

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6284888号
(P6284888)

(45) 発行日 平成30年2月28日 (2018. 2. 28)

(24) 登録日 平成30年2月9日 (2018. 2. 9)

(51) Int. Cl. F I
HO 4 N 21/436 (2011. 01) HO 4 N 21/436
HO 4 N 21/433 (2011. 01) HO 4 N 21/433

請求項の数 38 (全 50 頁)

(21) 出願番号	特願2014-558987 (P2014-558987)	(73) 特許権者	595020643
(86) (22) 出願日	平成25年2月28日 (2013. 2. 28)		クァアルコム・インコーポレイテッド
(65) 公表番号	特表2015-513842 (P2015-513842A)		QUALCOMM INCORPORATED
(43) 公表日	平成27年5月14日 (2015. 5. 14)		ED
(86) 国際出願番号	PCT/US2013/028383		アメリカ合衆国、カリフォルニア州 92
(87) 国際公開番号	W02013/130853		121-1714、サン・ディエゴ、モア
(87) 国際公開日	平成25年9月6日 (2013. 9. 6)		ハウス・ドライブ 5775
審査請求日	平成28年2月2日 (2016. 2. 2)	(74) 代理人	100108855
(31) 優先権主張番号	61/604, 086		弁理士 蔵田 昌俊
(32) 優先日	平成24年2月28日 (2012. 2. 28)	(74) 代理人	100109830
(33) 優先権主張国	米国 (US)		弁理士 福原 淑弘
(31) 優先権主張番号	61/604, 087	(74) 代理人	100103034
(32) 優先日	平成24年2月28日 (2012. 2. 28)		弁理士 野河 信久
(33) 優先権主張国	米国 (US)	(74) 代理人	100075672
			弁理士 峰 隆司

最終頁に続く

(54) 【発明の名称】 ワイヤレスディスプレイシステム内のソースデバイスでのフレームの取込みおよびバッファリング

(57) 【特許請求の範囲】

【請求項 1】

ワイヤレスディスプレイ (WD) システムにおいて 1 つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、

前記ソースデバイスでメディアデータのフレーム更新を取り込むことと、

前記 WD システム内で前記通信セッションを確立すると、および前記 1 つまたは複数のシンクデバイスへの送信用にどのフレーム更新を処理するよりも前に、処理パイプライン内の待ち時間を削減するために、最小サイズのバッファを含むように、およびハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成することであって、ここにおいて、前記最小サイズのバッファは、前記取り込まれたフレーム更新のうちの少なくとも直近の 1 つを保持することが可能であるサイズをそれぞれが有する少なくとも 2 つのバッファを備える、構成することと、

前記ソースデバイスの前記処理パイプラインに含まれる前記最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも前記直近の 1 つをバッファリングすることと、

ハードウェア加速を使用して前記最小サイズのバッファ内の前記フレーム更新を取り出すことと、

前記 1 つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファから取り出された前記フレーム更新を処理することと、

前記ソースデバイスのコード化ピクチャバッファに、前記 1 つまたは複数のシンクデバ

10

20

イスへの送信より前に、符号化されたフレーム更新をバッファリングすることと、を備え、

前記符号化されたフレーム更新をバッファリングすることが、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングすることを備える、

方法。

【請求項 2】

メディアデータのフレーム更新を取り込むことが、前記ソースデバイスのメモリにフレーム更新を記憶することを備える、請求項 1 に記載の方法。

【請求項 3】

前記フレーム更新のうちの少なくとも直近の 1 つをバッファリングすることが、前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つへのポインタのうちの 1 つをバッファリングすることを備える、請求項 2 に記載の方法。

【請求項 4】

前記最小サイズのバッファが一杯か一杯でないかを判定することと、

前記最小サイズのバッファが一杯でないと判定すると、または前記最小サイズのバッファが一杯であるときに前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除した後、前記最小サイズのバッファに前記フレーム更新のうちの前記直近の 1 つをバッファリングすることとを備える、請求項 1 に記載の方法。

【請求項 5】

ハードウェア加速を使用して前記フレーム更新を取り出すことが、前記ソースデバイスの中央処理装置（CPU）を使用せずに、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させることを備える、請求項 1 に記載の方法。

【請求項 6】

前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にビデオ処理エンジンとマルチメディアディスプレイプロセッサとの間の 2 つのライトバック（WB）バッファを備える、請求項 1 に記載の方法。

【請求項 7】

前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にエンコーダとビデオ処理エンジンとの間の 4 つのフレームバッファを備える、請求項 1 に記載の方法。

【請求項 8】

前記処理パイプラインを用いて前記フレーム更新を処理することが、

前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、

前記 1 つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを備える、請求項 1 に記載の方法。。

【請求項 9】

前記 1 つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信することと、

新しいフレーム更新を受信せずに、ある時間期間後に、前記 1 つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとをさらに備える、請求項 1 に記載の方法。

【請求項 10】

前記 1 つまたは複数のシンクデバイスへのビデオデータトランスポートよりもオーディオデータトランスポートを優先させるように、前記ソースデバイスの前記処理パイプラインを構成することをさらに備える、請求項 1 に記載の方法。

【請求項 11】

前記シンクデバイスのうちの少なくとも1つからトランスポートチャネル状態を記述するフィードバック情報を受信することと、

前記フィードバック情報に基づいて前記ソースデバイスの前記処理パイプラインを修正することとをさらに備える、請求項1に記載の方法。

【請求項12】

前記ソースデバイスで前記通信セッションを確立することが、前記通信セッション用の前記メディアデータを複数のシンクデバイスに広告することと、前記1つまたは複数のシンクデバイスから前記通信セッションへの参加の要求を受信することとを備える、請求項1に記載の方法。

【請求項13】

ワイヤレスディスプレイ(WD)システムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、メディアデータのフレーム更新を取り込むことと、処理パイプラインに含まれる最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、前記1つまたは複数のシンクデバイスへの送信用に前記最小サイズのバッファからの前記フレーム更新を処理することとを行うように構成された、1つまたは複数の処理ユニットを含む前記処理パイプラインと、

前記WDシステム内で前記通信セッションを確立すると、および前記1つまたは複数のシンクデバイスへの送信用にどのフレーム更新を処理するよりも前に、前記処理パイプライン内の待ち時間を削減するために、前記最小サイズのバッファを含むように、およびハードウェア加速を使用するように前記処理パイプラインを構成するように構成されたパイプラインマネージャであって、ここにおいて、前記最小サイズのバッファは、前記取り込まれたフレーム更新のうちの少なくとも直近の1つを保持することが可能であるサイズをそれぞれが有する少なくとも2つのバッファを備える、パイプラインマネージャと、

前記処理パイプラインによる処理用に前記最小サイズのバッファから前記フレーム更新を取り出すように構成されたハードウェア加速器とを備え、

前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記ソースデバイスのコード化ピクチャバッファに、前記1つまたは複数のシンクデバイスへの送信より前に、符号化されたフレーム更新をバッファリングするように構成され、

前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングするように構成された、

ソースデバイス。

【請求項14】

前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記ソースデバイスのメモリにフレーム更新を記憶するように構成された、請求項13に記載のソース。

【請求項15】

前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記メモリに記憶された前記フレーム更新のうちの前記直近の1つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の1つへのポインタのうちの1つをバッファリングするように構成された、請求項14に記載のソースデバイス。

【請求項16】

前記処理パイプラインの前記1つまたは複数の処理ユニットが、

前記最小サイズのバッファが一杯か一杯でないかを判定することと、

前記最小サイズのバッファが一杯でないと判定すると、または前記最小サイズのバッファが一杯であるときに前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除した後、前記最小サイズのバッファに前記フレーム更新のうちの前記直近の1つをバッファリングすることとを行うように構成された、請求項13に記載のソースデバイス。

【請求項17】

前記ハードウェア加速器が、前記ソースデバイスの中央処理装置（CPU）を使用せずに、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させるように構成された、請求項 13 に記載のソースデバイス。

【請求項 18】

前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にビデオ処理エンジンとマルチメディアディスプレイプロセッサとの間の 2 つのライトバック（WB）バッファを備える、請求項 13 に記載のソースデバイス。

【請求項 19】

前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にエンコーダとビデオ処理エンジンとの間の 4 つのフレームバッファを備える、請求項 13 に記載のソースデバイス。

10

【請求項 20】

前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、
前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、
前記 1 つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを行うように構成された、請求項 13 に記載のソースデバイス。

【請求項 21】

前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、
前記 1 つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信することと

20

、
新しいフレーム更新を受信せずに、ある時間期間後に、前記 1 つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとを行うように構成された、請求項 13 に記載のソースデバイス。

【請求項 22】

前記パイプラインマネージャが、前記 1 つまたは複数のシンクデバイスへのビデオデータトランスポートよりもオーディオデータトランスポートを優先させるように前記処理パイプラインを構成する、請求項 13 に記載のソースデバイス。

【請求項 23】

前記パイプラインマネージャが、
前記シンクデバイスのうちの少なくとも 1 つからトランスポートチャネル状態を記述するフィードバック情報を受信することと、
前記フィードバック情報に基づいて前記ソースデバイスの前記処理パイプラインを修正することとを行う、請求項 13 に記載のソースデバイス。

30

【請求項 24】

前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記通信セッション用の前記メディアデータを複数のシンクデバイスに広告することと、前記ソースデバイスで前記通信セッションを確立するために前記 1 つまたは複数のシンクデバイスから前記通信セッションへの参加の要求を受信することとを行うように構成された、請求項 13 に記載のソースデバイス。

40

【請求項 25】

ワイヤレスディスプレイ（WD）システムにおいて 1 つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立するための手段と、
メディアデータのフレーム更新を取り込むための手段と、

前記WDシステム内で前記通信セッションを確立すると、および前記 1 つまたは複数のシンクデバイスへの送信用にどのフレーム更新を処理するよりも前に、処理パイプライン内の待ち時間を削減するために、最小サイズのバッファを含むように、およびハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成するための手段であって、ここにおいて、前記最小サイズのバッファは、前記取り込まれたフレーム更

50

新のうちの少なくとも直近の１つを保持することが可能であるサイズをそれぞれが有する少なくとも２つのバッファを備える、構成するための手段と、

前記ソースデバイスの前記処理パイプラインに含まれる前記最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも前記直近の１つをバッファリングするための手段と、

ハードウェア加速を使用して前記最小サイズのバッファから前記フレーム更新を取り出すための手段と、

前記１つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファから取り出された前記フレーム更新を処理するための手段と、

前記ソースデバイスのコード化ピクチャバッファに、前記１つまたは複数のシンクデバイスへの送信より前に、符号化されたフレーム更新をバッファリングするための手段と、を備え、

ここにおいて、前記符号化されたフレーム更新をバッファリングするための前記手段が、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングするための手段を備える、

ソースデバイス。

【請求項 26】

前記ソースデバイスのメモリにフレーム更新を記憶するための手段をさらに備える、請求項 25 に記載のソースデバイス。

【請求項 27】

前記メモリに記憶された前記フレーム更新のうちの前記直近の１つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の１つへのポインタのうちの１つをバッファリングするための手段をさらに備える、請求項 26 に記載のソースデバイス。

【請求項 28】

前記最小サイズのバッファが一杯か一杯でないかを判定するための手段と、

前記最小サイズのバッファが一杯でないと判定すると、または前記最小サイズのバッファが一杯であるときに前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除した後、前記最小サイズのバッファに前記フレーム更新のうちの前記直近の１つをバッファリングするための手段とをさらに備える、請求項 25 に記載のソースデバイス。

【請求項 29】

前記ソースデバイスの中央処理装置（CPU）を使用せずに、ハードウェア加速を使用して、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させるための手段をさらに備える、請求項 25 に記載のソースデバイス。

【請求項 30】

前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングするための手段と、

前記１つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化するための手段とをさらに備える、請求項 25 に記載のソースデバイス。

【請求項 31】

前記１つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信するための手段と、

新しいフレーム更新を受信せずに、ある時間期間後に、前記１つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信するための手段とをさらに備える、請求項 25 に記載のソースデバイス。

【請求項 32】

ソースデバイス内で実行されると、

ワイヤレスディスプレイ（WD）システムにおいて1つまたは複数のシンクデバイスと前記ソースデバイスで通信セッションを確立することと、

前記ソースデバイスでメディアデータのフレーム更新を取り込むことと、

前記WDシステム内で前記通信セッションを確立すると、および前記1つまたは複数のシンクデバイスへの送信用にどのフレーム更新を処理するよりも前に、処理パイプライン内の待ち時間を削減するために、最小サイズのバッファを含むように、およびハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成することであって、ここにおいて、前記最小サイズのバッファは、前記取り込まれたフレーム更新のうちの少なくとも直近の1つを保持することが可能であるサイズをそれぞれが有する少なくとも2つのバッファを備える、構成することと、

