytes zu einer Sequenz von Pixeln mit der definierten ersten Farbe, wobei die verbleibenden Bits des zweiten Bytes die Sequenzlänge bestimmen;
– wenn das erste und zweite Bit des zweiten Bytes einen zweiten Wert (01) haben, Dekodieren der verbleibenden Bits des zweiten Bytes zusammen mit dem folgenden dritten Byte zu einer Sequenz von Pixeln mit der definierten ersten Farbe, wobei die verbleibenden Bits des zweiten Bytes und das dritte Byte die Sequenzlänge definieren, und wobei die Sequenzlänge die Anzeigebreite (B_{TV}) überschreiten kann;
– wenn das erste und zweite Bit des zweiten Bytes einen dritten Wert (10) haben, Dekodieren der verbleibenden Bits des zweiten Bytes zusammen mit dem dritten Byte zu einer Sequenz von Pixeln, wobei die verbleibenden Bits des zweiten Bytes die Sequenzlänge definieren und das dritte Byte die Pixelfarbe definiert; und
– wenn das erste und zweite Bit des zweiten Bytes einen vierten Wert (11) haben, Dekodieren der verbleibenden Bits des zweiten Bytes zusammen mit dem dritten und einem folgenden vierten Byte, wobei die verbleibenden Bits des zweiten Bytes und des dritten Bytes die Sequenzlänge definieren und das vierte Byte die Pixelfarbe definiert, und wobei die Sequenzlänge die Anzeigebreite (B_{TV}) überschreiten kann.

6. Verfahren nach dem vorhergehenden Anspruch, bei dem das Definieren einer Pixelfarbe aus dem ersten, dritten oder vierten Byte und aus dem ersten Wert des ersten Bytes die Verwendung einer Nachschlagetabelle umfasst.

7. Verfahren nach Anspruch 5 oder 6, bei dem der kodierte Datenstrom für eine Video-Darstellung eine getrennte Schicht ist, die andere Video-Daten auf der Anzeige überlagert, und das ferner den Schritt umfasst, einen Teil der getrennten Schicht für die Anzeige auszuwählen.

8. Vorrichtung zur Lauflängenkodierung eines Datenstroms, der Bitmap-formatierte Untertitel- oder Menüdaten für eine sichtbare Darstellung auf einer Anzeige umfasst, wobei die Untertitel- oder Menüdaten Graphiken (G) oder Text (TEXT) oder beides enthalten, und wobei Kodeworte eine Anzeige ihrer Kodewort-Länge enthalten, umfassend:
– Mittel zum Definieren einer ersten Farbe (Farbe 0);
– Mittel zum Kodieren von Pixeln der ersten Farbe unter Verwendung erster Kodeworte mit zwei oder drei Bytes, wobei die ersten Kodeworte einen Lauflängenwert umfassen, und wobei der in den ersten Kodeworten, die drei Bytes haben, enthaltene Lauflängenwert die Breite der Anzeige (B_{TV}) überschreiten kann;
– Mittel zum Kodieren von Pixeln einer anderen als der ersten Farbe unter Verwendung zweiter Kodeworte mit einem, drei oder vier Bytes, wobei die zweiten Kodeworte einen Farbwert (C...C) umfassen, und wobei die zweiten Kodeworte, die drei oder vier Bytes haben, einen Lauflängenwert umfassen, und wobei der in den zweiten Kodeworten, die vier Bytes haben, enthaltene Lauflängenwert die Breite der Anzeige (B_{TV}) überschreiten kann.

9. Vorrichtung zur Lauflängendekodierung eines kodierten Datenstroms, der komprimierte Bitmap-formatierte Untertitel- oder Menüdaten für Video-Anwendung enthält, umfassend:
– Mittel zum Bestimmen der Kodewort-Länge, wobei das erste Byte eines Kodewortes ausgewertet wird, und wobei, wenn das erste Byte einen anderen als einen definierten ersten Wert (00000000) hat, die Kodewort-Länge dann als die eines Bytes bestimmt wird und sonst das erste und zweite Bit des folgenden zweiten Bytes ausgewertet und in Abhängigkeit von diesem ersten und zweiten Bit die Kodewort-Länge

