

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4429643号
(P4429643)

(45) 発行日 平成22年3月10日 (2010. 3. 10)

(24) 登録日 平成21年12月25日 (2009. 12. 25)

(51) Int. Cl.

F I

H O 4 N 5/92 (2006. 01)

H O 4 N 5/92 H

H O 4 N 5/76 (2006. 01)

H O 4 N 5/76 A

H O 4 N 5/85 (2006. 01)

H O 4 N 5/85 A

請求項の数 75 (全 40 頁)

(21) 出願番号 特願2003-183599 (P2003-183599)
 (22) 出願日 平成15年6月26日 (2003. 6. 26)
 (65) 公開番号 特開2004-40791 (P2004-40791A)
 (43) 公開日 平成16年2月5日 (2004. 2. 5)
 審査請求日 平成18年5月16日 (2006. 5. 16)
 (31) 優先権主張番号 10/185, 800
 (32) 優先日 平成14年6月28日 (2002. 6. 28)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドモンド ワン マイ
 クロソフト ウェイ
 (74) 代理人 100140109
 弁理士 小野 新次郎
 (74) 代理人 100089705
 弁理士 社本 一夫
 (74) 代理人 100075270
 弁理士 小林 泰
 (74) 代理人 100080137
 弁理士 千葉 昭男
 (74) 代理人 100096013
 弁理士 富田 博行

最終頁に続く

(54) 【発明の名称】 デジタルデータレートおよび方向つき再生変更を処理する方法およびシステム

(57) 【特許請求の範囲】

【請求項 1】

デジタルデータの逆方向再生指示を受け取り、

前記指示の受け取りにตอบสนองして、逆方向再生の対象である前記デジタルデータの1つまたは複数のデータブロックが下流コンポーネントに渡されたかどうかを判定することであって、

前記1つまたは複数のデータブロックが下流コンポーネントに渡されていないと判定される場合、少なくとも部分的に逆方向再生順序を定義するように前記1つまたは複数のデータブロックを配列し、

前記配列されたデータブロックを下流コンポーネントに送り、

前記1つまたは複数のデータブロックが下流コンポーネントに渡されたと判定される場合、前記1つまたは複数のデータブロックに関して逆方向再生を実施させることを試みるように前記指示を下流コンポーネントに渡すこと

を含むことを特徴とする方法。

【請求項 2】

前記指示を受け取る動作が、前記逆方向再生の対象である前記1つまたは複数のデータブロックを読み取るように構成されたソースコンポーネントによって実行されることを特徴とする請求項1に記載の方法。

【請求項 3】

前記配列する動作は、1つまたは複数のデータブロックを、前記データブロックを含む

10

20

媒体から逆の順序で読み取ることを含むことを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記配列する動作は、1 つまたは複数のフラグを 1 つまたは複数のデータブロックに関連付けることを含み、前記 1 つまたは複数のフラグは、少なくとも 1 つの下流コンポーネントに、前記データブロックが逆方向再生の対象であることを示すことを特徴とする請求項 1 に記載の方法。

【請求項 5】

1 つまたは複数のフラグを使用して前記データブロックの副部分の順序を変更し、逆方向再生をもたらすことをさらに含むことを特徴とする請求項 4 に記載の方法。

【請求項 6】

前記デジタルデータは、ビデオデータを含むことを特徴とする請求項 1 に記載の方法。

【請求項 7】

前記データブロックは、キーフレームを含む副部分を有することを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記 1 つまたは複数のデータブロックを逆方向再生順序でレンダリングすることをさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 9】

コンピュータ実行可能命令をその上に有する 1 つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1 つまたは複数のプロセッサによって実行される際に、1 つまたは複数のプロセッサに、

デジタルデータの逆方向再生指示を受け取り、

前記指示の受け取りにตอบสนองして、逆方向再生の対象である前記デジタルデータの 1 つまたは複数のデータブロックが下流コンポーネントに渡されたかどうかを判定することであって、

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されていないと判定される場合、前記 1 つまたは複数のデータブロックを配列して少なくとも部分的に逆方向再生順序を定義し、

前記配列されたデータブロックを下流コンポーネントに送り、

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されたと判定される場合、前記 1 つまたは複数のデータブロックに関して逆方向再生を実施させることを試みるように前記指示を下流コンポーネントに渡すこと

を行わせることを特徴とするコンピュータ可読媒体。

【請求項 10】

前記命令は、1 つまたは複数のデータブロックを、前記データブロックを含む媒体から逆の順序で読み込むことによって、1 つまたは複数のプロセッサに前記データブロックを配列させることを特徴とする請求項 9 に記載のコンピュータ可読媒体。

【請求項 11】

前記命令は、1 つまたは複数のプロセッサに、1 つまたは複数のフラグを 1 つまたは複数のデータブロックに関連付けさせ、前記 1 つまたは複数のフラグは、少なくとも 1 つの下流コンポーネントに、前記データブロックが逆方向再生の対象であることを示すことを特徴とする請求項 9 に記載のコンピュータ可読媒体。

【請求項 12】

前記命令は、1 つまたは複数のプロセッサに、

1 つまたは複数のフラグを 1 つまたは複数のデータブロックに関連付けて、前記 1 つまたは複数のフラグは、少なくとも 1 つの下流コンポーネントに、前記データブロックが逆方向再生の対象であることを示し、

前記 1 つまたは複数のフラグを使用して前記データブロックの副部分の順序を変更し逆方向再生もたらすこと

10

20

30

40

50

を行わせることを特徴とする請求項 9 に記載のコンピュータ可読媒体。

【請求項 13】

ディジタルデータの逆方向再生指示を受け取り、

前記指示の受け取りにตอบสนองして、逆方向再生の対象である前記ディジタルデータの 1 つまたは複数のデータブロックが下流コンポーネントに渡されたかどうかを判定することであって、

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されていないと判定される場合、

前記 1 つまたは複数のデータブロックを、前記データブロックを含む媒体から逆の順序で読み込んで少なくとも部分的に逆方向再生順序を定義し、

1 つまたは複数のフラグを前記 1 つまたは複数のデータブロックに関連付けて、前記 1 つまたは複数のフラグは、少なくとも 1 つの下流コンポーネントに、前記データブロックが逆方向再生の対象であることを示し、

前記フラグを関連付けられた前記データブロックを下流コンポーネントに送り、

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されたと判定される場合、前記 1 つまたは複数のデータブロックに関して逆方向再生を実施させることを試みるように前記指示を下流コンポーネントに渡すこと

を含むことを特徴とする方法。

【請求項 14】

前記 1 つまたは複数のフラグを関連付けた後に、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードすることをさらに含むことを特徴とする請求項 13 に記載の方法。

【請求項 15】

前記 1 つまたは複数のフラグを関連付けた後に、

1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、

1 つまたは複数のフラグを使用して前記データブロックの副部分を逆の順序で下流コンポーネントに送ること

をさらに含むことを特徴とする請求項 13 に記載の方法。

【請求項 16】

前記 1 つまたは複数のフラグを関連付けた後に、

1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、

1 つまたは複数のフラグを使用して前記データブロックの副部分を逆の順序で下流コンポーネントに送ること

をさらに含み、前記データブロックの副部分は、レンダリングに関する前記副部分のスケジューリングおよび前記副部分のタイムスタンプの順序のさらなる操作に使用することができる 1 つまたは複数の関連する前記タイムスタンプを有する

ことを特徴とする請求項 13 に記載の方法。

【請求項 17】

前記 1 つまたは複数のフラグを関連付けた後に、

1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、

前記データブロックの副部分を逆の順序で下流コンポーネントに送るために 1 つまたは複数のフラグを使用すること

をさらに含み、前記データブロックの副部分は、レンダリングに関する前記副部分のスケジューリングおよび前記副部分のタイムスタンプの順序のさらなる操作に使用することができる 1 つまたは複数の関連する前記タイムスタンプを有し、

前記操作することは、タイムスタンプ値を操作することによってレート変更を実施することをさらに含む

ことを特徴とする請求項 13 に記載の方法。

【請求項 18】

前記 1 つまたは複数のフラグを関連付けた後に、

10

20

30

40

50

1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、
1 つまたは複数のフラグを使用して個々のデータブロックの副部分を逆の順序で下流コンポーネントに送ること
をさらに含み、前記データブロックの副部分は、レンダリングに関する前記副部分のスケジューリングおよび前記副部分のタイムスタンプの順序のさらなる操作に使用することができる 1 つまたは複数の関連する前記タイムスタンプを有し、
前記操作することは、前記送られたデータブロックの副部分の前記タイムスタンプの有効な順序を逆転することを含む
ことを特徴とする請求項 13 に記載の方法。

【請求項 19】

10

前記 1 つまたは複数のフラグを関連付けた後に、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、
1 つまたは複数のフラグを使用して前記データブロックの副部分の順序を逆転すること
をさらに含むことを特徴とする請求項 13 に記載の方法。

【請求項 20】

前記 1 つまたは複数のフラグを関連付けた後に、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、
デコードを用いて、1 つまたは複数のフラグを使用して前記データブロックの副部分の順序を逆転すること
をさらに含むことを特徴とする請求項 13 に記載の方法。

20

【請求項 21】

前記デジタルデータは、ビデオデータを含むことを特徴とする請求項 13 に記載の方法。

【請求項 22】

前記データブロックは、キーフレームを含む副部分を有することを特徴とする請求項 13 に記載の方法。

【請求項 23】

前記 1 つまたは複数のデータブロックを逆方向再生順序でレンダリングすることさらに含むことを特徴とする請求項 13 に記載の方法。

【請求項 24】

30

コンピュータ実行可能命令をその上に有する 1 つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1 つまたは複数のプロセッサによって実行される際に、1 つまたは複数のプロセッサに、請求項 13 に記載の方法を実施させることを特徴とする 1 つまたは複数のコンピュータ可読媒体。

【請求項 25】

ソースコンポーネントを用いて、デジタルデータの逆方向再生指示を受け取り、
ソースコンポーネントによって下流コンポーネントに照会して、どの下流コンポーネントが逆方向再生を備える機能をサポートするかを確かめ、

前記指示の受け取りに応答して、前記ソースコンポーネントによって、逆方向再生を指示された前記デジタルデータの 1 つまたは複数のデータブロックが前記ソースコンポーネントから下流コンポーネントに渡されたかを判定することであって、

40

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されていないと判定される場合、

前記ソースコンポーネントを用いて、前記 1 つまたは複数のデータブロックを、前記データブロックを含む媒体から逆の順序で読み込んで、少なくとも部分的に逆方向再生順序を定義し、

前記ソースコンポーネントを用いて、1 つまたは複数のフラグを 1 つまたは複数のデータブロックに関連付けて、前記 1 つまたは複数のフラグは、少なくとも 1 つの下流コンポーネントに、前記データブロックが逆方向再生の対象であることを示し、

前記フラグを関連付けられたデータブロックを下流コンポーネントに送り、

50

前記 1 つまたは複数のデータブロックが下流コンポーネントに渡されたと判定される場合、

前記ソースコンポーネントによって、逆方向再生機能をサポートする下流コンポーネントが少なくとも 1 つあるかを判定し、ある場合、前記ソースコンポーネントによって、前記 1 つまたは複数のデータブロックに関して逆方向再生を実施させることを試みるように前記指示を下流コンポーネントに渡し、前記逆方向再生は、逆方向再生をもたらす形で前記 1 つまたは複数のデータブロックの各々に関連するタイムスタンプを再割り当てすることにより下流コンポーネントによって実施されることを含むことを特徴とする方法。

【請求項 26】

10

前記 1 つまたは複数のフラグを関連付けた後に、デコーダを用いて、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードすることをさらに含むことを特徴とする請求項 25 に記載の方法。

【請求項 27】

前記 1 つまたは複数のフラグを関連付けた後に、デコーダを用いて、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、

1 つまたは複数のフラグを使用して前記データブロックの副部分を逆の順序で下流コンポーネントに送ること

をさらに含むことを特徴とする請求項 25 に記載の方法。

20

【請求項 28】

前記 1 つまたは複数のフラグを関連付けた後に、デコーダを用いて、1 つまたは複数のデータブロックを個々のデータブロックの副部分にデコードし、

1 つまたは複数のフラグを使用して、前記データブロックの副部分であって、レンダリングに関する前記副部分のスケジューリングおよび前記副部分のタイムスタンプの順序のさらなる操作に使用することができる 1 つまたは複数の関連するタイムスタンプを有する前記データブロックの副部分を逆の順序で下流コンポーネントに送ること

をさらに含むことを特徴とする請求項 25 に記載の方法。

【請求項 29】

30

前記デジタルデータは、ビデオデータを含むことを特徴とする請求項 25 に記載の方法。

【請求項 30】

前記データブロックは、キーフレームを含む副部分を有することを特徴とする請求項 25 に記載の方法。

【請求項 31】

レンダラを用いて、前記 1 つまたは複数のデータブロックを逆方向再生順にレンダリングすることをさらに含むことを特徴とする請求項 25 に記載の方法。

【請求項 32】

コンピュータ実行可能命令をその上に有する 1 つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1 つまたは複数のプロセッサによって実行される際に、1 つまたは複数のプロセッサに、請求項 25 に記載の方法を実施させることを特徴とする 1 つまたは複数のコンピュータ可読媒体。

40

【請求項 33】

デジタルデータの逆方向再生指示を受け取るように構成された第 1 コンポーネントであって、前記指示の受け取りに応答して、逆方向再生の対象である前記デジタルデータの 1 つまたは複数のデータブロックが前記第 1 コンポーネントの下流の第 2 コンポーネントに渡されたかどうかを判定し、

前記 1 つまたは複数のデータブロックが前記第 2 コンポーネントに渡されていないと判定される場合、

50

前記 1 つまたは複数のデータブロックを、前記データブロックを含む媒体から逆の順序で読み込むように構成されて、少なくとも部分的に逆方向再生順序を定義する、第 1 コンポーネント

を含み、前記第 1 コンポーネントが、さらに、1 つまたは複数のフラグを 1 つまたは複数のデータブロックに関連付けるように構成され、前記 1 つまたは複数のフラグは、前記データブロックが逆方向再生の対象であることを少なくとも 1 つの下流コンポーネントに示し、

1 つまたは複数のフラグを付けられたデータブロックを受け取り、下流コンポーネントに逆の順序で前記データブロックの副部分を送るよう構成される、第 2 コンポーネントを含み、

10

前記 1 つまたは複数のデータブロックが前記第 2 コンポーネントに渡されていないと判定される場合、前記 1 つまたは複数のデータブロックに関して逆方向再生を実施させることを試みるように前記指示を前記第 1 コンポーネントから前記第 2 コンポーネントに渡すことを特徴とするシステム。

【請求項 3 4】

前記データブロックの副部分は、前記副部分をスケジューリングしてレンダリングするのに使用する 1 つまたは複数の関連するタイムスタンプを有し、前記第 2 コンポーネントが、さらに、前記データブロックの副部分の前記タイムスタンプの順序を操作するように構成されることを特徴とする請求項 3 3 に記載のシステム。