10

前記ソースデバイスの前記処理パイプラインに含まれる前記最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも前記直近の1つをバッファリングすることと、

前記1つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファからの前記フレーム更新を処理することであって、ここにおいて、前記フレーム更新がハードウェア加速を使用して前記最小サイズのバッファから取り出される、処理することと、

前記ソースデバイスのコード化ピクチャバッファに、前記1つまたは複数のシンクデバイスへの送信より前に、符号化されたフレーム更新をバッファリングすること、をプログラマブルプロセッサに行わせる命令を記憶し、

20

ここにおいて、前記命令が、実行されると、前記プログラマブルプロセッサに、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングすることをさらに行わせる、

コンピュータ可読記憶媒体。

【請求項33】

前記命令が、実行されると、前記プログラマブルプロセッサに、前記ソースデバイスのメモリにフレーム更新を記憶することを行わせる、請求項32に記載のコンピュータ可読記憶媒体。

【請求項34】

30

前記命令が、実行されると、前記プログラマブルプロセッサに、前記メモリに記憶された前記フレーム更新のうちの前記直近の1つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の1つへのポインタのうちの1つをバッファリングすることを行わせる、請求項33に記載のコンピュータ可読記憶媒体。

【請求項35】

前記命令が、実行されると、前記プログラマブルプロセッサに、

前記最小サイズのバッファが一杯か一杯でないかを判定することと、

前記最小サイズのバッファが一杯でないと判定すると、または前記最小サイズのバッファが一杯であるときに前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除した後、前記最小サイズのバッファに前記フレーム更新のうちの前記直近の1つをバッファリングすることとを行わせる、請求項32に記載のコンピュータ可読記憶媒体。

40

【請求項36】

前記命令が、実行されると、前記プログラマブルプロセッサに、前記ソースデバイスの中央処理装置（CPU）を使用せずに、ハードウェア加速を使用して、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させることを行わせる、請求項32に記載のコンピュータ可読記憶媒体。

【請求項37】

前記命令が、実行されると、前記プログラマブルプロセッサに、

50

前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、

前記1つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを行わせる、請求項32に記載のコンピュータ可読記憶媒体。

【請求項38】

前記1つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信することと、

新しいフレーム更新を受信せずに、ある時間期間後に、前記1つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとを、実行されると前記プログラマブルプロセッサに行わせる命令をさらに記憶する、請求項32に記載のコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

[0001]本出願は、各々の内容全体が参照により本明細書に組み込まれる、2012年2月28日に出願された米国仮出願第61/604,086号、2012年2月28日に出願された米国仮出願第61/604,087号、2012年2月28日に出願された米国仮出願第61/604,090号、および2012年2月28日に出願された米国仮出願第61/604,094号の利益を主張する。

【0002】

[0002]本開示は、ワイヤレスソースデバイスとワイヤレスシンク(sink、受信)デバイスとの間でデータを送信することに関する。

【背景技術】

【0003】

[0003]ワイヤレスディスプレイ(WD)システムは、ソースデバイスと1つまたは複数のシンクデバイスとを含む。ソースデバイス、およびシンクデバイスの各々は、モバイルデバイス、またはワイヤレス通信機能を有する有線デバイスのいずれかであり得る。たとえば、モバイルデバイスとして、ソースデバイスおよびシンクデバイスのうちの1つまたは複数は、モバイル電話、タブレットコンピュータ、ラップトップコンピュータ、ワイヤレス通信カード付きポータブルコンピュータ、携帯情報端末(PDA)、ワイヤレスゲームデバイス、ポータブルメディアプレーヤ、またはワイヤレス通信機能を有する他のフラッシュメモリデバイスを備える場合がある。モバイルデバイスはまた、いわゆる「スマート」フォンおよび「スマート」パッドもしくはタブレット、または他のタイプのワイヤレス通信デバイスを含む場合がある。有線デバイスとして、たとえば、ソースデバイスおよびシンクデバイスのうちの1つまたは複数は、ワイヤレス通信機能を含むテレビジョン、デスクトップコンピュータ、モニタ、プロジェクタなどを備えることができる。

【0004】

[0004]ソースデバイスは、特定の通信セッションに参加しているシンクデバイスのうちの1つまたは複数に、オーディオおよび/またはビデオデータなどのメディアデータを送る。メディアデータは、ソースデバイスのローカルディスプレイとシンクデバイスのディスプレイの各々の両方で再生することができる。より具体的には、参加しているシンクデバイスの各々は、受信されたメディアデータをそのディスプレイおよびオーディオ機器にレンダリングする。場合によっては、シンクデバイスのユーザは、タッチ入力および遠隔制御入力などのユーザ入力をシンクデバイスに加えることができる。WDシステムでは、ユーザ入力はシンクデバイスからソースデバイスに送られる。ソースデバイスは、シンクデバイスから受信されたユーザ入力を処理し、シンクデバイスに送られる以後のメディアデータにユーザ入力の効果を加える。

【発明の概要】

【0005】

[0005]一般に、本開示は、ワイヤレスディスプレイ(WD)システム内のユーザ体験を

10

20

30

40

50

改善する技法を記載する。WDシステムは、メディアデータ、たとえば、オーディオデータおよび/またはビデオデータを再生用の1つまたは複数のシンクデバイスに供給するソースデバイスを含む。本技法は、シンクデバイスでのビデオの再生品質(すなわち、滑らかさ)を改善しながら、ソースデバイスとシンクデバイスとの間のメディアデータのエンドツーエンドの待ち時間を削減することに関する。

【0006】

[0006]より具体的には、本開示の技法は、WDシステムのソースデバイスでの低遅延の画面取込みとバッファリングとを含む。たとえば、WDシステム内で通信セッションを確立すると、ソースデバイスの処理パイプラインは、待ち時間を削減するために処理ステップ間で最小サイズのバッファを含むように構成され得る。本技法は、最小サイズのバッファにメディアデータから取り込まれた少なくとも直近のフレーム更新をバッファリングすることと、最小サイズのバッファが一杯になったとき、より古いフレーム更新を削除することを含む。加えて、処理パイプラインは、ソースデバイスのパイプライン処理内の処理用にバッファからフレーム更新を取り出すようにハードウェア加速を使用するように構成され得る。ハードウェア加速の使用は、フレームレートを増大させ、待ち時間を削減させるためのソースデバイスの中央処理装置(CPU)上の処理負荷を低減することができる。本技法はまた、シンクデバイスによる時宜を得た受信を保証して、WDシステム内の待ち時間をさらに削減するために、ソースデバイスからの符号化されたフレーム更新の再送信(すなわち、重複プッシュ)を含む。

【0007】

[0007]一例では、本開示は、WDシステムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、ソースデバイスでメディアデータのフレーム更新を取り込むことと、ソースデバイスの処理パイプラインに含まれる最小サイズのバッファに取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、ハードウェア加速を使用して最小サイズのバッファ内のフレーム更新を取り出すことと、1つまたは複数のシンクデバイスへの送信用に処理パイプラインを用いて、最小サイズのバッファから取り出されたフレーム更新を処理することとを備える方法に関する。

【0008】

[0008]別の例では、本開示は、WDシステムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、メディアデータのフレーム更新を取り込むことと、処理パイプラインに含まれる最小サイズのバッファに取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、1つまたは複数のシンクデバイスへの送信用に最小サイズのバッファからのフレーム更新を処理することとを行うように構成された、1つまたは複数の処理ユニットを含む処理パイプラインと、処理パイプラインによる処理用に最小サイズのバッファからフレーム更新を取り出すように構成されたハードウェア加速器とを備えるソースデバイスに関する。

【0009】

[0009]さらなる一例では、本開示は、WDシステムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立するための手段と、メディアデータのフレーム更新を取り込むための手段と、ソースデバイスの処理パイプラインに含まれる最小サイズのバッファに取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングするための手段と、ハードウェア加速を使用して最小サイズのバッファからフレーム更新を取り出すための手段と、1つまたは複数のシンクデバイスへの送信用に処理パイプラインを用いて、最小サイズのバッファから取り出されたフレーム更新を処理するための手段とを備えるソースデバイスに関する。

【0010】

[0010]別の例では、本開示は、ソースデバイス内で実行されると、WDシステムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、ソースデバイスでメディアデータのフレーム更新を取り込むことと、ソースデバイスの

処理パイプラインに含まれる最小サイズのバッファに取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、1つまたは複数のシンクデバイスへの送信用に処理パイプラインを用いて、最小サイズのバッファからのフレーム更新を処理することであって、ここにおいて、フレーム更新がハードウェア加速を使用して最小サイズのバッファから取り出される、処理することと、をプログラマブルプロセッサに行わせる命令を備える、コンピュータ可読媒体に関する。

【0011】

[0011]本開示の1つまたは複数の例の詳細が、添付の図面および以下の説明に記載される。他の特徴、目的、および利点は、その説明および図面、ならびに特許請求の範囲から明らかになる。

【図面の簡単な説明】

【0012】

【図1】[0012]本開示の技法をサポートして、ビデオの再生品質を改善しながらソースデバイスとシンクデバイスとの間のエンドツーエンドの待ち時間を削減することが可能なソースデバイスとシンクデバイスとを含むワイヤレスディスプレイ(WD)システムの例を示すブロック図。

【図2】[0013]本開示の技法を実装できるWDシステム内のソースデバイスの例を示すブロック図。

【図3】[0014]本開示の技法を実装できるWDシステム内のシンクデバイスの例を示すブロック図。

【図4】[0015]本開示の技法を実装できる送信機システムと受信機システムとを示すブロック図。

【図5】[0016]本開示の技法をサポートして、ソースデバイスの処理パイプライン内の待ち時間を削減することが可能なソースデバイスの例を示すブロック図。

【図6】[0017]本開示の技法をサポートして、シンクデバイスの処理パイプライン内の待ち時間を削減し、シンクデバイスでのビデオ再生を改善することが可能なシンクデバイスの例を示すブロック図。

【図7】[0018]シンクデバイスで取得されたユーザ入力データおよび/またはフィードバックデータをソースデバイスに配信するために使用され得る例示的なデータパケットを示す概念図。

【図8】[0019]処理パイプライン内のメディアデータの低遅延のフレーム取込みとバッファリングとをサポートすることが可能なソースデバイスの例示的な動作を示すフローチャート。

【図9】[0020]処理パイプライン内のカスタマイズされたビデオ再生をサポートすることが可能なシンクデバイスの例示的な動作を示すフローチャート。

【図10】[0021]処理パイプライン内のメディアデータのアプリケーション認識に基づいて、カスタマイズされたバッファリングをサポートすることが可能なシンクデバイスの例示的な動作を示すフローチャート。

【図11】[0022]WDシステム内のオーディオデータの優先トランスポートをサポートすることが可能なソースデバイスおよびシンクデバイスの例示的な動作を示すフローチャート。

【発明を実施するための形態】

【0013】

[0023]本開示では、ワイヤレスディスプレイ(WD)システム内のユーザ体験を改善する技法が記載される。WDシステムは、メディアデータ、たとえば、オーディオデータおよび/またはビデオデータを再生用の1つまたは複数のシンクデバイスに供給するソースデバイスを含む。本技法は、シンクデバイスでのビデオの再生品質(すなわち、滑らかさ)を改善しながら、ソースデバイスとシンクデバイスとの間のメディアデータのエンドツーエンドの待ち時間を削減することに関する。

【0014】

[0024]一例では、本技法は、WDシステムのソースデバイスでの低遅延の画面取込みとバッファリングとを含む。たとえば、WDシステム内で通信セッションを確立すると、パイプラインマネージャは、待ち時間を削減するために処理ステップ間で最小サイズのバッファを含むように、ソースデバイスの処理パイプラインを構成することができる。次いで、ソースデバイスは、メディアデータの少なくとも直近のフレーム更新を最小サイズのバッファにバッファリングし、最小サイズのバッファが一杯になったとき、より古いフレーム更新を削除する。加えて、処理パイプラインは、ソースデバイスのパイプライン処理を用いた処理用にバッファからフレーム更新を取り出すようにハードウェア加速を使用するように構成され得る。ハードウェア加速の使用は、フレームレートを増大させ、待ち時間を削減させるためのソースデバイスの中央処理装置(CPU)上の処理負荷を低減することができる。ソースデバイスはまた、シンクデバイスによる時宜を得た受信を保証して、WDシステム内の待ち時間をさらに削減するために、符号化されたフレーム更新を再送信する(すなわち、重複プッシュを実行する)ことができる。