als die von zwei, drei bzw. vier Bytes bestimmt wird;
– Mittel zum Dekodieren von Kodeworten, die bestimmt sind, eine Länge von einem Byte zu haben, zu einzelnen Pixeln mit einer Farbe, die durch das eine Byte definiert ist, wobei die Farbe sich von einer definierten ersten Farbe (Farbe 0) unterscheidet;
– Mittel zum Dekodieren von Kodeworten, die bestimmt sind, eine Länge von zwei Bytes zu haben, zu Sequenzen von Pixeln mit der definierten ersten Farbe, wobei die Sequenzlänge durch die verbleibenden Bits des zweiten Bytes des Kodewortes definiert ist;
– Mittel zum Dekodieren von Kodeworten, die bestimmt sind, eine Länge von drei Bytes zu haben, entweder zu Sequenzen von Pixeln mit der definierten ersten Farbe, wobei die Sequenzlänge die Breite der Video-Anzeige (B_{TV}) überschreiten kann und durch das dritte Byte und die verbleibenden Bits des zweiten Bytes definiert ist, oder zu Sequenzen von Pixeln mit gleicher Farbe abweichend von der definierten ersten Farbe, wobei die Sequenzlänge durch die verbleibenden Bits des zweiten Bytes definiert ist; und
– Mittel zum Dekodieren von Kodeworten, die bestimmt sind, eine Länge von vier Bytes zu haben, zu Sequenzen von Pixeln mit gleicher Farbe abweichend von der definierten ersten Farbe, wobei die Sequenzlänge die Breite der Video-Anzeige überschreiten kann.

10. Vorrichtung nach dem vorhergehenden Anspruch, die ferner Mittel zum Dekodieren von Kodeworten umfasst, die zwei Bytes haben und eine Sequenzlänge von null als Sync-Kodewort für Zeilenende einschließen.

11. Vorrichtung nach einem der Ansprüche 8 bis 10, die ferner Nachschlagetabellen-Mittel zum Zuordnen zwischen den Farbwerten, einschließlich der definierten ersten Farbe, und Anzeige-Farben umfasst.

12. Vorrichtung nach einem der Ansprüche 8 bis 11, bei der der kodierte Datenstrom auf mehrere Transportpakete verteilt ist.

13. Optische Disc, die einen lauflängenkodierten Datenstrom enthält, der Bitmap-formatierte Untertitel- oder Menüdaten für eine Video-Darstellung auf einer Anzeige umfasst, wobei die Untertitel- oder Menüdaten Graphiken (G) oder Text (TEXT) oder beides enthalten, und wobei Kodeworte eine Anzeige ihrer Kodewort-Länge umfassen, wobei der lauflängenkodierte Datenstrom umfasst:
– erste Daten, die eine bevorzugte Farbe definieren;
– erste Kodeworte mit zwei oder drei Bytes, die Pixel mit der bevorzugten Farbe kodieren, wobei die ersten Kodeworte einen Lauflängenwert umfassen, und wobei der in denjenigen ersten Kodeworten, die drei Bytes haben, enthaltene Lauflängenwert die Breite der Anzeige (B_{TV}) überschreiten kann;
– zweite Kodeworte mit einem, drei oder vier Bytes, die Pixel mit einer anderen als der bevorzugten Farbe kodieren, wobei die zweiten Kodeworte einen Farbwert (C...C) umfassen, und wobei diejenigen zweiten Kodeworte, die drei oder vier Bytes haben, einen Lauflängenwert enthalten, und wobei der Lauflängenwert, der in den zweiten Kodeworten mit vier Bytes enthalten ist, die Breite der Anzeige (B_{TV}) überschreiten kann.