【請求項 3 5】

20

前記第 2 コンポーネントは、デコードコンポーネントを含むことを特徴とする請求項 3 3 に記載のシステム。

【請求項 3 6】

前記第 2 コンポーネントは、レンダラコンポーネントを含むことを特徴とする請求項 3 3 に記載のシステム。

【請求項 3 7】

前記第 1 コンポーネントおよび前記第 2 コンポーネントは、フィルタグラフの少なくとも一部を含むことを特徴とする請求項 3 3 に記載のシステム。

【請求項 3 8】

前記データブロックは、キーフレームを含む副部分を有することを特徴とする請求項 3 3 に記載のシステム。

30

【請求項 3 9】

デジタルデータの逆方向再生の表示を受け取り、

媒体から前記デジタルデータの 1 つまたは複数のデータブロックを読み込み、1 つまたは複数のデータブロックを配列して少なくとも部分的に逆方向再生順序を定義し、

すべてのデータブロックにまたがって単調増加する形で、前記逆方向再生の対象であるデータブロックに加えて、これと同様に、タイムスタンプをデータブロックに関連付け、前記関連付ける動作は、前記データブロックが再生されるレートの変更をもたらす形で前記タイムスタンプを関連付けることを含み、

レート変更の対象であるデータブロックが下流コンポーネントに渡されたかどうかを判定し、渡された場合、レート変更を実施させることを試みるようにレート変更要求を下流コンポーネントに渡すこと

40

を含むことを特徴とする方法。

【請求項 4 0】

前記指示を受け取る動作が、前記逆方向再生指示の対象であるデータを読み取るように構成されたソースコンポーネントによって実行されることを特徴とする請求項 3 9 に記載の方法。

【請求項 4 1】

前記配列する動作は、1 つまたは複数のデータブロックを、前記データブロックを含む媒体から逆の順序で読み込むことを含むことを特徴とする請求項 3 9 に記載の方法。

50

【請求項 4 2】

前記配列する動作は、1つまたは複数のフラグを1つまたは複数のデータブロックに関連付けることを含み、前記1つまたは複数のフラグは、前記データブロックが逆方向再生の対象であることを少なくとも1つの下流コンポーネントに示すことを特徴とする請求項39に記載の方法。

【請求項 4 3】

前記配列する動作は、1つまたは複数のフラグを1つまたは複数のデータブロックに関連付けることを含み、1つまたは複数のフラグは、前記データブロックが逆方向再生の対象であることを少なくとも1つの下流コンポーネントに示し、前記1つまたは複数のフラグを使用して前記データブロックの副部分の順序を変更し逆方向再生をもたらすことをさらに含むことを特徴とする請求項39に記載の方法。

10

【請求項 4 4】

前記デジタルデータは、ビデオデータを含むことを特徴とする請求項39に記載の方法。

【請求項 4 5】

前記データブロックは、キーフレームを含む副部分を有することを特徴とする請求項39に記載の方法。

【請求項 4 6】

前記関連付けられたタイムスタンプに従ってデータブロックをレンダリングすることをさらに含むことを特徴とする請求項39に記載の方法。

20

【請求項 4 7】

コンピュータ実行可能命令をその上に有する1つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1つまたは複数のプロセッサによって実行される際に、1つまたは複数のプロセッサに、請求項39に記載の方法を実施させることを特徴とする1つまたは複数のコンピュータ可読媒体。

【請求項 4 8】

ソースコンポーネントで、デジタルデータを再生する再生レートの変更に関するレート変更要求を受け取り、

前記レート変更要求に関連するデータが、前記ソースコンポーネントから下流コンポーネントに供給されたかどうかを判定し、

30

前記データが下流コンポーネントに供給されていない場合に、少なくとも前記データがレート変更要求の対象であることを示すのに有効に前記データを処理し、

前記データが下流コンポーネントに供給されている場合に、下流コンポーネントに前記レート変更要求を渡して前記レート変更を実施させることを試みることを

含むことを特徴とする方法。

【請求項 4 9】

下流コンポーネントを用いて前記レート変更要求を受け取り、

前記レート変更要求に関連するデータが別の下流コンポーネントに供給されたかどうかを判定し、

前記データが別の下流コンポーネントに供給されていない場合に、少なくとも前記データがレート変更要求の対象であることを示すのに有効に前記データを処理し、

40

前記データが異なる下流コンポーネントに供給されている場合に、前記異なる下流コンポーネントに前記レート変更要求を渡して前記レート変更を実施させることを試み、前記データが既にレンダリングされている場合に、前記レート変更要求を捨てること

をさらに含むことを特徴とする請求項48に記載の方法。

【請求項 5 0】

最初の下流コンポーネントは、デマルチプレクサを含むことを特徴とする請求項48に記載の方法。

【請求項 5 1】

最初の下流コンポーネントは、デコーダを含むことを特徴とする請求項48に記載の方

50

法。

【請求項 5 2】

前記下流コンポーネントの 1 つは、レンダラを含むことを特徴とする請求項 4 8 に記載の方法。

【請求項 5 3】

前記下流コンポーネントの 1 つは、ビデオレンダラを含むことを特徴とする請求項 4 8 に記載の方法。

【請求項 5 4】

前記下流コンポーネントの 1 つは、ビデオデコーダを含むことを特徴とする請求項 4 8 に記載の方法。

10

【請求項 5 5】

ソースコンポーネントであって、
デジタルデータが再生される再生レートの変更に係るレート変更要求を受け取り、
前記レート変更要求に関連するデータが前記ソースコンポーネントから第 1 下流コンポーネントに供給されたかどうかを判定する
ように構成され、

前記データが前記第 1 下流コンポーネントに供給されていない場合に、前記ソースコンポーネントが、少なくとも前記データがレート変更要求の対象であることを示すのに有効に前記データを処理するように構成され、

前記データが前記第 1 下流コンポーネントに供給されている場合に、前記ソースコンポーネントが、前記レート変更要求を前記第 1 下流コンポーネントに渡すように構成されて前記レート変更を実施させることを試みる、

20

ソースコンポーネントと、
第 1 下流コンポーネントであって、
前記レート変更要求を受け取り、前記レート変更要求に関連するデータが第 2 下流コンポーネントに供給されたかどうかを判定する
ように構成され、

前記データが前記第 2 下流コンポーネントに供給されていない場合に、前記第 1 下流コンポーネントが、少なくとも前記データがレート変更要求の対象であることを示すのに有効に前記データを処理するように構成され、

30

前記データが前記第 2 下流コンポーネントに供給されている場合に、前記第 1 下流コンポーネントが、前記レート変更要求を前記第 2 下流コンポーネントに渡すように構成されて前記レート変更を実施させることを試みる、

第 1 下流コンポーネントと
を含むことを特徴とするシステム。

【請求項 5 6】

前記下流コンポーネントの 1 つは、デコーダを含むことを特徴とする請求項 5 5 に記載のシステム。

【請求項 5 7】

前記下流コンポーネントの 1 つは、レンダラを含むことを特徴とする請求項 5 5 に記載のシステム。

40

【請求項 5 8】

前記下流コンポーネントの 1 つは、ビデオレンダラを含むことを特徴とする請求項 5 5 に記載のシステム。

【請求項 5 9】

データ処理パイプラインの一部を含むレンダラによってレンダリングされる 1 つまたは複数のデータサンプルに情報を付加することであって、前記情報は、もしあればデータサンプルがその対象であるレート変更と、データサンプルをレンダリングすべき時に関連するタイムスタンプと、前記データサンプルが順方向または逆方向のどちらでレンダリングされるかとを定義し、

50

前記 1 つまたは複数のデータサンプルおよびそれらに関連する情報を前記パイプラインに沿って伝搬させ、

データサンプルおよびそれらに関連する情報を、前記データサンプルを媒体から読み込むソースコンポーネントの下流の 1 つまたは複数のコンポーネントで受け取り、

前記 1 つまたは複数の下流コンポーネントで、前記情報に従ってレート変更およびレンダリング方向の一方または両方を実施するのに十分な 1 つまたは複数のデータサンプルを処理し、前記データサンプルに関連するタイムスタンプ値を少なくともとも操作すること

を含むことを特徴とする方法。

【請求項 6 0】

前記処理する動作は、データサンプル順序を逆転して逆方向再生をもたらすことを含むことを特徴とする請求項 5 9 に記載の方法。

10

【請求項 6 1】

前記処理する動作は、データサンプル順序を逆転して逆方向再生をもたらすことを含み、

タイムスタンプ値を操作する前記動作およびデータサンプル順序を逆転する前記動作の少なくとも 1 つが、前記レンダラによって実行される

ことを特徴とする請求項 5 9 に記載の方法。

【請求項 6 2】

コンピュータ実行可能命令をその上に有する 1 つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1 つまたは複数のプロセッサによって実行される際に、1 つまたは複数のプロセッサに、請求項 5 9 に記載の方法を実施させることを特徴とする 1 つまたは複数のコンピュータ可読媒体。

20

【請求項 6 3】

ソースコンポーネントで、特定のソース時間を有するデータサンプルに関するレート変更要求を受け取り、

前記レート変更要求の受け取りに応答して、前記ソースコンポーネントに関連する 1 つまたは複数のバッファを調べて前記レート変更が行われる対応するデータサンプルが存在するかどうかを確かめ、

前記データサンプルが、前記 1 つまたは複数のバッファで見つかる場合に、レート変更情報を前記データサンプルに付加し、

30

前記データサンプルが、前記ソースコンポーネントのバッファで見つからない場合に、前記レート変更要求を次の順番の下流コンポーネントに送ること

を含むことを特徴とする方法。

【請求項 6 4】

前記付加する動作は、前記データサンプルがまだ前記バッファ内にある間に、前記レート変更情報を前記データサンプルに付加することを含むことを特徴とする請求項 6 3 に記載の方法。

【請求項 6 5】

前記付加する動作は、前記レート変更情報をキャッシングし、前記データサンプルが前記ソースコンポーネントによって処理される時に前記レート変更情報を前記データサンプルに付加することを含むことを特徴とする請求項 6 3 に記載の方法。

40

【請求項 6 6】

前記次の順番の下流コンポーネントを用いて前記レート変更要求を受け取り、

前記レート変更要求の受け取りに応答して、前記下流コンポーネントに関連する 1 つまたは複数のバッファを調べて前記レート変更が行われる対応するデータサンプルが存在するかどうかを確かめ、

前記データサンプルが、前記下流コンポーネントのバッファで見つかる場合に、レート変更情報を前記データサンプルに付加し、

前記データサンプルが、前記下流コンポーネントのバッファで見つからない場合に、前記レート変更要求を次の順番の下流コンポーネントに送ること

50

をさらに含むことを特徴とする請求項 6 3 に記載の方法。

【請求項 6 7】

前記データサンプルが、前記要求を受け取るコンポーネントのすべてのバッファで見つからない場合に、前記レート変更要求がレンダラによって受け取られるまで、前記レート変更要求を下流コンポーネントに送り続けることと、前記レート変更要求が前記レンダラによって受け取られ、前記データサンプルが前記レンダラによってレンダリングされている場合に、前記レート変更要求を破棄することとをさらに含むことを特徴とする請求項 6 6 に記載の方法。

【請求項 6 8】

前記ソースコンポーネントから下流のコンポーネントで前記レート変更要求を実施することをさらに含むことを特徴とする請求項 6 3 に記載の方法。

【請求項 6 9】

前記ソースコンポーネントから下流のレンダラで前記レート変更要求を実施することをさらに含むことを特徴とする請求項 6 3 に記載の方法。

【請求項 7 0】

コンピュータ実行可能命令をその上に有する 1 つまたは複数のコンピュータ可読媒体であって、前記コンピュータ実行可能命令は、1 つまたは複数のプロセッサによって実行される際に、1 つまたは複数のプロセッサに、請求項 6 3 に記載の方法を実施させることを特徴とする 1 つまたは複数のコンピュータ可読媒体。

【請求項 7 1】

レンダリングされるデータサンプルに関連するタイムスタンプ情報を生成し、管理するソースコンポーネントであって、

順方向再生および逆方向再生の対象であるデータサンプルに関する単調増加するタイムスタンプを生成し、

逆方向再生をもたらすためにデータサンプルを含むデータブロックの順序を逆転し、

逆方向で再生される個々のデータブロックにタグを付け、

レート変更をスケジューリングするために、下流コンポーネントに照会することによって前記レート変更をスケジューリングできるようにする

ように構成される、ソースコンポーネントと、

デコードコンポーネントであって、

レート変更の受け取りに応答してデータサンプルに対する補間を実行し、

下流コンポーネントにレート変更を渡し、

逆方向再生データブロックを、処理のためにより小さい逆方向ブロックに分解し、

データサンプルの出力タイムスタンプを修正してレート変更の実施を支援する

ように構成されたデコードコンポーネントと、

レンダラコンポーネントであって、

レート変更に関連するデータサンプルの有効プレゼンテーションタイムスタンプを計算し、

逆方向再生を意図されたデータサンプルを受け取り、

データサンプルをキャッシングして前記レンダラコンポーネントが前記データサンプルを逆の順序でレンダリングできるようにする

ように構成されたレンダラコンポーネントと

を含むことを特徴とするシステム。

【請求項 7 2】

コンポーネントは、逆方向再生データブロックを副ブロックに分解することができ、その後、適当に変更されたタイムスタンプと共に逆の順序で前記副ブロックを送ることができることを特徴とする請求項 7 1 に記載のシステム。

【請求項 7 3】

前記レンダラによって使用することができ、入力タイムスタンプを出力タイムスタンプにマッピングして、レート変更を実施するタイムスタンプマネージャをさらに含むことを

10

20

30

40

50

特徴とする請求項 7 1 に記載のシステム。

【請求項 7 4】

レンダリングされるデータサンプルに関連するタイムスタンプ情報を生成し、管理するソースコンポーネントであって、

順方向再生および逆方向再生の対象であるデータサンプルに関する単調増加するタイムスタンプを生成し、

データサンプルを含むデータブロックの順序を逆転して逆方向再生をもたらし、

逆方向で再生される各データブロックにタグを付け、

レート変更をスケジューリングするために、下流コンポーネントを照会することによって前記レート変更をスケジューリングできるようにする

ように構成される、ソースコンポーネントと、

データサンプルをデコードするように構成されたデコーダコンポーネントと、

データサンプルをレンダリングするように構成されたレンダラコンポーネントと、

レート変更を実施するために、次に使用可能なタイムスタンプと、前記タイムスタンプが発生するまでに使用可能な時間の長さとに関して前記ソースによって照会することができるタイムスタンプマネージャと

を含むことを特徴とするシステム。

【請求項 7 5】

前記タイムスタンプマネージャに関連するレジストリをさらに含み、前記レンダラが、前記タイムスタンプマネージャに登録し、データサンプルをレンダリングする時に前記レジストリを更新するように構成されることを特徴とする請求項 7 4 に記載のシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ビデオデータ、オーディオ/ビデオデータ、および類似物などのレンダリング可能なデジタルデータを処理する方法およびシステムに関する。具体的に言うと、本発明は、再生速度変更または再生レート変更および方向変更（すなわち、順方向データ再生および逆方向データ再生）を処理する方法およびシステムに関する。

【0002】

【従来の技術】

プロセッサ、アクセラレーションハードウェア、およびストレージデバイスの性能が向上を続けるので、通常（すなわち 1.0）を超える再生速度が可能である。しかし、バッファリングおよびストリーム位置合わせの問題によって、アプリケーションとユーザが知覚する実際の再生速度の変化の間の対話性の度合いが制限される。モノリシックな閉じた環境でこの問題を解くことが可能であるが、この場合に、解決策のすべてのコンポーネントが絡み合う。しかし、オープンでコンポーネント化された解決策の場合には、これははるかに困難な問題である。この問題を解決する複数の試みが行われたが、そのそれぞれが、その有用性を制限する基礎的な欠点および手落ちを有する。