10

【0015】

[0025]別の例では、本技法は、ソースデバイスから受信されたメディアデータのタイプに基づく、WDシステムのシンクデバイスでのカスタマイズされた再生を含む。メディアデータがビデオデータのみを含み、オーディオデータを含まない場合、シンクデバイスの処理パイプラインに含まれるレンダラは、ビデオデータの加速されたレンダリングを実行するように構成される。たとえば、メディアデータがオーディオデータを含まないことを検出すると、パイプラインマネージャは、シンクデバイスの処理パイプラインに含まれるレンダラでの同期を無効にして、存在しないオーディオデータとの同期を待たずに、レンダラがビデオデータをレンダリングすることを可能にすることができる。別の例として、メディアデータがビデオデータとオーディオデータの両方を含むことを検出すると、パイプラインマネージャは、オーディオレンダリング開始タイマを削減することができ、その結果、レンダラは削減された開始タイマに従って、同期されたオーディオデータとビデオデータとをレンダリングすることができる。加えて、パイプラインマネージャは、セットアップ時間に起因する待ち時間を削減するために、通信セッションの能力交渉期間の間に交換されたストリームヘッダ情報に基づいて、ソースデバイスからメディアデータを受信する前に、シンクデバイス内の処理パイプラインを構成することができる。

20

【0016】

30

[0026]さらなる一例では、本技法は、ソースデバイスから受信されたメディアデータについてのアプリケーション認識に基づく、WDシステムのシンクデバイスでのカスタマイズされたバッファリングを含む。シンクデバイスは、メディアデータ用のアプリケーションのタイプを知り、パイプラインマネージャは、シンクデバイスの処理パイプライン内のバッファのサイズを調整して、アプリケーションのタイプについての滑らかさと待ち時間との間の適切なバランスを実現する。場合によっては、メディアデータ用のアプリケーションのタイプは、ソースデバイスで検出され得、シンクデバイスは、ソースデバイスから受信された情報に基づいて、アプリケーションのタイプを知ることができる。他の場合、シンクデバイスは、アプリケーションのタイプ自体を検出することによって、メディアデータ用のアプリケーションのタイプを知ることができる。たとえば、メディアデータが、再生の品質または滑らかさがシンクデバイスで最も優先度が高く、上述された低遅延技法は目に見えるジッタをもたらす場合がある、ビデオ再生アプリケーション用であるとき、バッファサイズは増大されて、ビデオ再生アプリケーション用内のメディアデータの滑らかさを増大させる。反対に、メディアデータが、低遅延がシンクデバイスで最も優先度が高い、ユーザインターフェース(UI)アプリケーションまたはゲームアプリケーション用であるとき、バッファサイズは減少されて、UIアプリケーションまたはゲームアプリケーションについての待ち時間を削減する。

40

【0017】

[0027]本技法はまた、WDシステム内のソースデバイスとシンクデバイスとの間で、オーディオデータのトランスポートをビデオデータのトランスポートより優先することを含

50

む。ビデオデータパケットのトランスポートは、関連するオーディオデータパケットのトランスポートに結び付けられ、その結果、すべてのオーディオパケットがシンクデバイスに到達することを保証するので、削除されたパケットの受信を待つシンクデバイスでの待ち時間が削減される。パイプラインマネージャは、ソースデバイス内のビデオパイプライン経路よりも多くのバッファリングを含むように、オーディオパイプライン経路を構成することができる。加えて、ソースデバイスでのワイヤレスモデムソケットは、ビデオパイプライン経路よりもオーディオパイプライン経路に、より高い優先度のトランスポートキューを与えることができる。さらなるバッファリングにより、ソースデバイスで削除されるオーディオパケットがより少なくなることが保証される。より高い優先度のトランスポートキューにより、ビデオパイプライン経路内の遅延または失速(stall)を回避するために、オーディオパケットが対応するビデオパケットより前にソースデバイスでトランスポート用にキューイングされることが保証される。加えて、シンクデバイスは、ソースデバイスに通信チャネルのトランスポート状態を記述するフィードバック情報を供給することができ、パイプラインマネージャは、フィードバック情報に基づいてソースデバイスの処理パイプラインを修正することができる。

10

【0018】

[0028]図1は、本開示の技法をサポートして、ビデオの再生品質を改善しながらソースデバイス120とシンクデバイス160との間のエンドツーエンドの待ち時間を削減することが可能なソースデバイス120とシンクデバイス160とを含むワイヤレスディスプレイ(WD)システム100の例を示すブロック図である。図1に示されるように、WD

20

システム100は、通信チャネル150を介してシンクデバイス160と通信するソースデバイス120を含む。

【0019】

[0029]ソースデバイス120は、オーディオおよび/またはビデオ(A/V)メディアデータ121を記憶するメモリと、ディスプレイ122と、スピーカ123と、(エンコーダ124とも呼ばれる)オーディオおよび/またはビデオ(A/V)エンコーダ124と、オーディオおよび/またはビデオ(A/V)制御モジュール125と、送信機/受信機(TX/RX)ユニット126とを含むことができる。シンクデバイス160は、ディスプレイ162と、スピーカ163と、(デコーダ164とも呼ばれる)オーディオおよび/またはビデオ(A/V)デコーダ164と、送信機/受信機ユニット166と、ユーザ入力(UI)デバイス167と、ユーザ入力処理モジュール(UIPM)168とを含むことができる。図示された構成要素は、WDシステム100の1つの例示的な構成をなすにすぎない。他の構成は、図示された構成要素よりも少ない構成要素を含む場合があるか、または図示された構成要素以外のさらなる構成要素を含む場合がある。

30

【0020】

[0030]図1の例では、ソースデバイス120は、A/Vメディアデータ121のビデオ部分をディスプレイ122に表示することができ、A/Vメディアデータ121のオーディオ部分をスピーカ123に出力することができる。A/Vメディアデータ121は、ソースデバイス120にローカルに記憶されるか、ファイルサーバ、ハードドライブ、外部メモリ、ブルーレイ(登録商標)ディスク、DVD、もしくは他の物理記憶媒体などの外部記憶媒体からアクセスされるか、またはインターネットなどのネットワーク接続を介してソースデバイス120にストリーミングすることができる。場合によっては、A/Vメディアデータ121は、ソースデバイス120のカメラとマイクロフォンとを介して、リアルタイムにキャプチャすることができる。A/Vメディアデータ121は、映画、テレビ番組、または音楽などのマルチメディアコンテンツを含むことができるが、ソースデバイス120によって生成されたリアルタイムコンテンツも含むことができる。そのようなリアルタイムコンテンツは、たとえば、ソースデバイス120で動作しているアプリケーションによって生成されるか、または、たとえば、ビデオテレフォニーセッションの一部としてキャプチャされたビデオデータであり得る。場合によっては、そのようなリアルタイムコンテンツは、ユーザが選択する場合に利用可能なユーザ入力オプションのビデオフ

40

50

フレームを含むことができる。場合によっては、A/Vメディアデータ121は、ビデオのフレーム上にオーバーレイされたユーザ入力オプションを有する映画またはTV番組のビデオフレームなどの、異なるタイプのコンテンツの組合せであるビデオフレームを含むことができる。

【0021】

[0031]ディスプレイ122とスピーカ123とを介してローカルにA/Vメディアデータ121をレンダリングすることに加えて、ソースデバイス120のA/Vエンコーダ124は、A/Vメディアデータ121を符号化することができ、送信機/受信機ユニット126は、通信チャネル150を介してシンクデバイス160に符号化されたデータを送信することができる。シンクデバイス160の送信機/受信機ユニット166は、符号化されたデータを受信し、A/Vデコーダ164は、符号化されたデータを復号し、ディスプレイ162とスピーカ163とを介して復号されたデータを出力する。このようにして、ディスプレイ122およびスピーカ123によってレンダリングされているオーディオおよびビデオデータは、同時にディスプレイ162およびスピーカ163によってレンダリングすることができる。オーディオデータおよびビデオデータはフレームに構成することができ、オーディオフレームは、レンダリングされるときにビデオフレームと時刻同期することができる。

【0022】

[0032]A/Vエンコーダ124およびA/Vデコーダ164は、代替的にMPEG-4, Part 10, Advanced Video Coding (AVC) と呼ばれるITU-T H.264規格、または新興の高効率ビデオコーディング(HEVC)規格などの任意の数のオーディオおよびビデオ圧縮規格を実装することができる。多くの他のタイプのプロプライエタリな、または規格化された圧縮技法も使用することができる。概して、A/Vデコーダ164は、A/Vエンコーダ124の逆のコーディング演算を実行するように構成される。図1には示されないが、いくつかの態様では、A/Vエンコーダ124およびA/Vデコーダ164は、各々オーディオのエンコーダおよびデコーダと統合することができ、共通のデータストリームまたは別々のデータストリーム内のオーディオとビデオの両方の符号化を処理するために、適切なMUX-DEMUXユニット、または他のハードウェアとソフトウェアとを含むことができる。

【0023】

[0033]A/Vエンコーダ124は、上述のようなビデオ圧縮規格を実装することに加えて、他の符号化機能も実行することができる。たとえば、A/Vエンコーダ124は、A/Vデータメディア121がシンクデバイス160に送信されるより前に、A/Vメディアデータ121に様々なタイプのメタデータを追加することができる。場合によっては、A/Vメディアデータ121は、符号化された形態でソースデバイス120に記憶されるかまたはソースデバイス120で受信され、したがってA/Vエンコーダ124によるさらなる圧縮を必要としない場合がある。

【0024】

[0034]図1は、オーディオペイロードデータとビデオペイロードデータとを別々に搬送する通信チャネル150を示すが、場合によっては、ビデオペイロードデータとオーディオペイロードデータは共通のデータストリームの一部であり得ることを理解されたい。適用可能な場合、MUX-DEMUXユニットは、ITU H.223マルチプレクサプロトコル、またはユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠することができる。A/Vエンコーダ124およびA/Vデコーダ164は、各々1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組合せとして実装することができる。A/Vエンコーダ124およびA/Vデコーダ164の各々は、1つまたは複数のエンコーダまたはデコーダに含まれ、それらのいずれかは、複合エンコーダ/デコーダ(コーデック)の一部として統合することができる。したがって、ソー

スデバイス 120 およびシンクデバイス 160 の各々は、本開示の技法のうちの 1 つまたは複数を実行するように構成された専用の機械を備えることができる。

【0025】

[0035] ディスプレイ 122 およびディスプレイ 162 は、陰極線管 (CRT)、液晶ディスプレイ (LCD)、プラズマディスプレイ、発光ダイオード (LED) ディスプレイ、有機発光ダイオード (OLED) ディスプレイ、または別のタイプのディスプレイデバイスなどの様々なビデオ出力デバイスのうちのいずれも備えることができる。これらまたは他の例では、ディスプレイ 122 および 162 は、各々発光型ディスプレイまたは透過型ディスプレイであり得る。ディスプレイ 122 およびディスプレイ 162 はまた、それらが同時に入力デバイスとディスプレイデバイスの両方であるようなタッチディスプレイであり得る。そのようなタッチディスプレイは、ユーザがそれぞれのデバイスにユーザ入力を与えることを可能にする、静電容量式、抵抗式、または他のタイプのタッチパネルであり得る。

10

【0026】

[0036] スピーカ 123 は、ヘッドフォン、シングルスピーカシステム、マルチスピーカシステム、またはサラウンドサウンドシステムなどの様々なオーディオ出力デバイスのうちのいずれも備えることができる。加えて、ディスプレイ 122 およびスピーカ 123 はソースデバイス 120 の一部として示され、ディスプレイ 162 およびスピーカ 163 はシンクデバイス 160 の一部として示されているが、ソースデバイス 120 およびシンクデバイス 160 は、実際はそれらのデバイスのシステムであり得る。一例として、ディスプレイ 162 はテレビジョンであり得るし、スピーカ 163 はサラウンドサウンドシステムであり得るし、デコーダ 164 は有線またはワイヤレスのいずれかでディスプレイ 162 とスピーカ 163 とに接続された外部ボックスの一部であり得る。他の例では、シンクデバイス 160 は、タブレットコンピュータまたはスマートフォンなどの単一デバイスであり得る。さらに他の場合では、ソースデバイス 120 およびシンクデバイス 160 は、同様のデバイス、たとえば両方ともスマートフォン、タブレットコンピュータなどである。この場合、一方のデバイスがソースとして動作することができ、他方がシンクとして動作することができる。これらの役割は、以後の通信セッションでは反転される場合さえある。さらに他の場合、ソースデバイスは、スマートフォン、ラップトップコンピュータまたはタブレットコンピュータなどのモバイルデバイスを備えることができ、シンクデバイスは、(たとえば、AC 電源コードを有する) より固定されたデバイスを備えることができ、その場合、ソースデバイスは、シンクデバイスを介して大群衆に提示用のオーディオおよびビデオデータを配信することができる。