Es folgen 2 Blatt Zeichnungen

Anhängende Zeichnungen

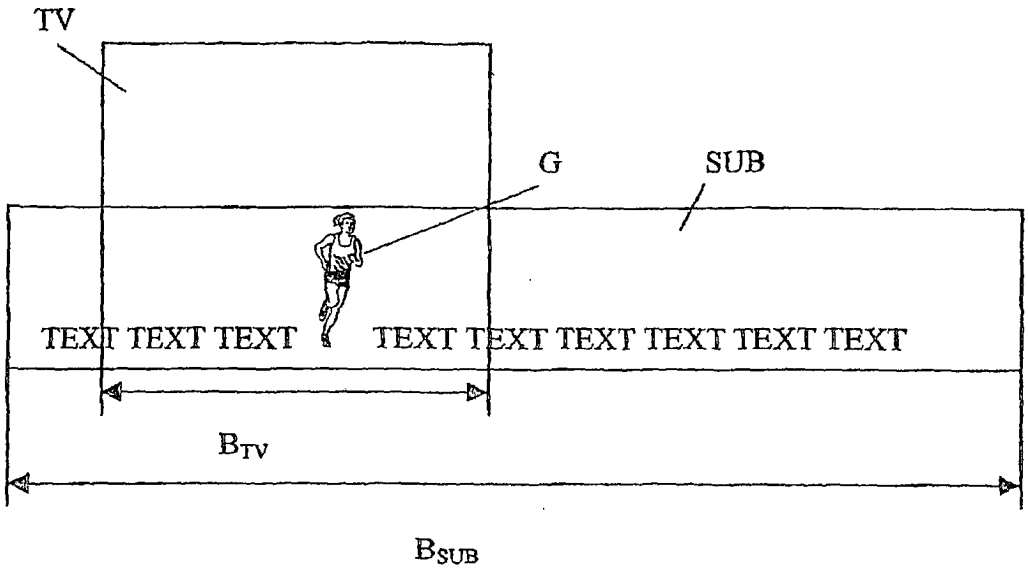

Fig.1

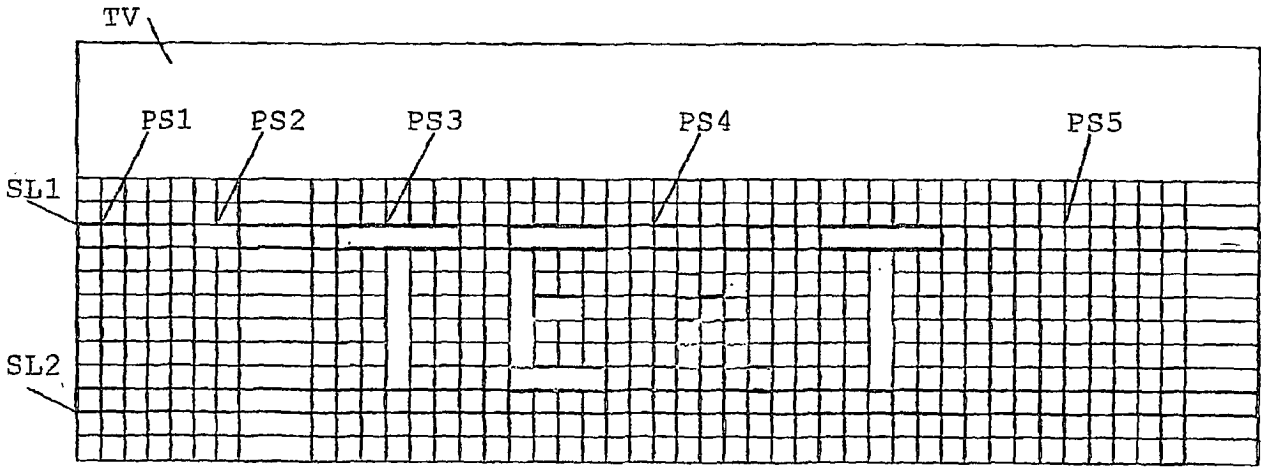

Fig.2

Code	Bedeutung
CCCCCCCC	ein Pixel in Farbe C (1 ≤ C ≤ 255)
00000000 00LLLLLL	L Pixel in Farbe 0 (1 ≤ L ≤ 63)
00000000 01LLLLLL LLLLLLLL	L Pixel in Farbe 0 (64 ≤ L ≤ 16383)
00000000 10LLLLLL CCCCCCCC	L Pixel in Farbe C (3 ≤ L ≤ 63, 1 ≤ C ≤ 255)
00000000 11LLLLLL LLLLLLLL CCCCCCCC	L Pixel in Farbe C (64 ≤ L ≤ 16383, 1 ≤ C ≤ 255)
00000000 00000000	Zeilenende
Mögliche Erweiterungen	
00000000 10000000 X 00000000 10000001 X 00000000 10000010 X 00000000 01000000 00000000 X 00000000 01000000 ... 00000000 01000000 00011111 X	

Fig.3

Syntax	Größe [bit]	Typ
erweitertes_Objekt-Daten-Segment () {		
Sync_Bite	8	bslbf
Segment_Typ	8	bslbf
Seiten_id	16	bslbf
Segment_Länge	16	uimsbf
erstes in der Sequenz	1	bslbf
letztes in der Sequenz	1	bslbf
reserviert	6	bslbf
wenn (erstes in der Sequenz=='1') {		
Objekt_Breite	16	uimsbf
Objekt_Höhe	16	uimsbf
}		
Objekt_Daten_Stück ()		
}		

Fig.4