【0003】

再生レート（再生速度とも称する）によって、データの各フレームが表示される時間の長さが決定される。高い再生レートでは、通常は、低い再生レートより短い時間の期間だけフレームが表示される。高い再生レートは、ほとんどのプロセッサストレージ検索機能およびハードウェア機能を超える可能性がある高い帯域幅要件を有する。通常、高い再生レートは、いわゆる「スキャンモード」を使用して近似され、スキャンモードでは、ストリームのデータの一部を破棄することによって、データストリームの（小さい）部分だけが選択的に提示される。これは、すばやく進行するスライドショーに多少似ている。

【0004】

コンピュータで実行されるものまたは対話型テレビジョンセットに関して実行されるものなどの多くのビデオアプリケーションが、ソース（またはソースフィルタ）を制御するユーザインターフェースからなる。ソース（またはソースフィルタ）は、最終的にユーザの

10

20

30

40

50

ためにデータをレンダリングできるようにデータを処理するデータ処理パイプラインの一部である。ソースは、メディアファイルを読み取り、通常は、データサンプルまたはバッファ（通常はたとえばMPEGを使用して圧縮される）を、処理のために他のタイプのデコードに渡す。デコードは、データを圧縮解除し、ユーザのためにデータをレンダリングするように構成され、それを行うことができるなんらかのタイプのレンダラに渡す。レンダラは、通常は、内部（または外部）クロックおよびデータサンプル自体に含まれるさまざまなタイミング情報を使用して、サンプルを正しい時刻に提示またはレンダリングする。レンダラが処理を開始する時に、初期レンダリングクロック時刻が、ソースおよびデコードに渡される。ソースは、初期レンダラ時刻の多少後に開始されるタイムスタンプを有するサンプルを作り始めることができる。タイムスタンプは、プレゼンテーションの作成された時刻に基づいてさまざまなデータサンプルをスケジューリングし、レンダリングするために、レンダラによって使用される。データ処理パイプラインの各ステージの間でサンプルがバッファリングされるので、パイプラインおよび/または処理コンポーネント（フィルタなど）の間の小さい遅延が発生する可能性がある。（グラフまたは）パイプラインの待ち時間は、ソース（フィルタ）からのサンプルの、それが提示されるかレンダリングされる時刻までの累積的伝搬遅延である。システムが、ビデオコンテンツなどのデータを、異なる再生レートで（順方向と逆方向の両方で）スムーズに再生できるようにすることが、開発者の目標である。しかし、データ処理パイプラインの性質およびさまざまなデータフォーマットによって、開発者に課題が提示され続ける。

10

【0005】

20

たとえば、異なるデータフォーマットすなわち、MPEG-2フォーマット、DVDフォーマット、およびHDTVフォーマットを検討されたい。

【0006】

MPEG-2

MPEG-2フォーマットは、「順方向デコーディング（forward decoding）」フォーマットと称するフォーマットである。MPEG-2フォーマットの例の表現を、図1で、全般的に10に示す。各ビデオシーケンスは、一連のGroup of Pictures（または「GOPs」）からなる。GOPは、ピクチャまたはフレームのシーケンスからなる。フレームは、3つのタイプすなわち、イントラフレーム（Iフレーム）、順方向予測フレーム（Pフレーム）、および両方向予測フレーム（Bフレーム）にエンコードすることができる。

30

【0007】

Iフレームまたは「キーフレーム」（Iフレーム12）などは、過去または未来のフレームを参照せずに、単一のイメージとしてエンコードされる。使用されるエンコード方式は、JPEG圧縮に似る。Pフレーム（Pフレーム18など）は、過去の基準フレームに関してエンコードされる。Pフレームは、基準フレームに対する変更が含まれるという点で、「デルタフレーム」とみなすこともできる。基準フレームは、PフレームまたはIフレームである。過去の基準フレームは、最も近い前の基準フレームである。Bフレーム（または両方向フレーム、フレーム14および16など）は、過去の基準フレーム、将来の基準フレーム、または両方のフレームに関してエンコードされる。将来の基準フレームは、最も近い次の基準フレーム（IまたはP）である。Bフレームは、隣接する基準フレームだけの関数である。

40

【0008】

GOP構造は、シーケンスへのランダムアクセスを支援することを意図されたものである。GOPは、通常は、Iフレームから始まる限り任意のサイズとすることができる、独立のデコード可能な単位である。

【0009】

MPEG-2フォーマットに関連する問題の1つが、逆方向でデータを再生できるようにすることである。順方向でのデータの再生は、フォーマット自体が順方向デコーディングであり、これは、まずIフレームをデコードし、その後GOP内の他のフレームに移らな

50

ければならないことを意味するので、通常は問題でない。しかし、逆方向でのデータの再生は、GOPを逆方向にデコードできないので、多少はより困難である。

【0010】

DVD

通常は、イメージが、DVDなどのディスクに記録される時に、ビデオは、実際には所定の時間期間（通常は1/2秒単位またはビデオオブジェクトベーシックユニット（video object basic units）（VOBU）を含む小さい単位に分解される。このフォーマットの利点は、ビデオを再生する時に、ビデオユニットを1つずつ進むことができることである。ビデオの任意の部分にジャンプしたい場合には、当のビデオユニットに単純にジャンプすることができ、オーディオとビデオが同期化される。すべてのストリームが同期化される位置を、「クリーンポイント（clean point）」と称する。したがって、ビデオユニットおよびオーディオユニットが圧縮される時に、これらは、正確に同時にレンダリングされるユニットに圧縮される、すなわち、オーディオとビデオの間のスキューがない。

10

【0011】

MPEG2の文脈で説明する時のIフレームへのすべての言及を、他のデータフォーマットでキーフレームに拡張することができる。用語Iフレームは、MPEG2の文脈の外で説明する時に、キーフレームと同義である。

【0012】

HDTV: ATSC (American Television Standards Committee) および DVB (欧州フォーマット)

20

高品位テレビジョンまたはHDTVでは、MPEG-2フォーマットも使用される。しかし、ビデオブロックとオーディオブロックが、わずかなスキューを伴って位置合せされる。この場合に、ストリーム内のある点に単純に早送りまたはジャンプすることはできない。というのは、その点にビデオサンプルがある可能性があるが、関連するオーディオサンプルが、ストリーム内の別の位置から開始されるからである。さらに、オーディオサンプルは、ブロックとして順方向にデコードすることだけができる。これは、ストリーム内でバックアップし、関連するオーディオサンプルを探さなければならないことを意味する。特定のフォーマットに依存して、対応するオーディオブロックまたはオーディオサンプルの先頭がどこにあるかを実際に知ることはできない。したがって、当のビデオサンプルおよびオーディオサンプルの両方の前のある点についてストリーム内で後を見続ける必要がある。

30

【0013】

オープンでコンポーネント化された解決策について異なる再生レートおよび再正方向を使用可能にすることを試みる時に、これらの異なるタイプのフォーマットに伴って課題が生じる。

【0014】

DVDからデータをレンダリングすることができる例示的なシステム200を示す図2を検討されたい。システム200には、DVD206からデータを読み取るソースコンポーネント204と通信するアプリケーション202が含まれる。DVDから読み取られるデータには、エンコードされ、一緒に多重化されたオーディオデータおよびビデオデータが含まれる。ソースは、DVDからデータを読み取る時に、データパケットにタイムスタンプを適用し、このタイムスタンプは、レンダリングのためにパケットを同期化し、スケジューリングするのに使用される。パケットは、デマルチプレクサ208に供給され、デマルチプレクサ208は、パケットを異なる構成部分すなわち、オーディオパケット、ビデオパケット、および存在する場合にサブピクチャ（sub picture）パケットに分割する。パケットは、その後、デマルチプレクサによって、ビデオデコーダ210（ビデオパケットのデコード用）、オーディオデコーダ212（オーディオパケットのデコード用）、およびサブピクチャデコーダ214（サブピクチャパケットのデコード用）などの関連するデコーダに供給される。パケットのそれぞれが、関連するタイミング情報を有し

40

50

、このタイミング情報によって、パケットがレンダリングされると思われる時が定義される。さまざまなデコーダは、関連するパケットを圧縮解除し、個々のデータサンプルまたはデータパケット（パケットのタイムスタンプを含む）を、ビデオレンダラ 216 およびオーディオレンダラ 218 などの適当なレンダラに送る。

【0015】

システム 200 には、通常は、グローバルクロック 220 も含まれ、グローバルクロック 220 は、そのタイムスタンプがグローバルクロックによって示される時刻と一致するデータサンプルをレンダリングする時を確認するために、さまざまなレンダラによって使用される。

【0016】

ユーザが、アプリケーション 202 を介して、異なる、おそらくはより速いレートでデータサンプルをレンダリングさせることを望むことを示すと仮定する。

【0017】

順方向レート変更を調整する過去の手法は、グローバルクロック 220 を操作することである。すなわち、データを通常のレートの 2 倍の速度で再生したい場合には、グローバルクロックの速度を操作することによって、望みのレート変更を実施することができる。この手法に関する問題は、オーディオレンダラが、周波数シフトおよびひずんだオーディオ出力（ユーザの経験の質が下がる）に関連する問題を経験する可能性があることである。さらに、ビデオレンダラが、クロック変更に従うことを試みる時に、ビデオレンダラの処理が遅れ、これによって、レンダラが、追い付こうとしてサンプルを捨てる結果になる可能性がある。この全体的な結果は、オーディオの周波数シフトと、ビデオのタグアンドプル（tug-and-pull）である。サブピクチャ成分（ビデオレンダラに送られるデータがそれから生じる可能性がある）も、グローバルクロック変更に関連する問題を有する可能性があり、したがって、たとえばサブピクチャが不適切な時刻にまたは不適切なビデオに関連してレンダリングされることになる可能性がある。したがって、出力の品質が、大きく劣化する可能性がある。

【0018】

順方向レート変更に対処することを試みるもう 1 つの手法は、ソース 204 に、デマルチプレクサ 208 に通知させ、デマルチプレクサ 208 に、ビデオデコーダ 210 に通知させて、適当なレート変更を行うことである。デコーダ 210 は、サンプルのタイムスタンプに対するスケーリング動作を行って、異なるレートでビデオ再生を行うことができる。この手法に関する問題は、ビデオデコーダ 210、オーディオデコーダ 212、およびサブピクチャデコーダ 214 が、同一の技法およびアルゴリズムを使用してサンプルを処理することの保証がないことであり、これは、異なるデコーダが異なるベンダ製である場合に特にそうである。したがって、レート変更が、わずかに異なる速度で行われる可能性があり、これによって、ビデオとオーディオがずれ始めることになる可能性がある。さらに悪いことに、サブピクチャが、同期化されなくなる可能性があり、これによって、誤った時にサブピクチャが表示される可能性がある。

【0019】

さらに、この 2 つの手法は、順方向再生されるビデオに関してのみ実際に使用され、逆方向再生されるビデオには使用されなかった。これらの過去の手法を使用すると、実際には、ビデオを逆方向に再生するようにビデオレンダラに伝える方法がなかった（かつ、現在もない）。ビデオレンダラは、通常は、ビデオをディスクから読み取る方法を制御できず、それに関する知識を有しない。

【0020】

【発明が解決しようとする課題】

本発明の目的は、レンダリング可能なデジタルデータを処理する改善された方法およびシステムを提供することである。

【0021】

【課題を解決するための手段】

10

20

30

40

50

さまざまな方法およびシステムによって、ビデオデータ、オーディオ/ビデオデータ、オーディオ/ビデオ/サブピクチャデータ、および類似物などのデジタルデータを、順方向と逆方向の両方での再生を可能にする形で処理できるようにする。さまざまな実施形態は、ユーザの経験を強化できる形での再生レート変更の処理にも対処する。

【0022】

【発明の実施の形態】

概要

下で説明するさまざまな方法およびシステムによって、ビデオデータ、オーディオ/ビデオデータ、およびオーディオ/ビデオ/サブピクチャデータなどのデジタルデータを、順方向と逆方向の両方でのスムーズな再生を可能にする形で処理できるようになる。さらに、再生レート変更を、ユーザの経験を強化する形で処理することができる。

10

【0023】

例示的なコンピューティングシステム

図3に、下で説明するシステムおよび関連する方法を実施することができる適切なコンピューティング環境300の例を示す。

【0024】

コンピューティング環境300は、適切なコンピューティング環境の1つの例にすぎず、メディア処理システムの使用の範囲または機能性に関する制限を暗示することを意図されたものではない。また、コンピューティング環境300を、例示的なコンピューティング環境300に示された任意のコンポーネントまたはその組合せに対する依存性または要件を有するものと解釈してもならない。

20

【0025】

説明されるさまざまな実施形態は、多数の他の汎用または特殊目的のコンピューティングシステム環境またはコンピューティングシステム構成と共に動作することができる。メディア処理システムと共に使用するのに適する可能性がある周知のコンピューティングシステム、コンピューティング環境、および/またはコンピューティング構成の例に、パーソナルコンピュータ、サーバコンピュータ、シンクライアント、シッククライアント、ハンドヘルドデバイス、ラップトップデバイス、マルチプロセッサシステム、マイクロプロセッサベースのシステム、セットトップボックス、プログラマブル民生用電子機器、ネットワークPC、ミニコンピュータ、メインフレームコンピュータ、上記のシステムまたはデバイスのいずれかを含む分散コンピューティング環境、および類似物が含まれるが、これに制限はされない。

30

【0026】

いくつかの実施形態で、このシステムおよび関連する方法を、コンピュータによって実行されるプログラムモジュールなどのコンピュータ実行可能命令の全般的な文脈で説明することができる。一般に、プログラムモジュールには、特定のタスクを実行するか特定の抽象データ型を実施する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。これらの実施形態は、通信ネットワークを介してリンクされたりリモート処理デバイスによってタスクが実行される分散コンピューティング環境でも実践することができる。分散コンピューティング環境では、プログラムモジュールを、メモリストレージデバイスを含むローカルおよびリモートの両方のコンピュータ記憶媒体に配置することができる。

40

【0027】

図3に示された例の実施形態によれば、コンピューティングシステム300は、1つまたは複数のプロセッサまたは処理ユニット302、システムメモリ304、およびシステムメモリ304を含むさまざまなシステムコンポーネントをプロセッサ302に結合するバス306を含むものとして図示されている。

【0028】

バス306は、メモリバスまたはメモリコントローラ、周辺バス、accelerated graphics port、およびさまざまなバスアーキテクチャのいずれかを使

50

用するプロセッサバスまたはローカルバスを含む、複数のタイプのバス構造のいずれかの1つまたは複数を表すことを意図されている。制限ではなく例として、そのようなアーキテクチャに、Industry Standard Architecture (ISA) バス、マイクロチャネルアーキテクチャ (MCA) バス、Enhanced ISA (EISA) バス、Video Electronics Standards Association (VESA) ローカルバス、およびメザニンバスとも称するPeripheral Component Interconnect (PCI) バスが含まれる。

【0029】

コンピュータ300には、通常はさまざまなコンピュータ可読媒体が含まれる。そのような媒体は、コンピュータ300によってローカルにおよび/またはリモートからアクセス可能なすべての使用可能媒体とすることができ、これには、揮発性および不揮発性の両方の媒体、取外し可能および取外し不能の両方の媒体が含まれる。