20

30

【0027】

[0037] 送信機 / 受信機ユニット 126 および送信機 / 受信機ユニット 166 は、各々様々なミキサ、フィルタ、増幅器、および信号変調用に設計された他の構成要素、ならびに、1 つまたは複数のアンテナ、およびデータを送受信するために設計された他の構成要素を含むことができる。通信チャネル 150 は、概して、ソースデバイス 120 からシンクデバイス 160 にビデオデータを送信するのに適した任意の通信媒体、または様々な通信媒体の集合を表す。通信チャネル 150 は、通常、Wi-Fi (登録商標)、Bluetooth (登録商標) などと同様の比較的短距離の通信チャネルである。しかしながら、通信チャネル 150 は、この点において必ずしも限定されずとは限らず、無線周波数 (RF) スペクトルまたは 1 つもしくは複数の物理伝送線路などの任意のワイヤレスまたは有線の通信媒体、あるいはワイヤレス媒体と有線媒体との任意の組合せを備えることができる。他の例では、通信チャネル 150 は、有線もしくはワイヤレスのローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどの、パケットベースネットワークの一部を形成する場合さえある。加えて、通信チャネル 150 は、ピアツーピアリンクを作成するために、ソースデバイス 120 およびシンクデバイス 160 によって使用することができる。

40

【0028】

50

[0038]ソースデバイス120およびシンクデバイス160は、たとえば、リアルタイムストリーミングプロトコル(RTSP)制御メッセージを使用して、能力ネゴシエーションに従って通信セッションを確立することができる。次いで、ソースデバイス120およびシンクデバイス160は、IEEE802.11規格ファミリーからの規格などの通信プロトコルを使用して、通信チャネル150を介して通信することができる。ソースデバイス120およびシンクデバイス160は、たとえば、Wi-Fi Direct(WFD)規格に従って通信することができ、その結果、ソースデバイス120およびシンクデバイス160は、ワイヤレスアクセスポイントまたはいわゆるホットスポットなどの仲介者を使用せずに互いに直接通信する。ソースデバイス120およびシンクデバイス160はまた、Tunneled Direct Link Setup(TDLS)を確立して、ネットワーク輻輳を回避または低減することができる。WFDおよびTDLSは、比較的短距離の通信セッションをセットアップすることを目的とする。この文脈では、比較的短距離は、たとえば、約70メートル未満を指す場合があるが、雑音の多いまたは遮るもののある環境では、デバイス間の距離は、約35メートル未満、または約20メートル未満など、さらに短い場合がある。

【0029】

[0039]本開示の技法は、時々WFDに関して記載される場合があるが、これらの技法の態様は他の通信プロトコルにも適合できると考えられる。限定ではなく例として、ソースデバイス120とシンクデバイスとの間のワイヤレス通信は、直交周波数分割多重化(OFDM)技法を利用することができる。限定はしないが、時分割多元接続(TDMA)、周波数分割多元接続(FDMA)、符号分割多元接続(CDMA)、またはOFDM、FDMA、TDMAおよび/もしくはCDMAの任意の組合せを含む、多種多様な他のワイヤレス通信技法も使用することができる。

【0030】

[0040]ソースデバイス120から受信されたデータを復号しレンダリングすることに加えて、シンクデバイス160は、ユーザ入力デバイス167からユーザ入力を受信することもできる。ユーザ入力デバイス167は、たとえば、キーボード、マウス、トラックボールもしくはトラックパッド、タッチスクリーン、ボイスコマンド認識モジュール、または任意の他のそのようなユーザ入力デバイスであり得る。UIPM168は、ユーザ入力デバイス167によって受信されたユーザ入力コマンドを、ソースデバイス120が解釈することが可能なデータパケット構造にフォーマットする。そのようなデータパケットは、送信機/受信機166により、通信チャネル150を介してソースデバイス120に送信される。送信機/受信機ユニット126はデータパケットを受信し、A/V制御モジュール125はデータパケットを解析して、ユーザ入力デバイス167によって受信されたユーザ入力コマンドを解釈する。データパケット内で受信されたコマンドに基づいて、A/V制御モジュール125は、符号化され送信されているコンテンツを変更することができる。このようにして、シンクデバイス160のユーザは、リモートで、かつソースデバイス120と直接対話することなしに、ソースデバイス120によって送信されているオーディオペイロードデータとビデオペイロードデータとを制御することができる。

【0031】

[0041]加えて、シンクデバイス160のユーザは、ソースデバイス120上のアプリケーションを起動し、制御することが可能である場合がある。たとえば、シンクデバイス160のユーザは、ソースデバイス120に記憶された写真編集アプリケーションを起動し、そのアプリケーションを使用して、ソースデバイス120にローカルに記憶された写真を編集することが可能である場合がある。シンクデバイス160は、その写真が実際はソースデバイス120上で編集されているが、その写真がシンクデバイス160上でローカルに編集されているように見え感じる、ユーザエクスペリエンスをユーザにもたらしすることができる。そのような構成を使用して、デバイスユーザは、いくつかのデバイスとともに使用するために1つのデバイスの能力を活用することが可能である場合がある。たとえば、ソースデバイス120は、大量のメモリとハイエンド処理能力とを有するスマートフォ

ンを備え得る。しかしながら、映画を見るとき、ユーザは、より大きいディスプレイスクリーンを有するデバイス上で映画を見ることを望む場合があり、その場合、シンクデバイス160は、タブレットコンピュータまたはさらに大きいディスプレイデバイスもしくはテレビジョンであり得る。電子メールを送ることまたは電子メールに回答することを行いたいとき、ユーザは、物理キーボードを有するデバイスを使用することを望む場合があり、その場合、シンクデバイス160はラップトップであり得る。どちらの場合でも、ユーザはシンクデバイスと対話していても、処理の大半は依然としてソースデバイス120によって実行することができる。ソースデバイスおよびシンクデバイスは、任意の所与のセッションにおいて、デバイスの能力をネゴシエーションおよび/または識別することにより、双方向対話を容易にすることができる。

10

【0032】

[0042]いくつかの構成では、A/V制御モジュール125は、ソースデバイス120のオペレーティングシステムによって実行されるオペレーティングシステムプロセスを備え得る。しかしながら、他の構成では、A/V制御モジュール125は、ソースデバイス120上で動作しているアプリケーションのソフトウェアプロセスを備え得る。そのような構成では、ユーザ入力コマンドはソフトウェアプロセスによって解釈することができ、その結果、シンクデバイス160のユーザは、ソースデバイス120上で動作しているオペレーティングシステムとは対照的に、ソースデバイス120上で動作しているアプリケーションと直接対話している。オペレーティングシステムとは対照的にアプリケーションと直接対話することにより、シンクデバイス160のユーザは、ソースデバイス120のオペレーティングシステムに対してネイティブでないコマンドのライブラリにアクセスすることができる。加えて、アプリケーションと直接対話することにより、コマンドは、異なるプラットフォーム上で動作しているデバイスによってより容易に送信され、処理されることが可能になり得る。

20

【0033】

[0043]シンクデバイス160で加えられたユーザ入力は、通信チャネル150を介してソースデバイス120に送り返すことができる。一例では、ユーザインターフェースバックチャネル(UIBC)とも呼ばれる逆方向チャネルアーキテクチャは、シンクデバイス160で加えられたユーザ入力を、シンクデバイス160がソースデバイス120に送信することを可能にするために実装することができる。逆方向チャネルアーキテクチャは、ユーザ入力をトランスポートするための上位レイヤメッセージと、シンクデバイス160およびソースデバイス120でユーザインターフェース能力をネゴシエーションするための下位レイヤフレームとを含むことができる。UIBCは、シンクデバイス160とソースデバイス120との間のインターネットプロトコル(IP)トランスポートレイヤ上に常駐することができる。このようにして、UIBCは、開放型システム間相互接続(OSI)通信モデルにおいてトランスポートレイヤより上位であり得る。ユーザ入力データを含んでいるデータパケットの信頼できる送信と順次配信とを促進するために、UIBCは、伝送制御プロトコル/インターネットプロトコル(TCP/IP)またはユーザデータグラムプロトコル(UDP)などの他のパケットベース通信プロトコルの上で動作するように構成することができる。UDPおよびTCPは、OSIレイヤアーキテクチャにおいて並列に動作することができる。TCP/IPにより、シンクデバイス160およびソースデバイス120は、パケットロスの場合の再送信技法を実装することが可能になり得る。

30

40

【0034】

[0044]UIBCは、クロスプラットフォームユーザ入力データを含む、様々なタイプのユーザ入力データをトランスポートするように設計することができる。たとえば、ソースデバイス120は、iOS(登録商標)オペレーティングシステムを実行することができ、一方シンクデバイス160は、Android(登録商標)またはWindows(登録商標)などの別のオペレーティングシステムを実行する。プラットフォームにかかわらず、UIPM168は、受信されたユーザ入力を、A/V制御モジュール125が理解で

50

きる形態にカプセル化することができる。ソースデバイスとシンクデバイスが異なるプラットフォーム上で動作するかどうかにかかわらず、多くの異なるタイプのソースデバイスおよびシンクデバイスがプロトコルを活用することが可能になるように、いくつかの異なるタイプのユーザ入力フォーマットをUIBCによってサポートすることができる。一般入力フォーマットを定義でき、プラットフォーム固有入力フォーマットを両方ともサポートでき、したがって、ユーザ入力がUIBCによってソースデバイス120とシンクデバイス160との間で通信できる方式でフレキシビリティが与えられる。

【0035】

[0045]本開示の技法は、WDシステム100内のユーザ体験を改善することができる。WDシステム100のユーザ向けの典型的な使用事例には、ソースデバイス120上で動作するユーザインターフェースアプリケーションを、逆ヒューマンインターフェースデバイス(HID)フィードバックでシンクデバイス160のより大きいディスプレイ画面に表示すること、ソースデバイス120上で動作するゲームアプリケーションをシンクデバイス160で表示すること、およびソースデバイス120上で動作するビデオ再生アプリケーションをシンクデバイス160で表示することが含まれる。これらの使用事例の各々について、ソースデバイス120は、全フレームバッファのコンテンツとフレーム更新エンドツーエンドとを、シンクデバイス160のユーザへの表示用にシンクデバイス160に送信することができる。ソースデバイス120の送信機/受信機126は、リアルタイムトランスポートプロトコル(RTP)/ユーザデータグラムプロトコル(UDP)または伝送制御プロトコル(TCP)を使用して、ソースデバイス160の送信機/受信機166に符号化されたビデオフレームをトランスポートすることができる。

【0036】

[0046]WDシステム100が高精細度マルチメディアインターフェース(HDMI(登録商標))または他のタイプのディスプレイケーブルと同様のユーザ体験を提供するために、WDシステム100は、約80ミリ秒(ms)未満のエンドツーエンドの待ち時間を実現する必要がある。しかしながら、ビデオ再生アプリケーションの使用事例の場合、WDシステム100は、過剰なさらなる待ち時間をもたらさずに、シンクデバイス160でのビデオ再生で高い品質(すなわち、滑らかさ)も実現する必要がある。本開示は、改善されたユーザ体験のための上述された待ち時間および品質のニーズを実現するために、個々に、または組合せてWDシステム100に適用され得るいくつかの技法を記載する。

【0037】

[0047]本開示の技法によれば、処理パイプラインは、シンクデバイス160でのビデオの再生品質を改善しながら、ソースデバイス120とシンクデバイス160との間で送信されるA/Vメディアデータ121のエンドツーエンドの待ち時間を削減するように、ソースデバイス120とシンクデバイス160の両方で構成され得る。一例では、本技法は、ソースデバイス120での低遅延の画面取込みとバッファリングとを含む。たとえば、WDシステム100内で通信セッションを確立すると、パイプラインマネージャは、待ち時間を削減するために処理ステップ間でA/Vメディアデータ121の少なくとも直近のフレーム更新を保持することが可能な最小サイズのバッファを含むように、ソースデバイス120の処理パイプラインを構成することができる。加えて、ソースデバイス120の処理パイプラインは、パイプライン処理を用いた処理用にバッファからフレーム更新を取り出すようにハードウェア加速を使用するように構成され得る。

【0038】

[0048]別の例では、本技法は、ソースデバイス120から受信されたA/Vメディアデータ121のタイプに基づく、シンクデバイス160でのカスタマイズされた再生を含む。A/Vメディアデータ121がビデオデータのみを含み、オーディオデータを含まない場合、シンクデバイス160の処理パイプラインに含まれるレンダラは、ビデオデータの加速されたレンダリングを実行するように構成され得る。A/Vメディアデータ121がビデオデータとオーディオデータの両方を含む場合、パイプラインマネージャは、オーディオレンダリング開始タイマを削減することができ、その結果、レンダラは削減された開

始タイマに従って、同期されたオーディオデータとビデオデータとをレンダリングすることができる。加えて、パイプラインマネージャは、セットアップ時間に起因する待ち時間を削減するために通信セッションの能力交渉期間の間に交換されたストリームヘッダ情報に基づいて、ソースデバイス120からA/Vメディアデータ121を受信する前に、シンクデバイス160内の処理パイプラインを構成することができる。

【0039】

[0049]さらなる一例では、本技法は、ソースデバイス120から受信されたA/Vメディアデータ121についてのアプリケーション認識に基づく、シンクデバイス160でのカスタマイズされたバッファリングを含む。シンクデバイス160は、A/Vメディアデータ121用のアプリケーションのタイプを知り、パイプラインマネージャは、シンクデバイス160の処理パイプライン内のバッファのサイズを調整して、アプリケーションのタイプについての滑らかさと待ち時間との間の適切なバランスを実現する。たとえば、A/Vメディアデータ121がビデオ再生アプリケーション用であるとき、バッファサイズは増大されて、ビデオ再生アプリケーションについての滑らかさを増大させる。反対に、A/Vメディアデータ12がユーザインターフェース(UI)アプリケーションまたはゲームアプリケーション用であるとき、バッファサイズは減少されて、UIアプリケーションまたはゲームアプリケーションについての待ち時間を削減する。

【0040】

[0050]本開示の技法はまた、ソースデバイス120とシンクデバイス160との間で、オーディオデータのトランスポートをビデオデータのトランスポートより優先することを含む。ビデオデータパケットのトランスポートは、関連するオーディオデータパケットのトランスポートに結び付けられ、その結果、すべてのオーディオパケットがシンクデバイスに到達することを保証するので、削除されたパケットの受信を待つシンクデバイスでの待ち時間が削減される。パイプラインマネージャは、ソースデバイス120内のビデオパイプライン経路よりも多くのバッファリングを含むように、オーディオパイプライン経路を構成することができる。加えて、ソースデバイス120でのワイヤレスモデムソケットは、ビデオパイプライン経路よりもオーディオパイプライン経路に、より高い優先度のトランスポートキューを与えることができる。加えて、シンクデバイス160は、UIBCを介して通信チャネル150のトランスポート状態を記述するフィードバック情報をソースデバイス120に供給することができ、パイプラインマネージャは、フィードバック情報に基づいてソースデバイス120の処理パイプラインを修正することができる。

【0041】

[0051]図1の例では、ソースデバイス120は、スマートフォン、タブレットコンピュータ、ラップトップコンピュータ、デスクトップコンピュータ、Wi-Fi対応テレビジョン、またはオーディオデータおよびビデオデータを送信することが可能な任意の他のデバイスを備えることができる。シンクデバイス160は、同様に、スマートフォン、タブレットコンピュータ、ラップトップコンピュータ、デスクトップコンピュータ、Wi-Fi対応テレビジョン、またはオーディオデータおよびビデオデータを受信し、ユーザ入力データを受信することが可能な任意の他のデバイスを備えることができる。場合によっては、シンクデバイス160は、ディスプレイ162、スピーカ163、UIデバイス167、およびA/Vエンコーダ164が、別個であるが相互動作可能なデバイスのすべての部品であるような、デバイスのシステムを含むことができる。ソースデバイス120は、同様に、単一のデバイスではなく、デバイスのシステムであり得る。

【0042】

[0052]本開示では、ソースデバイスという用語は、一般に、A/Vデータを送信しているデバイスを指すために使用され、シンクデバイスという用語は、一般に、ソースデバイスからA/Vデータを受信しているデバイスを指すために使用される。多くの場合、ソースデバイス120とシンクデバイス160は、一方のデバイスがソースとして動作し、他方がシンクとして動作する、同様または同等のデバイスであり得る。その上、異なる通信セッションではこれらのロールが逆になる場合がある。したがって、1つの通信セッショ

10

20

30

40

50

ン内のシンクデバイスは、次の通信セッションではソースデバイスになる場合があり、その逆もあり得る。

【 0 0 4 3 】

[0053]いくつかの例では、WDシステム100は、シンクデバイス160に加えて1つまたは複数のシンクデバイスを含む場合がある。シンクデバイス160と同様に、追加のシンクデバイスは、ソースデバイス120からA/Vデータを受信し、かつ確立されたUIBCを介してソースデバイス120にユーザコマンドを送信することができる。いくつかの構成では、複数のシンクデバイスは互いに独立して動作することができ、ソースデバイス120でのA/Vデータ出力は、シンクデバイス160および追加のシンクデバイスのうちの1つまたは複数で同時に出力され得る。代替の構成では、シンクデバイス160は主要なシンクデバイスであり得、追加のシンクデバイスのうちの1つまたは複数は二次的なシンクデバイスであり得る。そのような例示的な構成では、シンクデバイス160および追加のシンクデバイスのうちの1つが結合される場合があり、シンクデバイス160はビデオデータを表示することができ、一方追加のシンクデバイスは対応するオーディオデータを出力する。加えて、いくつかの構成では、シンクデバイス160は送信されたビデオデータのみを出力することができ、一方追加のシンクデバイスは送信されたオーディオデータを出力する。

10

【 0 0 4 4 】

[0054]図2は、本開示の技法を実装できるWDシステム内のソースデバイス220の一例を示すブロック図である。ソースデバイス220は、図1内のソースデバイス120と同様のデバイスであり得るし、ソースデバイス120と同じ方式で動作することができる。ソースデバイス220は、ローカルディスプレイ222と、スピーカ223と、プロセッサ231と、ディスプレイプロセッサ235と、オーディオプロセッサ236と、メモリ232と、トランスポートユニット233と、ワイヤレスモデム234とを含む。図2で示されたように、ソースデバイス220は、トランスポート、記憶、および表示のためにA/Vメディアデータを符号化および/または復号する1つまたは複数のプロセッサ(すなわち、プロセッサ231、ディスプレイプロセッサ235およびオーディオプロセッサ236)を含むことができる。A/Vメディアデータは、たとえばメモリ232で記憶することができる。メモリ232は、A/Vファイル全体を記憶することができるか、または、たとえば別のデバイスもしくはソースからストリームされたA/Vファイルの一部を記憶するだけのより小さいバッファを備えることができる。

20

30

【 0 0 4 5 】

[0055]トランスポートユニット233は、符号化されたA/Vメディアデータをネットワークトランスポート用に処理することができる。たとえば、符号化されたA/Vメディアデータはプロセッサ231によって処理され、ネットワークにわたる通信のネットワークアクセスレイヤ(NAL)ユニットに、トランスポートユニット233によってカプセル化することができる。NALユニットは、ワイヤレスモデム234により、ネットワーク接続を介してワイヤレスシンクデバイスに送ることができる。ワイヤレスモデム234は、たとえば、IEEE802.11規格ファミリーのうちの1つを実装するように構成されたWi-Fiモデムであり得る。ソースデバイス220はまた、A/Vメディアデータをローカルに処理し表示することができる。特に、ディスプレイプロセッサ235は、ローカルディスプレイ222で表示されるようにビデオデータを処理することができ、オーディオプロセッサ236は、スピーカ223での出力用にオーディオデータを処理することができる。

40

【 0 0 4 6 】

[0056]図1のソースデバイス120に関して上述されたように、ソースデバイス220はまた、シンクデバイスからユーザ入力コマンドを受信することができる。たとえば、ソースデバイス220のワイヤレスモデム234は、NALユニットなどのカプセル化されたユーザ入力データパケットを受信し、カプセル化されたデータユニットをカプセル化解除のためにトランスポートユニット233に送ることができる。トランスポートユニット

50

233はNALユニットからユーザ入力データ packets を抽出することができ、プロセッサ231はデータ packets を解析して、ユーザ入力コマンドを抽出することができる。ユーザ入力コマンドに基づいて、プロセッサ231は、ソースデバイス220によるA/Vメディアデータの処理を修正する。他の例では、ソースデバイス220は、トランスポートユニット233からユーザ入力データ packets を受信し、データ packets を解析してユーザ入力コマンドを抽出し、ユーザ入力コマンドに基づいてソースデバイス220によるA/Vメディアデータの処理を修正するようにプロセッサ231に指示する、ユーザ入力ユニットまたはユーザ入力ドライバ(図2には示されず)を含むことができる。

【0047】

[0057]図2のプロセッサ231は、一般に、限定はしないが、1つもしくは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、他の等価な集積回路もしくはディスクリット論理回路、またはそれらのいくつかの組合せを含む、多種多様なプロセッサのうちのいずれをも表す。図2のメモリ232は、限定はしないが、同期ダイナミックランダムアクセスメモリ(SDRAM)などのランダムアクセスメモリ(RAM)、読取り専用メモリ(ROM)、不揮発性ランダムアクセスメモリ(NVRAM)、電氣的消去可能プログラマブル読取り専用メモリ(EEPROM(登録商標))、フラッシュメモリなどを含む、多種多様な揮発性メモリまたは不揮発性メモリのいずれも備えることができる。メモリ232は、オーディオ/ビデオデータならびに他の種類のデータを記憶するためのコンピュータ可読記憶媒体を備えることができる。加えて、メモリ232は、本開示に記載される様々な技法を実行することの一部として、プロセッサ231によって実行される命令とプログラムコードとを記憶することができる。

【0048】

[0058]本開示の技法は、WDシステム内のソースデバイス220および1つまたは複数のシンクデバイスでのユーザ体験を改善するために、待ち時間を削減するようにソースデバイス220の処理パイプラインを構成することを含む。ソースデバイス220の処理パイプラインは、プロセッサ231および/またはトランスポートユニット233によって実行される1つまたは複数の処理ユニットを含む。本技法は、図5に示されるソースデバイス520の処理パイプラインに関して、さらに詳細に記載される。

【0049】

[0059]一例では、ソースデバイス220の処理パイプラインは、低遅延の画面取込みとメディアデータのバッファリングとを提供するように修正され得る。より具体的には、ソースデバイス220の処理パイプラインは、待ち時間を削減するためにプロセッサ231内の処理ステップ間で最小サイズのバッファを含むように構成され得る。次いで、ソースデバイス220は、メディアデータの少なくとも直近のフレーム更新を最小サイズのバッファにバッファリングし、最小サイズのバッファが一杯になったとき、より古いフレーム更新を削除する。

【0050】

[0060]ソースデバイス220の処理パイプラインはまた、処理用にバッファからフレーム更新を取り出すようにハードウェア加速を使用するように構成され得る。ハードウェア加速の使用は、ソースデバイス220の中央処理装置(CPU)上の処理負荷を低減することができる、このことは、WDシステム内のフレームレートを増大させ、待ち時間を削減する。ソースデバイス220はまた、シンクデバイスによる時宜を得た受信を保証して、WDシステム内の待ち時間をさらに削減するために、符号化されたフレーム更新を再送信する(すなわち、重複プッシュを実行する)ことができる。

【0051】

[0061]別の例として、本技法によれば、ソースデバイス220の処理パイプラインは、WDシステム内のソースデバイス220とシンクデバイスとの間で、オーディオデータのトランスポートをビデオデータのトランスポートより優先するように構成され得る。ビデオデータ packets のトランスポートは、関連するオーディオデータ packets のトランスポ

ートに結び付けられ、その結果、すべてのオーディオパケットがシンクデバイスに到達することを保証するので、削除されたパケットの受信を待つシンクデバイスでの待ち時間が削減される。

【 0 0 5 2 】

[0062]より具体的には、ソースデバイス 2 2 0 のオーディオパイプライン経路は、ビデオパイプライン経路よりも多くのバッファリングを含むように構成され得る。さらなるバッファリングにより、ソースデバイス 2 2 0 で削除されるオーディオパケットがより少なくなることが保証される。別の例として、ソースデバイス 2 2 0 の処理パイプラインは、ビデオパイプライン経路よりもオーディオパイプライン経路に、ワイヤレスモデム 2 3 4 でより高い優先度のトランスポートキューを与えるように構成され得る。より高い優先度のトランスポートキューにより、対応するオーディオパケットを待つビデオパケットによって生じるビデオパイプライン経路内の遅延または失速を回避するために、オーディオパケットが対応するビデオパケットより前にソースデバイス 2 2 0 でトランスポート用にキューイングされることが保証される。加えて、ソースデバイス 2 2 0 のワイヤレスモデム 2 3 4 は、通信チャネルのトランスポート状態を記述するフィードバック情報を、WDシステム内のシンクデバイスから受信することができる。それに応答して、ソースデバイス 2 2 0 は、フィードバック情報に基づいて処理パイプラインを修正することができる。

【 0 0 5 3 】

[0063]上述された技法のうちの 1 つまたは複数をソースデバイス 2 2 0 に適用すると、WDシステム内のエンドツーエンドの待ち時間が削減され、WDシステム内のソースデバイス 2 2 0 とシンクデバイスの両方でユーザ体験が改善され得る。