10

【0030】

図3では、システムメモリ304に、ランダムアクセスメモリ (RAM) 310などの揮発性の形のコンピュータ可読媒体および/または読取専用メモリ (ROM) 308などの不揮発性メモリの形のコンピュータ可読媒体が含まれる。起動中などにコンピュータ300内の要素の間で情報を転送するのを助ける基本ルーチンを含む基本入出力システム (BIOS) 312が、通常はROM 308に保管される。RAM 310には、通常は、処理ユニット302から即座にアクセス可能かつ/または処理ユニット302によって現在操作されるデータおよび/またはプログラムモジュールが含まれる。

20

【0031】

コンピュータ300に、さらに、他の取外し可能/取外し不能の揮発性/不揮発性のコンピュータ記憶媒体を含めることができる。例としてのみ、図3に、取外し不能不揮発性磁気媒体 (図示せず、通常は「ハードドライブ」と称する) から読み取り、それに書き込むハードディスクドライブ328、取外し可能不揮発性磁気ディスク332 (たとえば、「フロッピー (登録商標) ディスク」) から読み取り、それに書き込む磁気ディスクドライブ330、およびCD-ROM、DVD-ROM、または他の光学媒体などの取外し可能不揮発性光ディスク336から読み取り、それに書き込む光ディスクドライブ334を示す。ハードディスクドライブ328、磁気ディスクドライブ330、および光ディスクドライブ334は、それぞれ、1つまたは複数のインターフェース326によってバス306に接続される。

30

【0032】

ドライブおよびそれに関連するコンピュータ可読媒体は、コンピュータ300の、コンピュータ可読命令、データ構造、プログラムモジュール、および他のデータのストレージを提供する。本明細書で説明する例示的实施形態では、ハードディスク328、取外し可能磁気ディスク332、および取外し可能光ディスク336が使用されるが、当業者は、磁気カセット、フラッシュメモリカード、デジタルビデオディスク、ランダムアクセスメモリ (RAM)、読取専用メモリ (ROM)、および類似物など、コンピュータによってアクセス可能なデータを保管することができる他のタイプのコンピュータ可読媒体も、例示的オペレーティング環境で使用できることを諒解されたい。

40

【0033】

制限ではなく例としてオペレーティングシステム314、1つまたは複数のアプリケーションプログラム316 (たとえば、マルチメディアアプリケーションプログラム324)、他のプログラムモジュール318、およびプログラムデータ320を含む複数のプログラムモジュールを、ハードディスク328、磁気ディスク332、光ディスク336、ROM 308、またはRAM 310に保管することができる。ユーザは、キーボード338およびポインティングデバイス340 (「マウス」など) などの入力デバイスを介してコンピュータ300にコマンドおよび情報を入力することができる。他の入力デバイスに、オーディオ/ビデオ入力デバイス353、マイクロホン、ジョイスティック、ゲームパッド、衛星放送用パラボラアンテナ、シリアルポート、スキャナ、または類似物 (図示せず

50

）を含めることができる。これらおよび他の入力デバイスは、バス 306 に結合される入力インターフェース 342 を介して処理ユニット 302 に接続されるが、パラレルポート、ゲームポート、または universal serial bus (USB) などの他のインターフェースおよびバス構造によって接続することができる。

【0034】

モニタ 356 または他のタイプのディスプレイデバイスも、ビデオアダプタまたはビデオ/グラフィックスカード 344 などのインターフェースを介してバス 306 に接続される。モニタのほかに、パーソナルコンピュータには、通常は、スピーカおよびプリンタなどの他の周辺出力デバイス（図示せず）が含まれ、これらの周辺出力デバイスを、出力周辺インターフェース 346 を介して接続することができる。

10

【0035】

コンピュータ 300 は、リモートコンピュータ 350 などの 1 つまたは複数のリモートコンピュータへの論理接続を使用して、ネットワーク化された環境で動作することができる。リモートコンピュータ 350 には、通常は、コンピュータに関して本明細書で説明する要素および特徴の多数またはすべてを含めることができる。

【0036】

図 3 からわかるように、コンピューティングシステム 300 は、ローカルエリアネットワーク (LAN) 351 および一般的な広域ネットワーク (WAN) 352 を介して、リモートデバイス（たとえば、リモートコンピュータ 350）に通信的に結合される。そのようなネットワーキング環境は、オフィス、企業全体のコンピュータネットワーク、イントラネット、およびインターネットでありふれたものである。

20

【0037】

LAN ネットワーキング環境で使用される時には、コンピュータ 300 は、ネットワークインターフェースまたはネットワークアダプタ 348 を介して LAN 351 に接続される。WAN ネットワーキング環境で使用される時には、コンピュータ 300 に、通常はモデム 354 または、WAN 352 を介する通信を確立する他の手段が含まれる。モデム 354 は、内蔵または外付けとすることができるが、ユーザ入力インターフェース 342 または他の適当な機構を介してシステムバス 306 に接続することができる。

【0038】

ネットワーク化された環境では、パーソナルコンピュータ 300 に関して図示されたプログラムモジュール、またはその一部を、リモートメモリストレージデバイスに保管することができる。制限ではなく例として、図 3 に、リモートコンピュータ 350 のメモリデバイスに常駐するものとしてリモートアプリケーションプログラム 316 を示す。図示のネットワーク接続が、例示的であり、コンピュータの間で通信リンクを確立する他の手段を使用することができることを諒解されたい。

30

【0039】

第 1 の例示の実施形態 - 逆方向再生/レート変更

図 4 に、あるタイプのメディアで実施されるデジタルデータを処理し、その結果、そのデータを逆方向にレンダリングできるようにする第 1 の実施形態を概略的に示す。この例では、システムに、ユーザが、処理されレンダリングされる特定のデジタルメディアと対話できるインターフェースを提供するアプリケーション 400 が含まれる。このシステムには、レンダリングされるデータを読み取るための 1 つまたは複数のソース 402、レンダリングされるデータをデコードまたは圧縮解除する 1 つまたは複数のデコーダ 406、およびデータをレンダリングする 1 つまたは複数のレンダラ 408 も含まれる（各タイプのコンポーネントは 1 つだけ図示）。レンダラ 408 によってレンダリングされるデジタルメディアは、メディア 410 で提供され、メディア 410 には、この例では、DVD などのディスクが含まれる。しかし、どのタイプのメディアでも、デジタルメディアを保管するのに使用することができる。ソース、デコーダ、およびレンダラが、別々の個々のコンポーネントとして図示されているが、それぞれの機能性をより集積された形で提供できることを諒解されたい。

40

50

【 0 0 4 0 】

メディア 4 1 0 は、通常は、レンダリングすることができる大量のデータを保持する。データは、通常は、論理ユニットに編成される。論理ユニットは、ブロックとして表される。用語「ブロック」は、特定のタイプのデータまたはメディアを指定することを意図されたものではない。そうではなく、この用語は、この実施形態がその下で動作する原理を示すために使用される。したがって、図示されるブロックは、通常はキーフレームから始まるセクションにデータを分割する異なるタイプまたは異なるフォーマットのデータを表すことができる。たとえば、ブロックに、MPEG - 1 フォーマットまたは MPEG - 2 フォーマットでデータをフォーマットするのに使用される GOP を含めることができる。ブロックは、DVD でデータをフォーマットするのに使用されるものなどの VOB U を表すこともできる。

10

【 0 0 4 1 】

そのようなブロックの特性の 1 つが、個々のブロックが、ブロックの副コンポーネントを表す副部分からなることである。したがって、ブロックが GOP を表す場合には、個々の副部分に、1 つの I フレームおよび 1 つまたは複数の P フレームまたは B フレームを含めることができる。各ブロックの副部分は、各ブロック内で数字によって示される。たとえば、ブロック 1 は、副部分 1、2、3、および 4 からなり、これらの副部分のそれぞれが、ビデオフレームを表す。

【 0 0 4 2 】

ブロックは、ブロックとして処理されるために下流コンポーネントに渡される、より小さい独立の副ブロックに分解することもできる。たとえば、DVD では、VOB U に、複数の GOP を含めることができる。VOB U ブロックを、それぞれが 1 つの GOP を含む複数の副ブロックに分解することができる。これは、コンテンツを逆方向に再生する時に有用である。というのは、より大きいブロックを、より小さいブロックに分解でき、これらの小さいブロックの順序が、下流に送られる時に逆転されるからである。新しい副ブロックのそれぞれについて新しいタイムスタンプを計算する必要がある（図 1 4 参照）。

20

【 0 0 4 3 】

ユーザが、アプリケーション 4 0 0 を介して、特定のデータ（ビデオデータなど）を逆方向に再生することを望むことを示す時に、アプリケーション 4 0 0 は、逆方向再生が意図されていることをソース 4 0 2 に通知する。アプリケーションは、通常は、逆方向再生が行われる時間範囲またはソース時間も示すことができる。ソース 4 0 2 が、逆方向再生が行われる時間期間に入った時に、ソースは、相対ソース時間内のデータのブロックを逆に読み取る。

30

【 0 0 4 4 】

たとえば、図 4 を検討されたい。メディア 4 1 0 に、3 つの個々のブロック（ブロック 1、ブロック 2、およびブロック 3 として示されている）がグループ A として含まれることが図示されていることに留意されたい。実際には、メディア 4 1 0 に、図示されたものより多数のブロックが含まれる。しかし、この例では、ユーザが、グループ A のブロックを逆方向に再生することを望むことを、アプリケーション 4 0 0 を介してソース 4 0 2 に示したと仮定する。

40

【 0 0 4 5 】

ソースは、メディア 4 1 0 からデータを読み取る際に、ソース時間を記憶し、ソース時間がブロック 1 から 3 と一致する時に、ブロック 1 から 3 を逆の順序で読み取る。具体的に言うと、ブロック 3 が最初に読み取られ、その後、ブロック 2、その後、ブロック 1 が読み取られる。したがって、ソース 4 0 2 の左で、グループ B を定義するために、読み取られるブロックの順序付けがブロック 3 / ブロック 2 / ブロック 1 であることに留意されたい。その後、ソース 4 0 2 は、1 つまたは複数のフラグをブロックに追加して、それらが逆方向再生を意図されていることを示す。具体的に言うと、スタートフラグが、ブロック 3 に追加され、エンドフラグが、ブロック 1 に追加されることに留意されたい。フラグの任意の適切な配置を使用して、逆方向再生に関してブロックを指定することができる。た

50

例えば、フラグを、各ブロックの終りに追加することができる。ソース402は、ブロックにタイムスタンプを追加することによってブロックを処理することもできる。これは、各ブロックのソース時間を、レンダリングアクティビティをスケジューリングするのにレンダラによって使用される単調増加するタイムスタンプにマッピングすることによって行うことができる。

【0046】

その後、各ブロックが、ソース402によってデコーダ406に供給される。デマルチプレクサまたはスプリッタをソースとデコーダの間に入れることができるが、図を簡単にするために図示されていないことを諒解されたい。したがって、たとえば、ブロック3をデコーダ406に供給し、その後、ブロック2を、最後にブロック1を供給することができる。その代わりに、ソースが、ブロックを事前に解析し、処理またはデコード/圧縮解除のためにデコーダに逆の順序で一時にブロックのより小さい副ブロックを送ることができる。これによって、デコーダに伝搬されるブロックのサイズが減る。

10

【0047】

デコーダ406は、ブロックを受け取る時に、左から右へとデコードする。しかし、ブロックのフラグによって、デコーダに、ブロックをデコードし、レンダラ408に発行されるブロックの副部分(フレーム)の順序を逆転しなければならないことが示される。具体的に言うと、デコーダ406は、ブロック3を受け取り、普通にそのブロックをデコードし、他のブロックを処理している間、デコードされたブロック部分をバッファリングすることができる。デコーダ406は、ブロック2および1を同様に処理する。デコーダが、エンドフラグに達した時に、デコーダは、逆方向再生が行われる特定のブロックシーケンスが定義済みであることを知る。デコーダは、ブロック副部分のタイムスタンプを逆転することができる、そのブロック副部分をレンダラ408に逆の順序で発行することができる。発行されたブロック副部分は、グループCとして表される。

20

【0048】

上記からわかるように、レンダラは、ストリームが順方向再生される場合に有する順序に対して逆転されたデータのストリームを受け取る。レンダラは、レンダリングのためにストリームをスケジューリングすることができる。レンダラは、ここで、ストリームが逆方向で再生されることを知らない。というのは、その詳細が、ソースコンポーネントおよびデコーダコンポーネントの中に閉じこめられているからである。

30

【0049】

ユーザが、再生レートを変更するためにレート変更を実施することを望むと仮定する。この実施形態では、レート変更を下記のように実施することができる。ユーザが、通常すなわち1倍速より高速または低速のレートでデータをレンダリングさせることを望むと仮定する。レート変更を、アプリケーション400を介してソース402に通信することができる。レート変更情報を、ソースからデコーダ406に供給することができ、その結果、レート変更処理を、デコーダによって実行することができる。この実施形態では、デコーダが、それが処理するサンプルまたはブロック副部分のタイムスタンプを操作することによって、レート変更処理を実行することができる。たとえば、サンプルのグループが、1倍速でレンダリングされ、 t_1 から t_4 まで伸びるタイムスタンプを有する場合に、タイムスタンプを $t_1 - t_4$ から $t_1 - t_2$ (ただし、 $t_2 - t_1 = (t_4 - t_1) / 2$)に変更することによって、そのようなサンプルを2倍速でレンダリングすることができる。

40

【0050】

図5は、一実施形態による方法のステップを説明する流れ図である。この方法は、適切なハードウェア、ソフトウェア、ファームウェア、またはその組合せで実施することができる。図示の例では、この方法を、図4に図示されそれに関して説明されたシステムに関連して実施することができる。

【0051】

ステップ500で、逆方向再生が望まれることの指示を受け取る。このステップは、逆方向再生が望まれることの通信または呼出しをソフトウェアから受け取ることによって実施

50

することができる。この通信に、逆方向再生が望まれるソース時間の範囲を含めることができる。ステップ502で、1つまたは複数のデータブロックを配置して、再生順序を定義する。このステップは、指示されたソース時間範囲に対応するブロックを逆方向で読み取ることによって実施することができる。これを行うことができる方法の1つの例を、上で示した。ステップ504で、1つまたは複数のフラグをデータブロックに関連付ける。このフラグは、当の1つまたは複数のブロックが、逆方向再生の対象であることを示すことを意図されたものである。逆方向再生スタートフラグおよび逆方向再生エンドフラグを、当のブロックに関連付けることができる。

【0052】

ステップ506で、ブロックを個々のブロック副部分にデコードし、さらに、所望の逆方向再生に従ってブロック副部分を処理することができる。これには、たとえば、所定の時間期間の間のブロック副部分のバッファリングを含めることができる。たとえば、ブロック副部分を、逆方向再生を意図されたブロック副部分のすべてがデコードされるまでバッファリングすることができる。その代わりにまたはそれに追加して、ブロック副部分を、所定の数または量の副部分がバッファリングされるまでバッファリングすることができる。バッファリングされたデコードされたデータを、圧縮ソースデータと共に使用して、逆方向再生ストリームを完全に再生成することもできる。さらに、ブロック副部分を処理して、それに関連するタイムスタンプを操作することもできる。これには、個々の副部分のタイムスタンプの有効な順序を逆転することを含めることができる。これには、要求される可能性があるレート変更に起因するすべてのタイムスタンプ処理を含めることもできる。そのようなタイムスタンプ処理に、タイムスタンプの値の変更を含めることができる。