【 0 0 5 4 】

[0064]図 3 は、本開示の技法を実装できるWDシステム内のシンクデバイス 3 6 0 の例を示すブロック図である。シンクデバイス 3 6 0 は、図 1 のシンクデバイス 1 6 0 と同様のデバイスであり得、シンクデバイス 1 6 0 と同じ方式で動作することができる。シンクデバイス 3 6 0 は、プロセッサ 3 3 1 と、メモリ 3 3 2 と、トランスポートユニット 3 3 3 と、ワイヤレスモデム 3 3 4 と、ディスプレイプロセッサ 3 3 5 と、ローカルディスプレイ 3 6 2 と、オーディオプロセッサ 3 3 6 と、スピーカ 3 6 3 と、ユーザ入力インターフェース 3 7 6 とを含む。

【 0 0 5 5 】

[0065]シンクデバイス 3 6 0 は、ソースデバイスから送られたカプセル化されたデータユニットをワイヤレスモデム 3 3 4 で受信する。ワイヤレスモデム 3 3 4 は、たとえば、IEEE 8 0 2 . 1 1 規格ファミリーからのもう 1 つの規格を実装するように構成された Wi - Fi モデムであり得る。トランスポートユニット 3 3 3 は、カプセル化されたデータユニットをカプセル化解除することができる。たとえば、トランスポートユニット 3 3 3 は、カプセル化されたデータユニットから符号化されたビデオデータを抽出し、かつ復号され出力用にレンダリングされるべき符号化された A / V データをプロセッサ 3 3 1 に送ることができる。ディスプレイプロセッサ 3 3 5 は、ローカルディスプレイ 3 6 2 に表示されるべき復号されたビデオデータを処理することができ、オーディオプロセッサ 3 3 6 は、スピーカ 3 6 3 に出力するために復号されたオーディオデータを処理することができる。

【 0 0 5 6 】

[0066]オーディオおよびビデオデータをレンダリングすることに加えて、ワイヤレスシンクデバイス 3 6 0 は、ユーザ入力インターフェース 3 7 6 を介してユーザ入力データも受信することができる。ユーザ入力インターフェース 3 7 6 は、限定はしないが、タッチディスプレイインターフェース、キーボード、マウス、ボイスコマンドモジュール、(たとえば、カメラベースの入力キャプチャ機能を有する)ジェスチャキャプチャデバイスを含む、いくつかのユーザ入力デバイスのいずれも、またはいくつかのユーザ入力デバイスの任意の他のユーザ入力デバイスを表すことができる。ユーザ入力インターフェース 3 7 6 を介して受信されたユーザ入力は、プロセッサ 3 3 1 によって処理することができる。

この処理は、受信されたユーザ入力コマンドを含むデータパケットを生成することを含むことができる。生成されると、トランスポートユニット333は、そのデータパケットを、UIBCを介したソースデバイスへのネットワークトランスポート用に処理することができる。

【0057】

[0067]図3のプロセッサ331は、1つもしくは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、他の等価な集積回路もしくはディスクリート論理回路などの広範囲のプロセッサのうちの1つまたは複数、あるいはそれらの何らかの組合せを備えることができる。図3のメモリ332は、限定はしないが、同期ダイナミックランダムアクセスメモリ(SDRAM)などのランダムアクセスメモリ(RAM)、読取り専用メモリ(ROM)、不揮発性ランダムアクセスメモリ(NVRAM)、電氣的消去可能プログラマブル読取り専用メモリ(EEPROM)、フラッシュメモリなどを含む、多種多様な揮発性メモリまたは不揮発性メモリのいずれも備えることができる。メモリ232は、オーディオ/ビデオデータならびに他の種類のデータを記憶するためのコンピュータ可読記憶媒体を備えることができる。加えて、メモリ332は、本開示に記載される様々な技法を実行することの一部として、プロセッサ331によって実行される命令とプログラムコードとを記憶することができる。

【0058】

[0068]本開示の技法は、シンクデバイス360でのユーザ体験を改善するために、待ち時間を削減し、ビデオの再生品質(すなわち、滑らかさ)を改善するように、シンクデバイス360の処理パイプラインを構成することを含む。シンクデバイス360の処理パイプラインは、プロセッサ331および/またはトランスポートユニット333によって実行される1つまたは複数の処理ユニットを含む。本技法は、図6に示されるシンクデバイス660の処理パイプラインに関して、さらに詳細に記載される。

【0059】

[0069]一例では、シンクデバイス360の処理パイプラインは、WDシステム内のソースデバイスから受信されたメディアデータのタイプに基づいて、シンクデバイス360でメディアデータのカスタマイズされた再生を提供するように構成され得る。より具体的には、メディアデータがビデオデータのみを含み、オーディオデータを含まない場合、シンクデバイス360の処理パイプラインは、ビデオデータの加速されたレンダリングを実行するように構成され得る。たとえば、メディアデータがオーディオデータを含まないことを検出すると、シンクデバイス360は、同期を無効にし、存在しないオーディオデータとの同期を待たずに、ビデオデータをレンダリングすることができる。一方、メディアデータがビデオデータとオーディオデータの両方を含むことを検出すると、シンクデバイス360は、オーディオレンダリング開始タイマを削減し、削減された開始タイマに従って、同期されたオーディオデータとビデオデータとをレンダリングすることができる。

【0060】

[0070]加えて、シンクデバイス360の処理パイプラインは、ストリームヘッダ情報に基づいて、ソースデバイスからメディアデータを受信する前に構成され得る。ストリームヘッダ情報は、WDシステム内の通信セッションの能力交渉期間の間に、シンクデバイス360とソースデバイスによって交換され得る。このようにして、シンクデバイス360の処理パイプラインは、受信されたメディアデータの処理を直ちに開始し、処理パイプラインのセットアップ時間に起因するいかなる待ち時間も除去または削減することができる。

【0061】

[0071]別の例として、本技法によれば、シンクデバイス360の処理パイプラインは、ソースデバイスから受信されたメディアデータについてのアプリケーション認識に基づいて、メディアデータのカスタマイズされたバッファリングを提供するように構成され得る。より具体的には、シンクデバイス360は、受信されたメディアデータ用のアプリケー

ションのタイプを知り、処理パイプライン内のバッファのサイズを調整して、アプリケーションのタイプについての滑らかさと待ち時間との間の適切なバランスを実現することができる。場合によっては、メディアデータ用のアプリケーションのタイプはソースデバイスで検出され、シンクデバイス360にメディアデータとともに通知され得る。その場合、シンクデバイス360は、ソースデバイスから受信された指示に基づいて、メディアデータ用のアプリケーションのタイプを知る。他の場合、シンクデバイス360は、アプリケーションのタイプ自体を検出することによって、メディアデータ用のアプリケーションのタイプを知ることができる。

【0062】

[0072]メディアデータが、再生の品質または滑らかさがシンクデバイスで最も優先度が高く、上述された低遅延技法が目に見えるジッタをもたらし場合がある、ビデオ再生アプリケーション用であるとき、シンクデバイス360の処理パイプライン内のバッファサイズは増大されて、ビデオ再生アプリケーション用内のメディアデータの滑らかさを増大させる。反対に、メディアデータが、低遅延がシンクデバイスで最も優先度が高い、ユーザインターフェース（UI）アプリケーションまたはゲームアプリケーション用であるとき、シンクデバイス360の処理パイプライン内のバッファサイズは減少されて、UIアプリケーションまたはゲームアプリケーションについての待ち時間を削減する。

【0063】

[0073]さらなる例として、本技法によれば、シンクデバイス360のワイヤレスモデム334は、通信チャネルのトランスポート状態を記述するフィードバック情報を、WDシステム内のソースデバイスに送信することができる。シンクデバイス360は、前に受信されたメディアデータの誤り率および/またはパケット損失に基づいて、通信チャネルのトランスポート状態を判断することができる。ワイヤレスモデム334は、ソースデバイスにUIBCまたは他の逆チャネルを介してフィードバック情報を送信することができる。場合によっては、シンクデバイス360は、フィードバックストリーミング用にリアルタイムトランスポート制御プロトコル（RTCP）を使用することができる。それに応答して、ソースデバイスは、フィードバック情報に基づいて、シンクデバイス360に向けられたメディアデータのその処理を修正することができる。

【0064】

[0074]上述された技法のうちの1つまたは複数をシンクデバイス360に適用すると、WDシステム内のエンドツーエンドの待ち時間が削減され、ビデオの再生品質が増大されて、WDシステム内のシンクデバイス360でのユーザ体験が改善され得る。

【0065】

[0075]図4は、通信チャネル150を介して通信するために、図1の送信機/受信機126および送信機/受信機166によって使用できる、例示的な送信機システム410と受信機システム450とを示すブロック図である。送信機システム410では、いくつかのデータストリームのトラフィックデータが、データソース412から送信（TX）データプロセッサ414に供給される。各データストリームは、それぞれの送信アンテナを介して送信することができる。TXデータプロセッサ414は、データストリームごとのトラフィックデータを、そのデータストリーム用に選択された特定の符号化方式に基づいてフォーマットし、符号化し、インターリーブする。データストリームごとの符号化データは、直交周波数分割多重（OFDM）技法を使用して、パイロットデータと多重化することができる。限定はしないが、時分割多重アクセス（TDMA）、周波数分割多重アクセス（FDMA）、符号分割多重アクセス（CDMA）、またはOFDM、FDMA、TDMA、および/もしくはCDMAの任意の組合せを含む、多種多様な他のワイヤレス通信技法も使用することができる。

【0066】

[0076]図4に従って、パイロットデータは、通常、知られている方式で処理される知られているデータパターンであり、チャネル応答を推定するために受信機システム450で使用することができる。次いで、データストリームごとに多重化されたパイロットデータ

10

20

30

40

50

および符号化データは、変調シンボルを与えるために、そのデータストリーム用に選択された特定の変調方式（たとえば、2位相シフトキーイング（BPSK）、4位相シフトキーイング（QPSK）、M-PSK、またはM-QAM（直交振幅変調）、ここでMは2のべき乗であり得る）に基づいて、変調（たとえば、シンボルマッピング）される。データストリームごとのデータレート、符号化、および変調は、メモリ432と結合できるプロセッサ430によって実行される命令によって決定することができる。

【0067】

[0077]次いで、データストリーム用の変調シンボルがTX MIMOプロセッサ420に供給され、TX MIMOプロセッサ420はさらに（たとえば、OFDM用に）その変調シンボルを処理することができる。次いで、TX MIMOプロセッサ420は、 N_T 個の変調シンボルストリームを N_T 個の送信機（TMTx）422A～422T（「送信機422」）に供給することができる。いくつかの態様では、TX MIMOプロセッサ420は、データストリームのシンボルと、シンボルがそこから送信されているアンテナとにビームフォーミング重みを適用する。送信機422の各々は、それぞれのシンボルストリームを受信し処理して、1つまたは複数のアナログ信号を供給し、さらに、それらのアナログ信号を調整（たとえば、増幅、フィルタ処理、およびアップコンバート）して、MIMOチャネルを介して送信するのに適した変調信号を供給することができる。次いで、送信機422からの N_T 個の変調信号は、それぞれ N_T 個のアンテナ424A～424t（「アンテナ424」）から送信される。

【0068】

[0078]受信機システム450では、送信された変調信号は N_R 個のアンテナ452A～452R（「アンテナ452」）によって受信され、アンテナ452の各々からの受信信号は、受信機（RCVR）454A～454R（「受信機454」）のうちのそれぞれに供給される。受信機454の各々は、それぞれの受信信号を調整（たとえば、フィルタ処理、増幅、およびダウンコンバート）し、調整された信号をデジタル化してサンプルを供給し、さらにそれらのサンプルを処理して、対応する「受信」シンボルストリームを供給する。次いで、受信（RX）データプロセッサ460は、 N_R 個の受信機454から N_R 個の受信シンボルストリームを受信し、特定の変調方式に基づいて処理して、 N_T 個の「検出」シンボルストリームを供給する。次いで、RXデータプロセッサ460は、各検出シンボルストリームを復調し、デインターリーブし、復号して、データストリームのトラフィックデータを復元する。RXデータプロセッサ460による処理は、送信機システム410でのTX MIMOプロセッサ420およびTXデータプロセッサ414によって実行される処理と相補関係にある。

【0069】

[0079]メモリ472と結合できるプロセッサ470は、どのプリコーディング行列を使用すべきかを周期的に判定する。逆方向リンクメッセージは、通信リンクおよび/または受信データストリームに関する様々なタイプの情報を備えることができる。次いで、逆方向リンクメッセージは、データソース436からいくつかのデータストリームのトラフィックデータも受信するTXデータプロセッサ438によって処理され、変調器480によって変調され、送信機454によって調整され、送信機システム410に返送される。