【0053】

ステップ508で、逆の順序でブロック副部分を発行する。このステップでは、ステップ504のフラグを使用して、ブロック副部分を逆の順序で発行しなければならないことを示すことができる。したがって、たとえば、デコーダ406(図4)が、ブロック副部分を処理する時に、フラグの存在によって、デコーダがブロック副部分を逆の順序で発行しなければならないことを示すことができる。デコーダは、ブロック副部分を逆の順序で発行することができる。これを行う方法の1つの例を、上で示す。ステップ510で、副部分が発行されたのと逆の順序で、発行されたブロック副部分をレンダリングする。

【0054】

同一のアルゴリズムを適用して、ブロックを副ブロックに分割することができ、この副ブロックは、ソースまたはデコーダなどの上位コンポーネントが使用することができる。効果的に、ソースは、ビデオ全体を副ブロック(DVDの場合にはVOBU)に分割し、それらを、逆転されたタイムスタンプと共に逆の順序で発行する。デコーダは、VOBUを読み取り、キャッシングし、その後、キャッシングされたデータを逆の順序で発行することによって、VOBUをGOPブロックに分割してレンダラに渡すことができる。

【0055】

上で説明した処理およびシステムは、デコードされたブロック副部分をバッファリングするために、より多くのメモリの要件を有する。しかし、この手法の長所に、下記が含まれる。第1に、いくつかのシステムは、効果的に位置において後ろにジャンプし、複数のフレームを順方向に再生し、その後、もう一度位置において後ろにジャンプし、複数のフレームを順方向に再生する処理を繰り返すことによって、逆方向再生の達成を試みる。この処理は、作業が反復されるので計算的に極度に非効率的になる可能性があり、通常はユーザの経験の質が低下する。他の手法では、一時に1データブロックだけが操作される。

【0056】

上のシステムおよび方法では、実際に、逆方向再生が望まれる時に複数のブロックの順序を変更することができる。さらに、フラグを使用することによって、システムが、個々のブロック副部分またはフレームを逆の順序で実際にレンダリングできるようになる。また、ブロック副部分のタイムスタンプを操作することによって、レート変更をスムーズに提供することができる。他の長所は、当業者に明白であろう。

【 0 0 5 7 】

性能の利点として、上で説明した方法およびシステムは、グラフィックスプロセッサ内のさまざまなパイプライン化されたビデオアクセラレーション技法を利用することができる。これは、デコーダについて特にそうである。これによって、データパイプラインが、継続的に効率的に動作できるようになり、初期化スタートアップコストを防ぐことができる。さまざまなパイプライン化されたビデオアクセラレーション技法が、米国特許出願第 0 9 / 8 3 9 6 8 2 号（発明の名称：拡張可能なマルチメディアアプリケーションプログラムインターフェイスおよび関連する方法、出願日：2 0 0 1 年 4 月 2 0 日、その開示内容は参照されることによってこの明細書に組み込まれる）。

【 0 0 5 8 】

第 2 の例示的实施形態 - 逆方向再生 / レート変更

上で説明した実施形態では、デコーダが、アプリケーションからのすべてのレート変更要求ならびにすべての逆方向再生要求を処理するように構成された。しかし、レンダリング可能なデータを処理するシステムのチェーンまたは処理パイプラインに、より多くのコンポーネントが含まれる可能性がある。たとえば、処理パイプラインの 1 タイプを、フィルタグラフ (f i l t e r g r a p h) と称する。フィルタグラフは、フィルタ（または処理ユニット）の有向ネットワーク (d i r e c t e d n e t w o r k) である。フィルタグラフには、最終的にレンダリングされるデータを処理するように構成された多数の異なるタイプのフィルタを含めることができる。ソースフィルタおよびデコーダフィルタのほかに、そのようなフィルタグラフに、変換フィルタ、効果フィルタ、ネットワークリ

【 0 0 5 9 】

再生およびレート変更処理を、システム、パイプライン、またはフィルタグラフにまたがって分散させることが有利になる可能性がある。この理由の 1 つが、実装の柔軟性が高まること、ならびに、オープンで拡張可能なアーキテクチャである。たとえば、ユーザが、既にデコーダによって処理されたが、レート変更を行うことができない下流コンポーネントでキューイングされているサンプルのレート変更を要求する場合に、すべてではないにしてもほとんどの現在のシステムが、通常は、そのレート変更を無視する。

【 0 0 6 0 】

通常、レート変更は、2 つの形の 1 つで行うことができる。第 1 に、ユーザは、レート変更を現在または今、行うことを要求することができる。おそらく、ユーザは、再生レートに影響する、UI 内のスライダを操作する。第 2 に、レート変更を、ソースコンテンツ内の所与の位置でスケジューリングすることができる。たとえば、おそらく、ユーザは、ソース時間 0 0 : 0 0 : 0 5 から開始される 2 倍速へのレート変更が望ましいことを示す。この 2 つのシナリオは、根本的に異なる。第 1 のシナリオは、要求のタイミングおよびその瞬間の処理パイプラインの状態のゆえに、より困難なシナリオである。ソースコンテンツが、既にソースによって読み取られ、パイプラインの次のコンポーネントに転送されている可能性がある。第 2 のシナリオは、レート変更が行われる場所がわかっており、したがって、レート変更を前もってスケジューリングすることができるので、より容易な問題である。

【 0 0 6 1 】

上で注記したように、ソースは、データを処理する時に、特定のメディアからデータを読み取り、データに関連するソース時間を、レンダリングのスケジューリングに使用されるタイムスタンプにマッピングする。たとえば、次の表に、1 倍速での順方向再生についてソースによって実行されるマッピング動作の表現を示す。

【 0 0 6 2 】

【表 1】

10

20

30

40

ソース時間	タイムスタンプ
00:00:01	フレーム1 (0秒)
00:00:02	フレーム2 (1秒)
00:00:03	フレーム3 (2秒)
00:00:04	フレーム4 (3秒)
00:00:05	フレーム5 (4秒)

10

【 0 0 6 3 】

上の表では、00:00:01から00:00:05に対応するソース時間が、レンダリングされる特定のフレームと一致するタイムスタンプにマッピングされる。たとえば、00:00:01は、フレーム1に対応するタイムスタンプにマッピングされ、00:00:02は、フレーム2に対応するタイムスタンプにマッピングされるなどである。したがって、1倍速での順方向再生に関して、1対1マッピング関係がある。

【 0 0 6 4 】

逆方向再生の要求がある時に、マッピング動作が、多少変更される。この例では、ソースで、まだソース時間がタイムスタンプにマッピングされる。しかし、ソースは、タイムスタンプの連続的な単調増加するストリームを提供するだけではなく、逆方向再生要求も考慮に入れた形でそれを行う。たとえば、下の表に、ソース時間範囲00:00:01から00:00:05までの、1倍速での逆方向再生に関するマッピング動作を示す。

20

【 0 0 6 5 】

【表2】

ソース時間	タイムスタンプ
00:00:01	フレーム5 (0秒)
00:00:02	フレーム4 (1秒)
00:00:03	フレーム3 (2秒)
00:00:04	フレーム2 (3秒)
00:00:05	フレーム1 (4秒)

30

【 0 0 6 6 】

ソース時間が、逆方向再生を反映して再割り当てされているが、対応するタイムスタンプが、単調増加し続けることに留意されたい。したがって、レンダラは、このデータサンプルを受け取る時に、逆方向再生に影響するようにレンダリングをスケジューリングする。

40

【 0 0 6 7 】

ソースコンテンツの別の部分へのジャンプがあると仮定する。おそらく、ソースコンテンツは、ソース時間範囲00:00:01から00:00:05までについてレンダリングされ、その後、ソース時間範囲00:00:20から00:00:24までへのジャンプがある。この場合に、マッピング動作によって、下記の単調増加するタイムスタンプのストリームが作られる。

【 0 0 6 8 】

【表3】

ソース時間	タイムスタンプ
00:00:01	フレーム1 (0秒)
00:00:02	フレーム2 (1秒)
00:00:03	フレーム3 (2秒)
00:00:04	フレーム4 (3秒)
00:00:05	フレーム5 (4秒)
00:00:20	フレーム6 (5秒)
00:00:21	フレーム7 (6秒)
00:00:22	フレーム8 (7秒)
00:00:23	フレーム9 (8秒)
00:00:24	フレーム10 (9秒)

10

【 0 0 6 9 】

したがって、この実施形態で、ソースの責任は、アプリケーションから要求を受け取り、ソースデータの位置（すなわちソース時間）に基づくコンテキストを単調増加する出力タイムスタンプにマッピングすることである。それを行う際に、データの連続的なストリームを、レンダリングのためにレンダラに供給することができる。これは、他のコンポーネントが、タイムスタンプおよび特定のプロセスのどこにいるかを記憶する必要がないという点で、他のコンポーネント、特にレンダラの観点から、有利である。そうではなく、他のコンポーネントは、単純に、タイムスタンプの線形ストリームを有するサンプルを受け取り、処理することができる。

20

【 0 0 7 0 】

ソースでの処理を容易にするために、ソースが、レート変更を行った時およびそのレート変更が将来に発生するイベントのリストのどこにマッピングされたかのヒストリを維持することができる。その場合に、アプリケーションが、特定の時でのレート変更の要求を始める場合に、ソースは、過去にレート変更をどのように編成したかを知っている。ソースは、レート変更に関連するソースコンテンツが、既に処理のために下流コンポーネントに渡されたかどうか知っている。すでに渡されている場合に、ソースは、たとえばレート変更要求を下流コンポーネントに渡し、これらのコンポーネントにレート変更実施の関連処理を行わせることによって、レート変更を実施させることを試みるステップを行うことができる。他の手法では、既にソースから出たソースコンテンツに関するそのようなレート変更要求は、単純に削除されていた。

30

【 0 0 7 1 】

コンポーネント能力を確かめるためのパイプラインへの照会

上および下で説明する実施形態のいくつかの有利な特徴の1つが、ソースが、コンポーネントの機能を確認するために処理パイプライン内のコンポーネントに照会する能力を有することである。たとえば、ソースは、デコーダなどのコンポーネントに照会して、そのコンポーネントが逆再生をサポートするかどうか、サポートされる再生レート、最大再生レート、最小再生レート、および類似物を確認することができる。それを行うことによって、ソースは、ソースデータを読み取る方法と、いくつかの場合に、他のコンポーネントに送るソースデータのフォーマットを、知的に判断できるようになる。

40

【 0 0 7 2 】

一例として、下記を検討されたい。レート変更を実施し、それを知的に行うために、ソースが、パイプライン全体またはフィルタグラフによってサポートされる最大レートを知ることが有利になる可能性がある。これは、順方向再生と逆方向再生の両方にあてはまる可

50

能性がある。この実施形態では、1つまたは複数のパイプラインコンポーネントに、ソースが照会することのできるインターフェース（単に呼出し可能なメソッドの集合である）が与えられる。パイプラインを構成するかインスタンス化する時に、ソースは、1つまたは複数のインターフェースに照会して、コンポーネントの機能を確認することができる。たとえば、ソースは、コンポーネントに照会して、それらのどれかが逆方向再生をサポートするかどうかを確認することができる。そうでない場合には、これによって、ソースが、逆方向再生をサポートしない、より単純なモードに入り、それをアプリケーションに示すようにすることができ、これによって、アプリケーションが、よりスマートなUIをユーザに提示するようになる。これには、たとえば、ブロック全体を逆に読み取り、適当なフラグを用いてタグを付けるのではなく（いずれにせよデコーダが操作することはできない）、逆の順序でブロックのキーフレームを読み取ることを含めることができる。さらに、ソースは、サポートされる再生レートを確認するために照会することができる。これを知ることによって、ソースは、その処理を適応させて、パイプラインを圧倒しないことを保証することができる。たとえば、特定の再生速度が、パイプラインによってサポートされない場合に、ソースは、パイプラインによるさらなる処理のために、あるブロック部分だけ（たとえばIフレームだけ）を送ることができる。これは、パイプラインが、効率的に処理できないデータの重荷を過度に負わされないことを保証すると同時に、できる限り要求された再生レートに近いレンダリングされたディスプレイを提供するのを試みるのに役立つ。

10

【0073】

20

照会処理を実施する方法の1つは、次の通りである。ソースは、最初に始動されるかインスタンス化される時に、直接にまたは間接的に、パイプラインのすべてのコンポーネントに、その機能、たとえば最大順方向レートおよび最大逆方向レートを照会する。これを行う方法の1つが、ソースが、最初の下流コンポーネントに照会し、最初の下流コンポーネントが、その最初の下流コンポーネントに照会し、以下同様に繰り返すことである。一例として、図6を検討されたい。図6には、ソース602、デマルチプレクサ604、デコーダ606および606a、コンポーネント608、608a、610、および610a（任意の適当なコンポーネントとすることができる）、およびレンダラ612および612aを含む例示的なパイプライン600が示されている。この場合に、ソース602が、デマルチプレクサ604に照会し、デマルチプレクサ604が、デコーダ606および606aに照会し、以下同様である。その後、照会結果を、パイプラインの上流に返すことができる。

30

【0074】

この例では、ソースが、さまざまなコンポーネントの最大順方向再生レートに関心を持つと仮定する。この例では、コンポーネントの最大レートのそれぞれが、そのコンポーネントの真下に示されている。したがって、たとえば、レンダラ612の最大レートは、2倍速であり、レンダラ612aの最大レートは、3倍速である。ソースがパイプラインの最大レート全般に関して判断を行える形は、2つある。まず、ソースは、保守的な形で、さまざまなコンポーネントのすべてのレートの最小値を単純にとることができる。したがって、この特定の例では、すべてのレートの最小値は2倍速である。したがって、この知識から、ソースは、2倍速未満の要求されたレートのすべてを、有意な量のデータを除去する必要なしにスムーズにレンダリングできると仮定することができる。第2の、より積極的な手法は、さまざまなコンポーネントの間に乗法的効果があると仮定することである。たとえば、第1のコンポーネントが2倍速をサポートでき、その下流のコンポーネントが3倍速をサポートできる場合に、これらのコンポーネントが一緒になって提供できる最大レート変更は、6倍速である。したがって、この図で、レンダラ612は、コンポーネント610によって照会される時に、2倍速（コンポーネントの間の括弧内に示されている）の再生レートを報告する。したがって、コンポーネント610は、乗法的効果のゆえに、4倍速の再生レートをコンポーネント608に報告することができる。同様に、コンポーネント608は、8倍速の再生レートをコンポーネント606に報告することができ、

40

50

以下同様である。この例では、デマルチプレクサ604が、1倍速再生レートだけをサポートすると仮定する。デマルチプレクサ604は、ソース602に、2つの再生レートすなわち16倍速および32倍速を報告することができる。この手法によれば、ソースは、2つの再生レートうちの低い方すなわち16倍速を、全般的にスムーズな再生レートを達成するためにパイプラインによってサポートされる最大再生レートとして選択することができる。

【0075】

ソースが、パイプラインのレート能力（たとえば16倍速）を知ったならば、ソースは、要求されたレートが16倍速以下である時にデータブロック全体をパイプラインに供給できることを知り、そのデータが、このレートでレンダリングされる。ソースは、16倍速を超える要求されたレート変更について、より高いレートを提供することを試みるためになんらかの追加処理を行わなければならないことを知ることができる。これには、パイプラインによる処理のために切り捨てられたおおよび／または選択的なブロックだけを送ることを含めることができる。このモードを、「スキャンモード」と称し、このモードでは、たとえば、キーフレームだけを、処理のためにソースによってパイプラインに送ることができる。

10

【0076】

図7は、一実施形態による方法のステップを説明する流れ図である。この方法は、適切なハードウェア、ソフトウェア、ファームウェア、またはその組合せで実施することができる。図示の例では、この方法を、図6に図示されそれに関して説明されたシステムに関連して実施することができる。

20

【0077】

ステップ700で、データ処理パイプラインを設ける。任意の適切なデータ処理パイプラインを設けることができる。適切なパイプラインには、通常は、なんらかのタイプのメディア（ディスクなど）からソースデータを読み取るか、データを個々のデータストリームに多重化解除するか、なんらかの形でデータをデコードし、処理するか、ユーザが楽しむためにデータをレンダリングする、1つまたは複数のコンポーネントが含まれる。これらの異なる機能性は、異なるコンポーネントによって実行される傾向があるが、そうである必要はない。

【0078】

ステップ702で、再生サポート機能の提供に関してコンポーネントに照会する。再生サポート機能の例には、再生レート（最大、最小、おおよび／またはレートの範囲）、再正方向（順方向、逆方向）、おおよび類似物が、制限なしに含まれる。ステップ704で、照会されたコンポーネントの下流にコンポーネントがあるかどうかを判定する。ある場合には、ステップ706で、次のコンポーネントに進み、ステップ702に戻って、そのコンポーネントに照会して、その再生サポート機能を提供する。追加の下流コンポーネントがない場合には、ステップ708で、再生サポート機能をデータ処理パイプラインの上へ報告する。

30

【0079】

このステップは、2つの異なる形で実施することができる。まず、上で注記したように、各コンポーネントの機能を、ソースに報告することができる。ソースは、その後、知的なデータ処理判断を行うことができる。たとえば、コンポーネントのそれぞれの再生レートをソースに報告することができ、その結果、ソースが、最小再生レートを選択できるようになる。第2に、コンポーネントのそれぞれの機能を、他のすべてのコンポーネントの機能に関して検討することができる。たとえば、特定のパイプラインまたは副パイプライン内のすべてのコンポーネントの再生レートを乗算することができ、その結果、パイプラインまたは副パイプラインによってサポートされる全体的な有効再生レートを確かめることができる。確かめられたならば、ソースは、上で注記したようにデータを処理する方法について知的な判断を行うことができる。

40

【0080】

50

上で説明した例では、すべての適切に構成されたコンポーネント（フィルタ）を、データマニピュレータ（data manipulator）と見なすことができる。すなわち、個々のコンポーネントが、データを部分的に処理し、データをどのように操作したかに関する適当なヒントを渡すことができる。次のコンポーネントは、さらに、所望の再生レートまたは方向の達成に向かってデータを処理することができる。

【0081】

必要であれば、たとえば、ソースが、デコーダに供給するデータの量を減らすと同時に、データに関するタイミング情報（レート変更など）を維持することができる。これによって、デコーダに、レート変更が望まれることのヒントが与えられる。デコーダは、ソースからデータを受け取ることができ、デコードのほかに、レート変更の達成を試みる形でそのデータを処理することができる。チェーン内のコンポーネントは、所望の再生の達成に対処する形でデータを処理するという選択肢を有することができ、あるいは、コンポーネントは、レート変更に関する情報と共にデータを次のコンポーネントに渡すことができる。そのような処理には、データブロック部分の逆転、逆方向再生要求を受け取った場合のタイムスタンプの再割り当て、およびデータからの関連するフラグの除去を含めることができる。

10

【0082】

したがって、この例では、ソースが、ブロックレベルでストリームを逆にし、逆方向再生が意図されていることを示すフラグを追加することができる。ブロックは、その後、下流の他のコンポーネントに伝搬される。コンポーネントは、逆方向フラグを有するデータに出会った時に、ブロックを別の下流のコンポーネントに渡すこと、または、データを処理して、たとえばフレームの順序を逆にし、フラグを除去し、データを次のコンポーネントにプッシュすることという選択肢を有する。最終的に、レンダラが、適当に増加するタイムスタンプを有する順方向ストリームに見えるものを受け取る。

20

【0083】

ソースから出たブロックのレート変更

上で説明した実施形態は、レート変更が所与の時刻に行われ、レート変更が実施されるデータブロックをソースがまだ読み取っていない時に良好に動作する。もう1つの状況があり、この場合には、上で簡単に述べたように、ソースがデータブロックを読み取り、かつ/または処理のために下流コンポーネントにブロックを渡した後に、レート変更要求が受け取られる。

30

【0084】

これから説明する実施形態では、処理をコンポーネントパイプラインまたはフィルタグラフにまたがって分散させることによって、柔軟性が強化される。具体的に言うと、アプリケーションは、特定の時刻またはソース時間にレート変更を実行するようにソースコンポーネントに要求することができる。その後、レート変更情報が、レート変更に関連するソース時間を有するデータサンプルがそのキューに含まれるコンポーネントに出会うまで、パイプラインに沿って伝搬される。そのコンポーネントは、レート変更要求をキューに入れることができ、その結果、レート変更が、そのサンプルについて処理されるようになる。対応するサンプルが、コンポーネントによって処理される時に、コンポーネントは、出力データ（たとえばサンプルのタイムスタンプ）の変更などの必要な処理をすべて実行することができる。さらなる処理が必要な場合には、コンポーネントは、レート変更要求（ならびにすべての他の有用な情報）を、各出力サンプルと共に渡すことができる。最終的に、レンダラが、サンプルを受け取り、レンダリングをスケジューリングすることができる。分散処理の一部として再生またはレート変更の処理を実施するようにレンダラを構成することもできることを諒解されたい。

40

【0085】

もう1つの重要な利点は、ソースのレート変更キューが、まだ送っていないか読み取っていないデータだけに制限されることである。キューの残りは、パイプラインに沿って分散され、このパイプラインは、ほとんどの場合に比較的短い。

50

【 0 0 8 6 】

例として、下記を検討されたい。アプリケーションが、レート変更要求をソースに送る時に、ソースは、そのバッファを調べ、レート変更がそれに関して行われるデータが、バッファに置かれているかどうかを確認する。ソースは、通常は、レート変更が行われるソース時間を記録することによってこれを行うことができる。ソースのバッファ内のデータが、レート変更が行われるソース時間より後の対応するソース時間を有する場合に、データサンプルは、既に下流コンポーネントに渡されている。ソースは、レート変更の対象になるデータを含むバッファを突き止めた場合に、レート変更情報をそのサンプルに関連付けることができる。

【 0 0 8 7 】

すなわち、この実施形態によれば、すべてのデータサンプルに、(1)すべてのレート変更、(2)サンプルを再生しなければならない時間範囲、および(3)サンプルを順方向または逆方向のどちらで再生するかを定義するプロパティを含むリストが含まれる。このリストは、データサンプルと共に伝搬され、その結果、パイプライン内のコンポーネントが、過去に行われたすべてのレート変更のリストを維持する必要がなくなる。

【 0 0 8 8 】

ソースが、そのバッファを検査し、レート変更が行われるサンプルが見つからない場合には、ソースは、レート変更要求を次の下流コンポーネントに送る。次のコンポーネントは、そのバッファを検査して、そのバッファにレート変更が行われるサンプルが含まれるかどうかを確認することによって、同様の処理を実行する。コンポーネントが、適切なデータサンプルを見つけた場合には、そのコンポーネントは、サンプルを処理する時にレート変更情報をサンプルに追加するか、バッファを調べ、バッファ内のサンプルにレート変更情報をアタッチするのいずれかを行うことができる。この処理は、データパイプライン内の適切に構成されたコンポーネントによって実行することができる。

【 0 0 8 9 】

図 8 は、一実施形態による方法のステップを説明する流れ図である。この方法は、適切なハードウェア、ソフトウェア、ファームウェア、またはその組合せで実施することができる。図示の例では、この方法を、図 6 に図示されそれに関して説明されたシステムに関連して実施することができる。

【 0 0 9 0 】

ステップ 8 0 0 で、特定のソース時間に関するレート変更要求を受け取る。通常、レート変更要求は、アプリケーションから発し、ソースに渡される。レート変更要求の受け取りに応答して、ステップ 8 0 2 で、1 つまたは複数のバッファを検査して、レート変更が行われる対応するデータサンプルが存在するかどうかを確認する。検査されるバッファは、レート変更要求を受け取ったコンポーネントに関連するバッファである。サンプルが見つかる場合に(ステップ 8 0 4)、ステップ 8 0 6 で、レート変更情報をサンプルにアタッチする。このステップは、2 つの形で実施することができる。たとえば、レート変更情報を、サンプルがバッファにある間にサンプルにアタッチすることができる。これは、バッファが読取 / 書込キューである時に実施することができる。その代わりに、レート変更情報を、キャッシングし、サンプルがコンポーネントによって処理される時にサンプルにアタッチすることができる。このステップは、バッファが読取専用キューである時に実施することができる。

【 0 0 9 1 】

その一方で、サンプルが、コンポーネントのバッファ内で見つからない場合には、ステップ 8 0 8 で、レート変更要求を次の下流コンポーネントに送る。ステップ 8 1 0 で、下流コンポーネントに関連するバッファを検査する。このステップは、レート変更要求を受け取ったコンポーネントによって実施することができる。サンプルが見つかる場合に(ステップ 8 1 2)、この方法は、ステップ 8 0 6 に分岐することができる。その一方で、サンプルが見つからない場合には、ステップ 8 1 4 で、別の下流コンポーネントがあるかどうかを判定する。そうである場合には、この方法は、ステップ 8 0 8 に分岐し、レート変更

要求を次のコンポーネントに送る。その一方で、追加の下流コンポーネントがない場合には、ステップ 816 で、レート変更要求を破棄する。おそらく、このステップは、レンダラによって実施される。レート変更要求の破棄は、レンダリングの出力レートに対する一時的な影響だけを有する。

【0092】

クリーンポイントがないデータのレート変更

多重化されたデータストリーム (ATSC MPEG2 など) は、ビデオのキーフレームで位置合せされる二次ストリーム (secondary stream) を必ずしも有しない場合がある。これによって、コンテンツを、キーフレームによって区切られた別個のブロックに分割することを試みる時に問題が生じる。

10

【0093】

主ストリーム (primary stream) データのブロックに対応する任意の時間間隔の二次ストリームのデータを生成するには、下記の 2 つの解決策がある。

【0094】

1) トリミング

ソースは、すべてのストリームがその間隔に存在し、各ストリームがその間隔の開始タイムスタンプにスキップするように指示されるように、コンテンツを後ろ向きに検索する。

【0095】

2) フラグメント (fragment) の破棄

ソースは、主ストリームの先頭を探し、二次ストリームは、次のフレームの先頭までブロックフラグメントを破棄する。ストリームは、この間隔の間のデータがなく、不連続性を、間隔の先頭で送らなければならない。

20

【0096】

この時間間隔の後のデータを終了するために、切捨マーカ (truncation marker) を各ストリームに挿入する必要がある。図 16 に、二次ストリームについて余分なデータを読み取る方法と、ビデオストリーム間隔と一致するように二次ストリームを修正できる方法を示す。順方向再生について、フルフレームモード中の同一方向のレート変更は、クリーンポイントによって影響されない。

【0097】

順方向および逆方向のキーフレームスキャンモードについて、二次ストリームは、トリミングまたはフラグメント化を必要としない。大量のデータが、スキップされつつあるので、フラグメント化によってソースの設計が単純になるのでフラグメント化が好ましい場合がある。より複雑なシークを、機能強化として追加することができるが、これによって、シークおよびリーダとの対話に起因する性能コストが生じる場合がある。

30

【0098】

スムーズ逆方向再生について、トリミングされた二次ストリームが、連続的な逆方向オーディオのために必要である。しかし、ソースでのバッファリング要件 (バッファ空間の 2 倍) および余分な高速デコード / 破棄が、実現可能でない場合がある。しかし、オーディオデコードは、ビデオデコードと比較して比較的高速である。

ソースは、トリミング範囲の外のサンプルを破棄する責任を負う。

40

【0099】

実施例

上で説明した実施形態の諸態様を、具体的な実施例に関して説明する。下で説明する例が、例示を目的とし、請求される主題の応用分野を制限するものと解釈されてはならないことを理解されたい。

【0100】

これから説明する例では、下記のアーキテクチャの態様が使用される。IRateChangeData インターフェースは、各サンプルに対して設けられ、レート変更情報、フラグ、およびレート変更位置を維持するのに使用される。サンプルアロケータが、IRateChangeData インターフェースをサポートするサンプルを作成する。インタ

50

ーフェース `IRateChangeControl` (通常は、フィルタの入力ピンまたはコンポーネントへの入力ピンにある) は、レート変更挿入を要求する: `ScheduleRateChange` メソッドを提供する。このインターフェースは、順方向および逆方向の最大デコードレートを照会するメソッド: `GetMaxRate` もサポートする。メソッド: `GetFurthestPosition` は、まだ処理されていない最も遠いストリーム時間を照会し、その後でレート変更をスケジューリングできる時間間隔を判定するために提供される。

【0101】

もう1つのメソッド: `ScheduleRateChangeNow` では、次の使用可能な時間位置でレート変更をスケジューリングすることと、実際の位置を返すことを要求する。このメソッドは、失敗したレート変更から回復することが最も少なく可能であるストリームに対して呼び出すことができる。: `ScheduleRateChange` メソッドは、: `ScheduleRateChangeNow` によって返される位置を使用する他のストリームに対して呼び出すことができる。

10

【0102】

さらに、この例では、データフォーマットに関する下記の仮定にそって描かれる。スムーズな逆転、逆方向および順方向のスキャン再生 (用語「スキャン」は、パイプラインまたはグラフによってサポートされる最大レートを超える再生速度に起因する切り捨てられた再生を表すのに使用される) について、通常は、ソースが、ストリームをクリーンポイントによって区切られるばらばらのブロックに分解できなければならない。クリーンポイントは、圧縮データ内で、すべてのストリームが同期化される位置である (すなわち、各ストリームが、新しいサンプルで同一のプレゼンテーションタイムスタンプで開始される)。たとえば、DVD ソースでは、ソースブロックとして `VOBU` が使用される。他のコンテンツフォーマットについて、ソースは、ストリームを `GOP` に分解する。というのは、ビデオおよびオーディオのストリームで、キーフレーム境界で新しいフレームが開始されるからである。一般的な `MPEG-2` コンテンツなどのクリーンポイントがないコンテンツについては、前の節の「クリーンポイントがないデータのレート変更」を参照されたい。

20

【0103】

レート変更要求のスケジューリングおよび処理に関して:

30

- ・フィルタは、`IRateChangeControl::ScheduleRateChange` 要求をデコーダに送って、特定のソースタイムスタンプにミッドサンプル (`mid-sample`) レート変更を挿入する。デコーダは、レート変更位置を含むサンプルに出会うかどうかを判定する。そのタイムスタンプを既に処理したか、レート変更を処理しない場合に、デコーダは、入力ピンに対応させることによって制御される出力ピンのそれぞれにレート変更要求を渡す。