【0070】

[0080]送信機システム410では、受信機システム450からの変調信号は、アンテナ424によって受信され、受信機422によって調整され、復調器440によって復調され、受信機システム450によって送信された逆方向リンクメッセージを抽出するためにRXデータプロセッサ442によって処理される。次いで、プロセッサ430は、ビームフォーミング重みを決定するためにどのプリコーディング行列を使用すべきかを判定し、抽出されたメッセージを処理する。

【0071】

[0081]図5は、本開示の技法をサポートして、ソースデバイス520の処理パイプライン550内の待ち時間を削減することが可能なソースデバイス520の例を示すブロック

10

20

30

40

50

図である。ソースデバイス 520 は、図 1 からのソースデバイス 120 または図 2 からのソースデバイス 220 と同様のデバイスであり得、ソースデバイス 120 またはソースデバイス 220 と同じ方式で動作することができる。

【0072】

[0082]ソースデバイス 520 は、ローカルディスプレイ 522 と、ディスプレイプロセッサ 535 と、メモリ 532 と、ワイヤレスモデムソケット 570 と、ワイヤレスモデム 534 と、処理パイプライン 550 とを含む。処理パイプライン 550 は、バッファと、プロセッサ 531 およびトランスポートユニット 533 によって実行される処理ユニットとを含む。具体的には、処理パイプライン 550 は、プロセッサ 531 内のビデオ処理エンジン (VPE) 560 およびエンコーダ 562 と、トランスポートユニット 533 内のパケットサイズ 564 とを含む。加えて、処理パイプライン 550 は、ディスプレイプロセッサ 535 と VPE 560 との間に位置するライトバック (WB) バッファ 540 と、VPE 560 とエンコーダ 562 との間のフレームバッファ 542 と、エンコーダ 562 とパケットサイズ 564 との間のコード化ピクチャバッファ (CPB) 544 とを含む。

10

【0073】

[0083]ソースデバイス 520 はまた、ハードウェア加速器 536 とパイプラインマネージャ 538 とを含む。本開示の技法によれば、パイプラインマネージャ 538 は、ソースデバイス 520 でメディアデータについての低遅延の画面取込みとバッファリングとを提供するように、処理パイプライン 550 を構成する。具体的には、パイプラインマネージャ 538 は、最小サイズのバッファを含み、ハードウェア加速器 536 を使用してメモリ 532 および最小サイズのバッファからメディアデータを取り出すように、処理パイプライン 550 を構成する。

20

【0074】

[0084]一例では、WD システム内の 1 つまたは複数のシンクデバイスと通信セッションを確立すると、パイプラインマネージャ 538 は、待ち時間を削減するために処理ステップ間で最小サイズのバッファを含むように、ソースデバイス 520 の処理パイプライン 550 を構成する。ある場合には、最小サイズのバッファは、メディアデータがディスプレイプロセッサ 535 と VPE 560 との間を通り抜けるために、削減された数の WB バッファ 540 を含む場合がある。たとえば、パイプラインマネージャ 538 は、メディアデータの少なくとも直近のフレーム更新を保持するために、2 つのみの WB バッファ 540 、たとえばピンポンバッファを含むように、処理パイプライン 550 を構成することができる。WB バッファ 540 は、VPE 560 による処理の前にフレーム更新を保持することができる。加えて、WB バッファ 540 は、メモリ 532 にライトバックされる前に、少なくとも直近のフレーム更新を保持するように構成され得る。たとえば、2 つの WB バッファ 540 は、32 個未満のエントリ、場合によっては 16 個未満のエントリを保持するように構成され得る。

30

【0075】

[0085]他の場合には、最小サイズのバッファはまた、メディアデータが VPE 560 とエンコーダ 562 との間を通り抜けるために、削減された数のフレームバッファ 542 を含む場合がある。たとえば、パイプラインマネージャ 538 は、メディアデータの少なくとも直近のフレーム更新を保持するために、4 つのみのフレームバッファ 542 、たとえば、2 つのピンポンバッファと 2 つの保持バッファとを含むように、処理パイプライン 550 を構成することができる。加えて、最小サイズのバッファは、メディアデータがエンコーダ 562 とパケットサイズ 564 との間を通り抜けるために、削減された数の CPB 544 を含む場合がある。たとえば、パイプラインマネージャ 538 は、メディアデータの少なくとも直近のフレーム更新を保持するために、6 個のみの CPB 544 、たとえば 2 つのピンポンバッファと 4 つの保持バッファとを含むように、処理パイプライン 550 を構成することができる。

40

【0076】

[0086]ソースデバイス 520 の処理パイプライン 550 内で最小サイズのバッファを使

50

用すると、本技法によれば、フレーム更新が処理の前と間により少なく小さいバッファを通り抜けるので、ソースデバイス520での待ち時間が削減される。加えて、WBバッファ540のサイズを32個未満のエントリを有する2つのピンポンバッファに最小化すると、処理パイプライン550のさらなるダウンストリームを詰まらせるのではなく、ソースデバイス520の処理パイプライン550の早い段階でフレームドロップが発生することが可能になる。

【0077】

[0087]ソースデバイス520の処理パイプライン550が最小サイズのバッファを含むように修正されたとき、パイプラインマネージャ538はまた、直近のフレーム更新がバッファリングされ、処理され、シンクデバイスに送信されることを保証するように、処理パイプライン550内のフレーム取込みプロセスを修正することができる。本技法は、最小サイズのバッファにメディアデータから取り込まれた少なくとも直近のフレーム更新をバッファリングすることと、最小サイズのバッファが一杯になったとき、より古いフレーム更新を削除することを含む。たとえば、各フレーム更新（たとえば、部分フレームまたは全体フレームのいずれか）は、ソースデバイス520内のWDシステム専用のWBバッファ560にキューイングされ得るか、または蓄積され得る。最小サイズのWBバッファ560がさらなるフレーム更新のために一杯になった場合、WBバッファ560は、より古いフレーム更新を削除し、より最近のフレーム更新を維持するように構成され得る。最小サイズのWBバッファ560は、新しいフレーム更新が受信されるやいなや処理用にフレーム更新を出力することができない可能性がある。本技法によれば、より古いフレーム更新のうちの1つまたは複数は、新しいフレーム更新がバッファリングされ、処理され、送信されることを可能にするために、WBバッファ560から削除され得る。

【0078】

[0088]別の例として、処理パイプライン550は、メモリ532、WBバッファ540、フレームバッファ542、および/またはCPB544からフレーム更新を取り出すようにハードウェア加速を使用するように構成され得る。たとえば、処理パイプライン550は、ハードウェア加速器536を使用して、ソースデバイス520の中央処理装置（CPU）の代わりに処理パイプライン550内の処理ユニットにより、処理用にフレーム更新を取り出すことができる。ハードウェア加速器536を使用すると、ソースデバイス520のCPU上の処理負荷が低減され得、このことは、フレームレートを増大させ、ソースデバイス520内の待ち時間を削減することができる。

【0079】

[0089]メディアデータの取り込まれたフレームのピクセル用イメージフォーマットは、通常、RGB888、RGB565、またはYUV生フォーマットのうちの1つである。ソースデバイス520のCPUを使用してメディアデータの取り込まれたフレームの標準メモリコピーにアクセスすると、高いフレームレートの更新を実現するために多くのCPU負荷がかかりすぎる。WDシステム用のメディアデータ処理がソースデバイス550で過剰なCPU処理負荷を消費するとき、ユーザは、WDシステム内のソースデバイス550およびシンクデバイス上で実行されているメディアアプリケーションの待ち時間または緩慢さに気付く場合がある。本技法は、ハードウェア加速器536を使用して直接メモリアクセス（DMA）を実行して、メモリ532またはバッファのうちの1つに保持されたフレーム更新を取り出し、CPUから独立して、処理パイプライン550内の処理ユニット間でフレーム更新を移動させることを含む。

【0080】

[0090]ソースデバイス520内のエンコーダ562は、ハードウェアベースおよび/またはソフトウェアベースであり得る。VPE560がフレーム更新をフレームバッファ542内にレンダリングした後、エンコーダ562はフレーム更新を符号化する。符号化されたフレーム更新は、パケットタイザ564によるパケット情報およびワイヤレスモデム534によるシンクデバイスへの送信の前に、CPB544にバッファリングされ得る。場合によっては、CPB544は、CPB544にバッファリングするために符号化された

フレーム更新のさらなるメモリコピーを作成する代わりに、メモリ 532 に記憶された符号化されたフレーム更新へのポインタをバッファリングするように構成され得る。一例として、ポインタは、メモリ 532 内の符号化されたフレーム更新のソースアドレスを指定することができ、その結果、符号化されたフレーム更新は C P B 544 内で参照されるが、実際のメモリコピーはメモリ 532 に保持される。このようにして、ソースデバイス 520 は、符号化されたフレーム更新のさらなるメモリコピーを生成および記憶することのコストを回避することができる。他の例では、本技法は、他の方法を使用して、さらなるメモリコピーのコストを節約することができる。より具体的には、本技法は、符号化されたフレーム更新のさらなるメモリコピーを作成せずに C P B 544 内の符号化されたフレーム更新を参照するために、マッピングされたメモリまたはメモリ 532 への送受信要求を使用することを含むことができる。

10

【0081】

[0091]さらなる一例では、本技法はまた、シンクデバイスによる時宜を得た受信を保証して、WDシステム内の待ち時間をさらに削減するために、ソースデバイス 520 からの符号化されたフレーム更新の再送信（すなわち、重複プッシュ）を含むことができる。重複プッシュは、ソース 520 とシンクデバイスとの間で多くのメディアデータパケットが削除され得る不十分な通信チャネルの場合、特に有用であり得る。一例では、ソースデバイス 520 は、失われたフレームを再送信するためにシンクデバイスからのフィードバックに依拠しない場合がある。代わりに、ソースデバイス 520 のワイヤレスモデム 534 は、新しいフレーム更新がシンクデバイスへの送信用に受信されなかった場合、所定の時間期間の後、フレーム更新を自動的に再送信することができる。たとえば、ワイヤレスモデム 534 は、新しいフレーム更新が受信されなかった場合、約 13 秒後に最新のフレーム更新を自動的に再送信することができる。他の場合、ワイヤレスモデム 534 は、約 10 秒またはそれ未満の後に最新のフレーム更新を再送信することができる。

20

【0082】

[0092]ソースデバイス 520 で実行される重複プッシュにより、ディスプレイコンテンツが、信頼できないトランスポートに起因してシンクデバイスで破損したフレームから回復することが可能になり得る。たとえば、ソースデバイス 520 のワイヤレスモデム 534 が、たとえば、伝送制御プロトコル（TCP）と比較して相対的に信頼できないトランスポートプロトコルであるユーザデータグラムプロトコル（UDP）を使用するとき、ワイヤレスモデム 534 は、所定の時間期間の後新しいフレーム更新が受信されなかった場合、重複プッシュを実行して最新のフレーム更新を再送信することができる。いくつかの例では、重複プッシュ間隔は、送信されているメディアデータのタイプに基づいて、動的に調整可能であり得る。ユーザインターフェース（UI）アプリケーションまたはゲームアプリケーション用のメディアデータの場合、失われたフレームは目立ち、WDシステム内のUIアプリケーションのユーザ体験に悪影響を及ぼす。ビデオ再生アプリケーション用のメディアデータの場合、失われたフレームは容易に目立たず、ビデオ再生アプリケーションのユーザ体験に少ししか影響を与えないか、まったく影響を与えない。したがって、UIアプリケーションおよびゲームアプリケーションの場合、ワイヤレスモデム 534 は、ビデオ再生アプリケーションよりも頻繁に重複プッシュを実行するように構成され得る。たとえば、UIアプリケーションおよびゲームアプリケーションの場合、重複プッシュ間隔は、約 10 秒またはそれ未満に削減され得る。

30

40

【0083】

[0093]場合によっては、ソースデバイス 520 のエンコーダ 562 および / またはシンクデバイスの各々のデコーダからフレームをプッシュするために、同様の重複プッシュメカニズムが使用され得る。通常、エンコーダおよびデコーダは、コード化されたフレームを各々自動的に出力しないが、次のフレームを受信して前のコード化されたフレームを押し出す必要がある。したがって、本技法は、さらなるプッシュを与えてコード化されたフレーム更新を出力することによって、ソースデバイス 520 と 1 つまたは複数のシンクデバイスとの間の同期を改善することができる。