【0104】

- ・その代わりに、即時レート変更をスケジューリングするために、フィルタは、`IRateChangeControl::ScheduleRateChangeNow` 要求をデコーダ (通常は、たとえば顕著な「音飛び」を引き起こすオーディオなど、データ消失から簡単には回復できないストリームのデコーダ) に送って、次の使用可能なソースタイムスタンプにミッドサンプルレート変更を挿入する。この要求は、あるコンポーネントが、このレート変更挿入要求を処理しなければならないと判断するまで、再帰的に下流に渡される。実際の位置が、ストリームを介して最初のフィルタまで返される。フィルタは、`ScheduleRateChangeNow` 要求によって返されるタイムスタンプを使用して、他のストリームにレート変更要求を伝搬させる。

40

【0105】

- ・デコーダが要求を処理する時に、デコーダは、レート変更タイムスタンプを含むサンプルを受け取る時に、処理されるレート変更情報を内部的にキューイングする。サンプルに出会った時に、デコーダは、すべてのレート変更補間を実行し、レート変更情報を、出力

50

サンプルの `IRateChangeData` インターフェースにコピーする。

【0106】

・データストリーム内のタイムスタンプ情報は、ソースまたは中間フィルタによって直接には変更されない。各フィルタが、単純に、ストリームの内容を修正する。

【0107】

レート変更の計算に関して：

・ソースフィルタは、`IRateChangeControl::GetMaxRate` メソッドを使用して、デコーダの順方向および逆方向の最大再生ビットレートを照会する。グローバルフラグに真がセットされている場合には、フィルタは、対応する出力ピンおよびその最大内部レートの両方の最大レートを計算し、返さなければならない。

10

【0108】

`IRateChangeData` インターフェースに関して：

・`IRateChangeData` インターフェースは、レート変更の存在を示すことと、サンプルに関連するレート変更情報を保管することの両方に使用される。

【0109】

・`IRateChangeData` インターフェースは、下記の情報を維持する：

○ `LONG Rate` - 符号付き固定小数点スケール値 ($\times 10000$)。負の値は、逆方向再生を示す。0の値は、レート変更イベントがこのサンプルに存在しないことを示す (フラグフィールドにまだサンプル情報が含まれる場合がある)。

○ `REFERENCE TIME Location` - レート変更のソースタイムスタンプ位置。

20

○ `DWORD Flags` - 下記のフラグのビット単位のOR。

`Flag__ReverseStart` - サンプルの「逆方向再生」グループの最初のサンプル。

`Flag__ReverseEnd` - サンプルの「逆方向再生」グループの最後のサンプル。

`Flag__ScanMode` - 順方向または逆方向のスキャンモードでのビデオセクションの部分。

`Flag__TruncationPoint` - サンプルをこの点で切り捨てなければならないことの位置指定。「Rate」フィールドは使用されない。

30

`Flag__SkipPoint` - サンプルの先頭からこの点までを切り捨てなければならないことの位置指定。「Rate」フィールドは使用されない。

【0110】

異なる再生レートでのタイムスタンプおよびデータの解釈に関して：

・ソースフィルタは、タイムスタンプが連続し、単調増加することを保証する責任を負う。タイムスタンプは、連続的な線形のビデオに見えるように修正される。不連続は、ソースデータに途切れがあることを示す。

【0111】

・再正方向の変更は、サンプル境界でのみ発生することができ、サンプルの「逆方向再生」グループの中途では発生しない (下を参照されたい)。

40

【0112】

・スムーズ順方向再生で (最大順方向レート未満)、サンプルは、図9に示されているように普通に送られる。

【0113】

・スムーズ逆方向再生に関して、デコーダが、サンプルのブロックを順方向にデコードし、その後、フレームを逆の順序で送達すると仮定する。`Flag__ReverseStart` フラグ (`IRateChangeData::Flags` データフィールド内の) によって、このサンプルが、逆方向に再生されることを意図されたサンプルのグループの最初のサンプルであることが示される。`Flag__ReverseEnd` フラグによって、サンプルが、サンプルの「逆方向再生」グループの最後のサンプルであることが示される

50

。サンプルが再生される間隔の最初と最後のタイムスタンプは、変更されない。データの表示順序は、図 10 に示されているように逆転される。

【 0 1 1 4 】

・最初の間隔のフレームの表示タイムスタンプは、下記のように計算される：

$$\begin{aligned} R_{frame} &= \text{IntervalStart} + \text{IntervalLength} - \text{Offset} \\ &= R_{s3} + (R_{e3} - R_{s3}) - (R_{old} - R_{s3}) \\ &= (R_{e3} - R_{old}) + R_{s3} \end{aligned}$$

【 0 1 1 5 】

・最大レートを超える順方向（または逆方向）スキャン再生中には、キーフレーム（I フレーム）だけが、時間間隔に表示されるビデオの近似としてデコーダに送られる（不連続性なしに）。出力タイムスタンプは、図 11 に示されているように、元の間隔の終了タイムスタンプと同一である。タイムスタンプは、1 倍速で再生される線形のビデオとして見えるように修正される。開始タイムスタンプおよび終了タイムスタンプによって、サンプルが再生される期間が示される。

10

【 0 1 1 6 】

・スキャン（キーフレームだけを送る）とスムーズモード（すべてのビデオを送る）の間の推移によって、本来的に実際のデータが変更される。最大レート未満のレートでのレート変更によって、サンプルがスムーズモードで送られることが指示され、そうでない場合には、サンプルがスキャンモードで送られる。各サンプルの *I R a t e C h a n g e D a t a* フラグに、デコーダがサンプルのモード（スムーズまたはスキャン、逆方向または順方向）を決定するのを助けるヒントが含まれる。

20

【 0 1 1 7 】

・方向変更は、レートの符号の変化によって示される。方向変更は、サンプル境界でのみ行うことができ、サンプルの「逆方向再生」グループの途中で行うことはできない。

【 0 1 1 8 】

・図 12 に示されているように、複数の同一方向のレート変更を、1 つのサンプル内で、およびスキャンモードと非スキャンモードの混合されたサンプルにまたがって、行うことができる。

【 0 1 1 9 】

・この例のレート変更機構では、ビデオフォーマットを、キーフレームによって分離されたブロックに分割できることだけが仮定される。タイムスタンプが単調増加することを保証するために、各ブロック（キーフレームの間）の長さは、終了タイムスタンプと後続キーフレームの次のタイムスタンプを計算するために、ソースフィルタが計算可能でなければならない。

30

【 0 1 2 0 】

・タイムスタンプは、通常は、パイプラインを介して伝搬する際に、フィルタによって修正されない。レンダラが、タイムスタンプを修正して（レート変更コマンドストリームを使用して）、出力タイムスタンプを作る。コンポーネントは、タイムスタンプを修正しない場合に、位置について照会された時またはレート変更情報を下流に伝搬する時に、すべての出力タイムスタンプ情報を入力タイムスタンプにマッピングしなければならない（逆も同様）。

40

【 0 1 2 1 】

・ソースフィルタが、タイムスタンプをサンプルに付加せず、それらが逆方向再生ブロックの一部であることを示すことが可能である。この状況では、インバンドタイムスタンプを読み取る最初のフィルタが、ブロック全体に属するサンプルに対して単調に増加するタイムスタンプを生成する責任を負う。

【 0 1 2 2 】

・中間フィルタは、図 13 に示されているように、逆方向サンプルを、受け取られた順序と逆の順序で送られる、より小さい逆方向サンプルに分割することを許可される。

【 0 1 2 3 】

50

上で説明した例によって、複数の長所が実施される。第1に、レート変更が、同一方向かつ同一モードである場合に、レート変更をリアルタイムの対話的変更として提供することができる。さらに、スキャンモードで最大個数のフレームが表示される時に、デコーダのビットレートを超えない。さらに、タイムスタンプは、レンダラのみによって修正され、タイムスタンプマッピングコードを、集中化し、隠すことができる。

【0124】

インターフェース

この特定の実施例では、下記のインターフェースを使用することができる：

Interface IRateChangeControl

10

{

**HRESULT ScheduleRateChange(TIME SourcePosition,
LONG Rate)**

**HRESULT ScheduleRateChangeNow(LONG Rate,
TIME* pActualSourcePosition)**

HRESULT ScheduleTruncation(TIME SourcePosition)

HRESULT ScheduleSkip(TIME SourcePosition)

20

**RESULT GetMaxRate(BOOL IncludeDownstream,
DWORD* pForwardRate, DWORD* pReverseRate)**

**HRESULT GetFurthestPosition(BOOL IncludeDownstream,
TIME* pSourcePosition, TIME* pValidDuration)**

}

【0125】

メソッド **ScheduleRateChange()** によって、レート変更挿入を要求する。

30

・ **SourcePosition** パラメータによって、レート変更を実行するタイムスタンプを、ソースタイムスタンプに関して指定する。

【0126】

・ **Rate** によって、レート × 10000 を指定する。このレートは、逆方向再生の場合に負になる。

【0127】

メソッド **ScheduleRateChangeNow()** によって、最も早い使用可能な時にレート変更挿入が要求され、このメソッドは、そのレート変更が行われる時刻を返す。

・ **pActualSourcePosition** パラメータによって、レート変更がスケジューリングされた実際のタイムスタンプが返される。

40

【0128】

・ **Rate** によって、レート × 10000 を指定する。このレートは、逆方向再生の場合に負になる。

【0129】

メソッド **ScheduleTruncation()** によって、データ切捨を要求する。

・ **SourcePosition** パラメータによって、サンプルを切り捨てるタイムスタンプを指定する。

【0130】

・ メソッド **ScheduleSkip()** によって、サンプルの先頭からのデータスキップ

50

プを要求する。

【0131】

・SourcePositionパラメータによって、サンプルの先頭からの、スキップされるタイムスタンプを指定する。

【0132】

メソッドGetMaxRate()によって、最大順方向デコードレートおよび最大逆方向デコードレートを照会する。

・pForwardRateパラメータおよびpBackwardRateパラメータによって、最大相対順方向再生レートおよび最大相対逆方向再生レートを示す正の値が返される。

10

【0133】

・IncludeDownstreamフラグによって、コンポーネントがその計算に下流依存性を含めなければならないことが示される。

メソッドGetFurthestPosition()によって、まだ処理されていないもっとも遠いストリーム時刻を照会する。

【0134】

・pSourcePositionは、出力の現在のタイムスタンプ(ソースタイムスタンプに関する)を示す戻りパラメータである。

【0135】

・pValidPeriodは、レート変更をスケジューリングできるソース位置の後の持続時間(すなわち、このソース位置の次の使用可能なソース位置までの時間)を示す戻りパラメータである。

20

【0136】

コンポーネントの責任

この特定の実施例では、さまざまなコンポーネントが、下で概要を示す責任を有する。これらの責任が、読者に責任の1つの可能な割振りを示すために説明されることを諒解されたい。下の説明は、請求される主題の応用分野を特定の実施形態に制限することを意図されたものではない。

【0137】

一般に、ソースは、連続するデータストリームを生成する責任を負い、デコーダは、ストリームを修正する責任を負い、レンダラは、最終的にレート変更コンポーネントである。レンダラが、レート変更機構をサポートしない場合には、前のフィルタが、レート変更を実行するためにタイムスタンプを変更することによって、レンダラの代わりに動作する責任を負う。

30

【0138】

ソース

ソースは、タイムスタンプ情報を生成し、管理する責任を負う。ソースは、タイムスタンプ不連続性を置き換え、逆方向再生ブロックを順方向ストリームブロックに反転し、それぞれに逆方向で再生するタグを付けることによって、単調に増加するストリームを生成しなければならない。

40

【0139】

ソースは、特定の位置でのレート変更に関するユーザ要求を実施する方法を見つける責任も負う。また、ソースは、「今」実行しなければならないレート変更をスケジューリングする方法を、下記のいずれかによってを見つける。

【0140】

・下流フィルタの位置を照会し、使用可能な位置および使用可能な時間間隔に基づいて実際の変更をスケジューリングする場所を判断することによって。ソースは、使用可能な位置からの最短距離を実施して、コンポーネントを介する伝搬遅延を考慮に入れることができる。

【0141】

50

・ストリームに対して `ScheduleRateChangeNow` メソッドを実行し、要求の（返された）位置を `ScheduleRateChange` を使用して他のストリームに伝搬することによって。

【0142】

サンプルは、変化するフォーマット（たとえば、順方向サンプル対逆方向サンプル）で送られるので、ソースは、最も最近の方向変更のブロック（たとえば、GOPまたはVOBU）履歴を維持する。異なる方向でのレート変更は、最も最近の方向変更の後でのみ行われる可能性がある。

【0143】

デコーダおよび中間コンポーネント

10

デコーダは、主に、レンダラを助けるストリームプリプロセッサとして働く（例えば、ビットレートの切捨、スローモーション用のデータの追加、および類似物によって）。レート変更を受け取る時に、デコーダは、できる限り補間を実行する。デコーダは、後続処理のために次のフィルタにレート変更を渡すこともできる。

【0144】

デコーダは、逆方向再生ブロックを、レンダラによる処理のためにより小さい逆方向ブロックに分解することがもできる。たとえば、DVDデータの場合に、元のVOBUブロックを、複数のより小さいサブブロック（この例ではGOP）に分解することができる。サブブロックの順序は、逆転され、そのタイムスタンプも、逆転される（図14を参照されたい）。レンダラは、新しいブロックを受け取り、受け取る時に各ブロックを逆転することができる。これは、レンダラが維持を必要とするバッファリングされる未圧縮データの量を減らすという長所を有する。

20

【0145】

次のフィルタ（通常はレンダラ）が、レート変更をサポートしない場合には、デコーダが、出力タイムスタンプを修正して、レート変更を実行しなければならない。

【0146】

レンダラ

レンダラは、最終的にレート変更を実施する。レンダラは、レート変更後の有効プレゼンテーションタイムスタンプを計算する。レンダラは、レート変更履歴も維持し、その結果、たとえばレート変更を行えない時、またはレンダラが過去に発生したレート変更要求を受け取る時に、サルベージ/障害状況进行处理できるようにする。

30

【0147】

さらに、レンダラは、出力フレームのバッファリングを実行できる場合に、アクセラレーション技法を使用して、実際に逆方向再生を実行することができる。レンダラは、デコーダから逆の順序のサンプルを受け取り、デコードされたフレームの記録またはキャッシングを開始する。逆ブロックの終りに達したならば、キャッシングされたフレームを、逆の順序でレンダリングすることができる。

【0148】

レンダラのタイムスタンプマネージャ

タイムスタンプマネージャコードを、上で注記したように、レンダラが入力タイムスタンプ（およびレート変更要求とおそらくは現在のクロック位置）をサンプルの出力タイムスタンプにマッピングするのに使用することができるレート変更管理機能に抽象化することができる。

40

【0149】

一例として、図14および以下の説明を検討されたい。図14に、指定された場所でユーザによって行われるレート変更要求を示す。12フレームのコンテンツが、さまざまなブロックサイズでレンダラに送られたと仮定する。ユーザは、時刻00:00:15に2倍速のレート変更を要求した。

【0150】

ソースは、コンテンツ固有の位置を、対応する出力サンプル7のタイムスタンプにマッピ

50

ングする。レート変更要求を、各出力ピンに送って、次のフィルタがレート変更を挿入することを要求する。この例では、ビデオデコーダが、バッファリングされたサンプル7を有し、したがって、ビデオデコーダは、レート変更情報をサンプル7に付加する。オーディオデコーダは、既にサンプル7を処理しており、したがって、要求をオーディオレンダラに渡す。レンダラは、次のバッファを、異なるレートで再生される2つのセクションに分割する。各レンダラは、タイムスタンプマネージャに問い合わせ、各入力タイムスタンプを出力タイムスタンプにマッピングする。

【0151】

ユーザが即座にレートを変更することを望む時に、第2の状況が発生する。ソースは、次に使用可能なタイムスタンプおよびそれが発生するまでの使用可能な時間の長さを照会する。このタイムスタンプを使用して、ソースが、パイプラインまたはフィルタグラフを介してレート変更を伝搬させる。

10

【0152】

次に使用可能なタイムスタンプは、タイムスタンプマネージャによって、または分散された再帰的な形で計算することができる。タイムスタンプマネージャは、チェーンに沿った各フィルタがストリームに対するレート変更を考慮に入れる（レート変更について各キューイングされたサンプルを照会することによって）必要がないという長所を有する。例として、図15を検討されたい。

【0153】

この図では、タイムスタンプマネージャにレジストリが含まれる。各レンダラは、その現在の進行状況をタイムスタンプマネージャに登録し、更新する。タイムスタンプマネージャは、次の使用可能な時刻を計算して、レート変更をスケジューリングし、その時刻をソースに渡すことができる（たとえば、フレーム5で、レート変更を伝搬させるのに2フレーム）。ソースは、レンダラが（ソースタイムスタンプで）どこにあるかを見つけることもでき、アプリケーションに対するソース位置を計算することができる。タイムスタンプマネージャは、各レンダラの位置を外挿し、ソースに返す。

20

【0154】

タイムスタンプマッピング機能およびレジストリ機能を中央位置で行わせることによって、これらの機能を他のパイプライン内のコンポーネントから除去できるという点で、標準化および均一性が強化される。さらに、タイムスタンプマッピングの集中化された位置を提供することによって、エラーを減らすことができる。

30

【0155】

結論

上で説明した方法およびシステムは、レンダリング可能なデジタルデータを処理する改良された方法およびシステムを提供することができる。さまざまな方法およびシステムによって、ビデオデータ、オーディオ/ビデオデータ、オーディオ/ビデオ/サブピクチャデータ、および類似物などのデジタルデータを、ユーザの経験を強化する形での順方向と逆方向の両方での再生を可能にする形で処理できるようにする。さまざまな実施形態は、ユーザの経験を強化できる形での再生レート変更も処理することができる。

【0156】

40

構造的特徴および/または方法論的ステップに固有の言語で本発明を説明したが、請求項で定義される発明が、説明した特定の特徴またはステップに必ずしも制限されないことを理解されたい。そうではなく、特定の特徴およびステップは、請求される発明を実施する好ましい形として開示されたものである。

【図面の簡単な説明】

【図1】1つまたは複数の実施形態に従って処理することができるデータの例示的ブロックを示す図である。

【図2】データブロックを処理するシステムのブロック図である。

【図3】説明される実施形態の原理をその中で実施することができる例示的コンピューティング環境のブロック図である。

50

【図 4】一実施形態によるデータブロックを処理するシステムのブロック図である。

【図 5】一実施形態による方法のステップを説明する流れ図である。

【図 6】1つまたは複数の説明される実施形態のデータブロックを処理するシステムおよびある態様を示すブロック図である。

【図 7】一実施形態による方法のステップを説明する流れ図である。

【図 8】一実施形態による方法のステップを説明する流れ図である。

【図 9】1つまたは複数の説明される実施形態を理解するのに有用な形で例示的なデータブロックを示す図である。

【図 10】1つまたは複数の説明される実施形態を理解するのに有用な形で例示的なデータブロックを示す図である。

10

【図 11】1つまたは複数の説明される実施形態を理解するのに有用な形で例示的なデータブロックを示す図である。

【図 12】1つまたは複数の説明される実施形態を理解するのに有用な形で例示的なデータブロックを示す図である。

【図 13】1つまたは複数の説明される実施形態を理解するのに有用な形で例示的なデータブロックを示す図である。

【図 14】一実施形態によるデータブロックを処理する例示的システムのブロック図である。

【図 15】一実施形態によるデータブロックを処理する例示的システムのブロック図である。

20

【図 16】一実施形態による複雑なデータを処理する技法を示す図である。

【符号の説明】

200 システム

202 アプリケーション

204 ソースコンポーネント

206 DVD

208 デマルチプレクサ

210 ビデオデコーダ

212 オーディオデコーダ

214 サブピクチャデコーダ

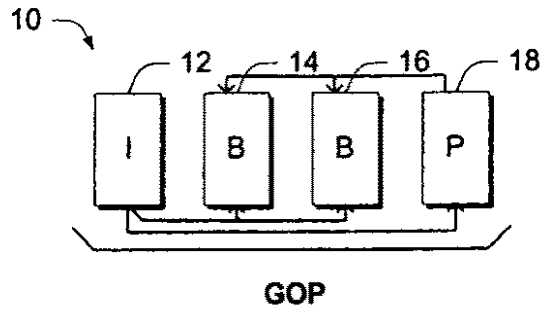
216 ビデオレンダラ

218 オーディオレンダラ

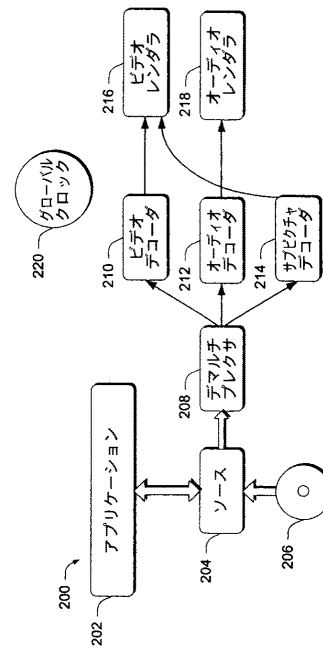
220 グローバルクロック

30

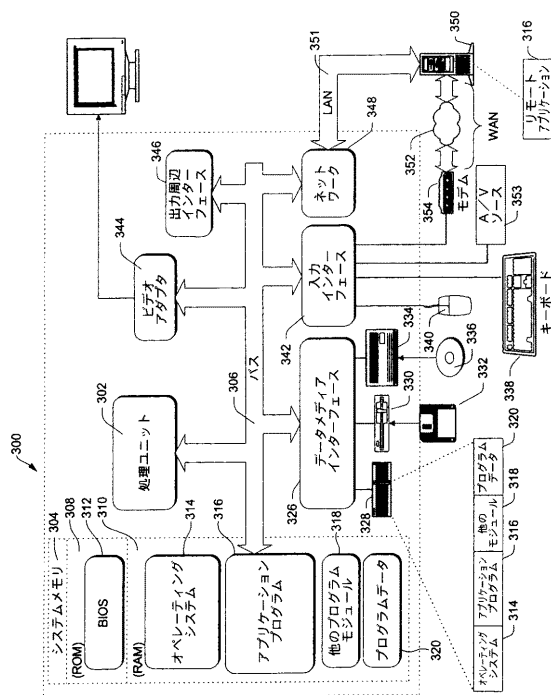
【図 1】



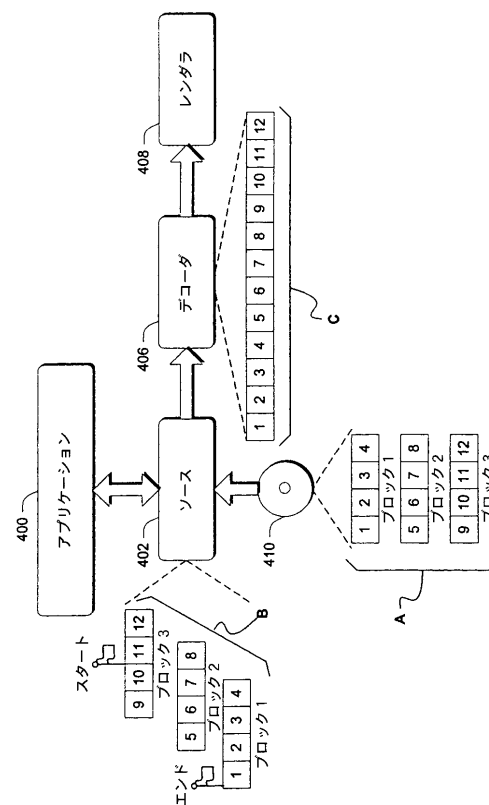
【図 2】



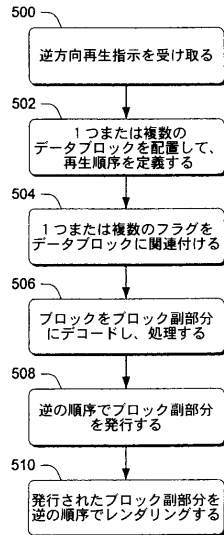
【図 3】



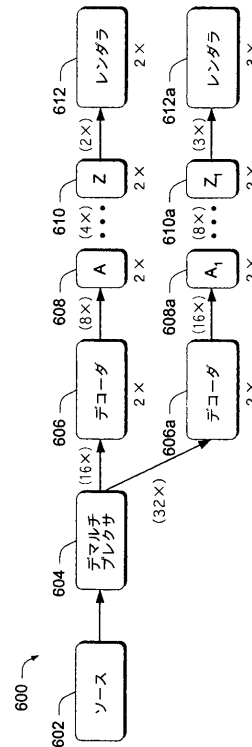
【図 4】



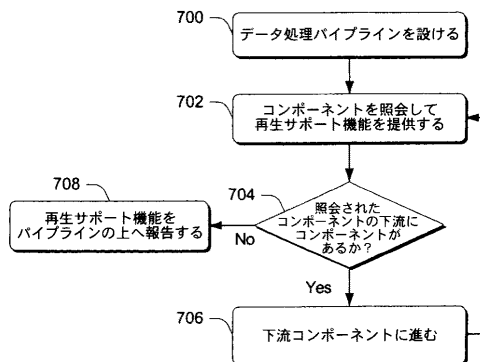
【図 5】



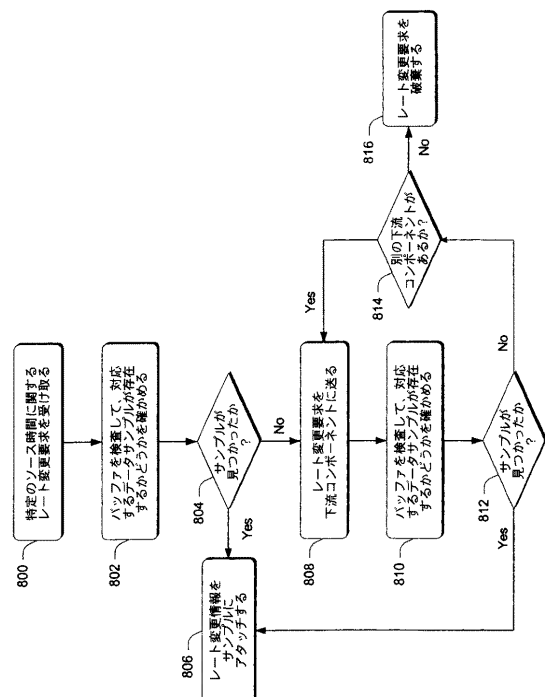
【図 6】



【図 7】



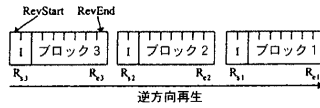
【図 8】



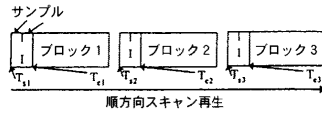
【図 9】



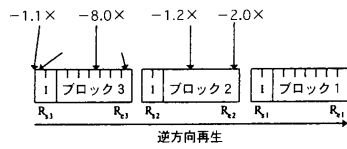
【図 10】



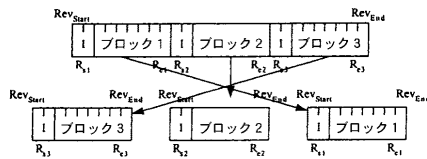
【図 11】



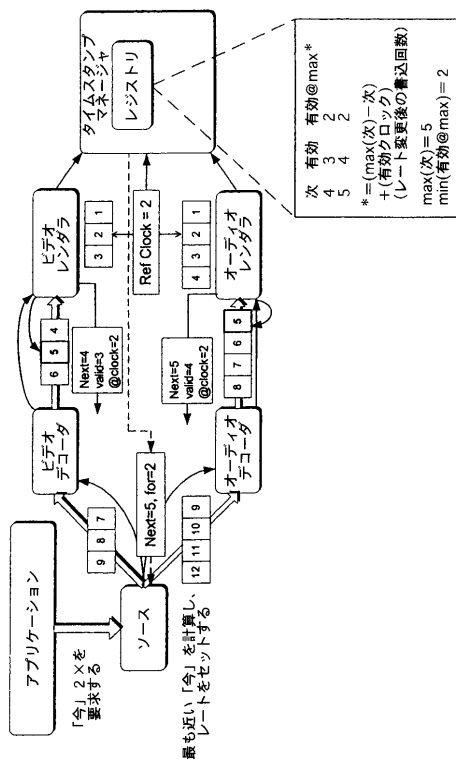
【図 12】



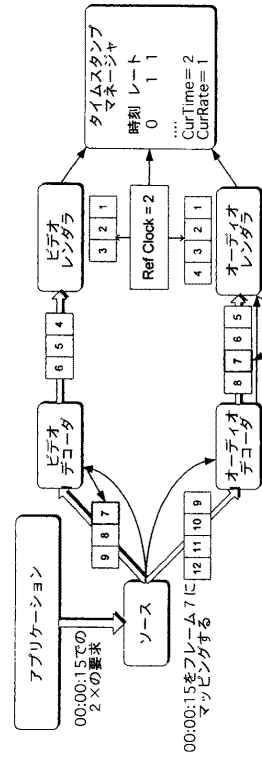
【図 13】



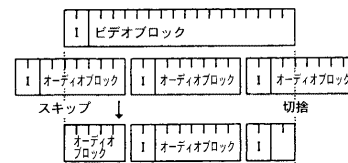
【図 15】



【図 14】



【図 16】



フロントページの続き

- (74)代理人 100091063
弁理士 田中 英夫
- (74)代理人 100153028
弁理士 上田 忠
- (74)代理人 100120112
弁理士 中西 基晴
- (74)代理人 100113974
弁理士 田中 拓人
- (72)発明者 グレン エフ．エバンス
アメリカ合衆国 9 8 0 3 4 ワシントン州 カークランド ノースイースト 1 3 3 プレイス
7 8 3 3
- (72)発明者 アロック チャクラバーティ
アメリカ合衆国 9 8 0 0 6 ワシントン州 ベルビュー 1 4 1 プレイス サウスイースト
5 7 2 4
- (72)発明者 マティス エー．ゲイツ
アメリカ合衆国 9 8 1 1 2 ワシントン州 シアトル ブロードムーア ドライブ イースト
1 3 1 8

審査官 日下 善之

- (56)参考文献 特開平 1 1 - 0 3 9 1 7 7 (J P , A)
特開平 0 9 - 2 8 2 1 6 6 (J P , A)
特開平 0 2 - 1 1 0 7 8 3 (J P , A)
特開 2 0 0 2 - 0 1 0 2 1 3 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

H04N 5/92
H04N 5/76
H04N 5/85