50

【 0 0 8 4 】

[0094]加えて、本開示の技法はまた、W Dシステム内のソースデバイス5 2 0とシンクデバイスとの間で、オーディオデータのトランスポートをビデオデータのトランスポートより優先することを含む。ビデオデータパケットのトランスポートは、関連するオーディオデータパケットのトランスポートに結び付けられ、その結果、すべてのオーディオパケットがシンクデバイスに到達することを保証するので、削除されたパケットの受信を待つシンクデバイスでの待ち時間が削減される。パイプラインマネージャ5 3 8は、ソースデバイス5 2 0内のビデオパイプライン経路よりも多くのバッファリングを含むように、オーディオパイプライン経路を構成することができる。さらなるバッファリングにより、ソースデバイス5 2 0で削除されるオーディオパケットがより少なくなることが保証される。ワイヤレスモデムソケット5 7 0は、ビデオパイプライン経路よりもオーディオパイプライン経路に、より高い優先度のトランスポートキューを与えることができる。より高い優先度のトランスポートキューにより、キューイングされるべき対応するオーディオパケットを待つビデオパケットに起因するビデオパイプライン経路内の遅延または失速を回避するために、オーディオパケットが対応するビデオパケットより前にソースデバイス5 2 0でトランスポート用にキューイングされることが保証される。

10

【 0 0 8 5 】

[0095]従来、オーディオとビデオの両方の処理パイプライン経路は、同じまたは同様の量のバッファリングとトランスポート優先度とを有する。オーディオデータは、すべてのメディアデータがシンクデバイスで受信されることを保証するために、より注意深い処理とトランスポートとを必要とするので、ソースデバイス内の従来のビデオパイプライン経路は、関連するオーディオデータの準備ができるまでビデオデータのトランスポートを遅延または失速させ、このことはW Dシステム内にさらなる待ち時間をもたらす。オーディオパイプライン経路を優先させることによって、ビデオパイプライン経路内のさらなる遅延または失速時間が不要になる。

20

【 0 0 8 6 】

[0096]本技法によれば、オーディオデータの優先トランスポートは、パケットがオーディオトラフィックを含むか、またはビデオトラフィックを含むかを示すように、メディアデータパケット内のサービスタイプ(T O S)フィールドを設定することによって実現され得る。たとえば、通信チャネルがW i - F i マルチメディア(W M M)をサポートする場合、T O Sフィールドは、音声と、ビデオと、ベストエフォートと、バックグラウンドとを含む優先順位づけ目的で、W M Mによって規定されたアクセスカテゴリを示すことができる。メディアデータパケット内のT O Sフィールドにより、ソースデバイス5 5 0内のワイヤレスモデムソケット5 7 0が、オーディオパケットとビデオパケットとを、異なる優先レベルを有する異なるキューに入れることが可能になる。

30

【 0 0 8 7 】

[0097]加えて、ソースデバイス5 2 0は、シンクデバイスとの通信チャネルのトランスポート状態を記述するフィードバック情報をシンクデバイスから受信することができる。たとえば、フィードバック情報は、シンクデバイスで判断され、ソースデバイス5 2 0に通信される誤り率またはパケット損失を含む場合がある。場合によっては、フィードバック情報は、W Dシステム内のシンクデバイスとソースデバイス5 2 0との間で確立されたU I B Cを介して、シンクデバイスからソースデバイス5 2 0に送られ得る。他の場合、フィードバック情報は、異なるソケットリンクを介してソースデバイス5 2 0に送られ得る。加えて、フィードバック情報は、R T C Pまたは他の何らかのカスタマイズされたシグナリングを使用して通信され得る。

40

【 0 0 8 8 】

[0098]それに応答して、ソースデバイス5 2 0のパイプラインマネージャ5 3 8は、フィードバック情報に基づいてソースデバイス5 2 0の処理パイプライン5 5 0を修正することができる。一例では、不十分な通信チャネルを記述するフィードバック情報を受信したことに応答して、ソースデバイス5 2 0は、メディアデータパケットを再送信する重複

50

プッシュ間隔を、たとえば10秒以上または13秒以上に増大させるか、またはメディアデータ packets を処理および送信するためのデータレートを変更することができる。別の例では、フィードバック情報にตอบสนองして、ソースデバイス520は、全体的に、バッファサイズと符号化パラメータとを調整するか、またはトランスポートプロトコルのタイプ間を動的に切り替えることができる。たとえば、不十分な通信チャネルを記述するフィードバック情報にตอบสนองして、ソースデバイス520は、比較的信頼できないトランスポートプロトコルであるリアルタイムプロトコル(RTP)/ユーザデータグラムプロトコル(UDP)の通信チャネルから、RTP/伝送制御プロトコル(TCP)のトランスポートチャネルに切り替えることができる。

【0089】

10

[0099]図6は、本開示の技法をサポートして、シンクデバイス660の処理パイプライン650内の待ち時間を削減し、シンクデバイス660でのビデオ再生を改善することが可能なシンクデバイス660の例を示すブロック図である。シンクデバイス660は、図1からのシンクデバイス160または図3からのシンクデバイス260と同様のデバイスであり得、シンクデバイス160またはシンクデバイス260と同じ方式で動作することができる。

【0090】

[0100]シンクデバイス660は、ローカルディスプレイ622と、ディスプレイプロセッサ635と、メモリ632と、ワイヤレスモデムソケット670と、ワイヤレスモデム634と、処理パイプライン650とを含む。処理パイプライン650は、バッファと、プロセッサ631およびトランスポートユニット633によって実行される処理ユニットとを含む。具体的には、処理パイプライン650は、トランスポートユニット633内のパーサ680と、プロセッサ631内のデコーダ682およびレンダラ684とを含む。加えて、処理パイプライン650は、パーサ680とデコーダ682との間に位置するバッファ692と、デコーダ682とレンダラ684との間に位置するレンダリングキュー684と、レンダラ684とディスプレイプロセッサ635との間のフレームバッファ696とを含む。

20

【0091】

[0101]シンクデバイス660は、パイプラインマネージャ638も含む。本開示の技法によれば、パイプラインマネージャ638は、ソースデバイスから受信されたメディアデータのタイプに基づいて、シンクデバイス660でのカスタマイズされた再生を提供するように、処理パイプライン650を構成する。具体的には、パイプラインマネージャ638は、メディアデータがオーディオデータを含むかどうかに基づいてレンダリングを修正するように、処理パイプライン650を構成する。

30

【0092】

[0102]たとえば、メディアデータがビデオデータのみを含み、オーディオデータを含まない場合、シンクデバイス660の処理パイプライン650に含まれるレンダラ684は、ビデオデータの加速されたレンダリングを実行するように構成される。メディアデータがオーディオデータを含まないことを検出すると、パイプラインマネージャ638は、レンダラ684での同期を無効にすることができる。次いで、レンダラ684は、存在しないオーディオデータとの同期を待たずにビデオデータをレンダリングすることができる。

40

【0093】

[0103]従来、メディアデータ処理パイプラインは、ビデオデータとオーディオデータの両方、またはオーディオデータがないビデオデータのためのために設計される。ビデオデータのみを受信すると、ビデオデータとオーディオデータの両方のために設計されたメディアデータ処理パイプラインは、ビデオデータとオーディオデータの両方を処理するのに必要な速度と同じ速度でビデオデータを処理し、存在しないオーディオデータとの同期用の待ち時間を含む。さらなる処理時間は、処理パイプラインに待ち時間を不必要に加える。本開示の技法により、シンクデバイス660のデコーダ682が、受信されたメディアデータのタイプを検出し、メディアデータのタイプに基づいてレンダラ684での再生処理

50

速度を動的に調整することが可能になる。たとえば、デコーダ 682 は、メディアデータ内のオーディオタイムスタンプの存在または不在を検出することができ、オーディオタイムスタンプの存在により、ソースデバイスからのメディアデータがオーディオデータを含むことが示される。

【0094】

[0104]たとえば、UIアプリケーション用のメディアデータの場合、ビデオデータに付随するオーディオデータは存在しない。UIアプリケーションデータの場合、パイプラインマネージャ 638 は、シンクデバイス 660 のレンダラ 684 での同期を無効にすることができる。このようにして、レンダラ 684 は、存在しないオーディオデータの処理および同期のための待ち時間なしに、可及的速やかにUIアプリケーション用のビデオデータをレンダリングすることができる。ビデオ再生アプリケーション用のメディアデータの場合、ビデオデータに付随するオーディオデータが存在する場合がある。オーディオデータ付きのビデオ再生アプリケーションデータの場合、パイプラインマネージャ 638 は同期を有効にすることができ、レンダラ 684 は、同期されたビデオデータとオーディオデータとを通常の処理速度でレンダリングすることができる。

10

【0095】

[0105]加えて、本技法により、パイプラインマネージャ 638 がレンダラ 684 に適用可能なメディア時間に基づいて、レンダラ 684 でのサンプル時間を調整することも可能になる。このようにして、ソースデバイスによって設定されたメディアデータのサンプル時間がシンクデバイス 660 にあるレンダラ 684 に適用可能ではないとき、パイプラインマネージャ 638 は、メディア時間に基づいてメディアデータに適用可能なサンプル時間を選択することができる。

20

【0096】

[0106]別の例として、メディアデータがビデオデータとオーディオデータの両方を含む場合、パイプラインマネージャ 638 は、オーディオレンダリング開始タイマを削減することができる、レンダラ 684 は削減された開始タイマに従って、同期されたオーディオデータとビデオデータとをレンダリングすることができる。このようにして、本技法は、メディアデータがオーディオデータを含むときでも、シンクデバイス 660 での待ち時間を削減することができる。

【0097】

30

[0107]いくつかの再生エンジンでは、着信したオーディオパケットの消失または削除を回避するために、オーディオのレンダリングは起動時に遅延され得る。遅延時間は、Bluetoothシステムなどの何らかの通信システムに必要であり得る。WDシステムなどの他の通信システムの場合、同期が有効であるとき、遅延は不要であり、エンドツーエンドの待ち時間を増加させる。たとえば、Bluetoothシステムに必要なアドバンストオーディオ配信プロファイル(A2DP)初期化時間、たとえば、1秒以上を補償するために、シンクデバイス 660 のレンダラ 684 でのオーディオレンダリング開始タイマは、比較的高く設定され得る。オーディオデータのレンダリングは開始時間だけ遅延され、着信したサンプルはレンダラ 684 に入力される前にバッファリングされる必要がある。オーディオデータが遅延された場合、オーディオデータとビデオデータの同期を保つために、対応するビデオデータも遅延される必要がある。

40

【0098】

[0108]本開示に記載された技法によれば、WDシステムの場合、レンダラ 684 でのオーディオレンダリング開始タイマは削減されて、シンクデバイス 660 内の待ち時間を削減し、オーディオデータとビデオデータの失速を回避する。たとえば、WDシステムの場合、パイプラインマネージャ 638 は、シンクデバイス 660 でのより速いレンダリングと再生とを提供するために、オーディオレンダリング開始タイマを削減または除去することができる。オーディオレンダリング開始タイマは、1秒未満、50ミリ秒(ms)未満、場合によっては20ms未満になるように削減され得る。Bluetoothシステムの場合、パイプラインマネージャ 638 は、比較的高い、たとえば1秒以上になるように

50

オーディオレンダリング開始タイマを動的にリセットすることができる。

【0099】

[0109]別の例では、パイプラインマネージャ638は、セットアップ時間に起因する待ち時間を削減するために通信セッションの能力交渉期間の間に交換されたストリームヘッダ情報に基づいて、ソースデバイスからメディアデータを受信する前に、シンクデバイス660内の処理パイプライン650を構成することができる。処理パイプライン650を事前構成するために、ソースデバイスおよびシンクデバイス660は、通信チャネルを介して送信されるべきメディアデータについての情報を含むストリームヘッダ情報を最初に交換する。ストリームヘッダ情報は、リアルタイムストリーミングプロトコル(RTSP)または他のプロプライエタリなメッセージを使用して交換され得る。次いで、パイプラインマネージャ638は、構成バッファサイズと、レンダリング開始タイマと、同期待ちタイマと、プログラマブルデコード設定とを含む、受信されたヘッダ情報に基づいて処理パイプライン650をセットアップする。処理パイプライン650が構成された後、シンクデバイス660は、メディアデータの送信を開始するようにソースデバイスに通知する。

10

【0100】

[0110]加えて、ソースデバイスから受信されたメディアデータを処理する前に、シンクデバイス660は、デコード682から受信されたメディアデータの1つまたは複数のサンプルをフラッシュすることができる。メディアデータの最初の数個のサンプルは、通信セッションの交渉期間の間に生成された、ソースデバイスパイプラインからの古くて失速したサンプルを備える可能性がある。本技法は、ソースデバイスからのフレームを再送信して、シンクデバイス660にあるデコード682から失速したサンプルをプッシュするために、重複プッシュを提供する。このようにして、シンクデバイス660は、メディアデータの古くて失速したサンプルに対していかなる処理時間も浪費しない。

20

【0101】

[0111]場合によっては、シンクデバイス660にあるパイプラインマネージャ638は、レンダラ684によるレンダリング用にデコード682からサンプルをプッシュするためにデコード682にダミーフレームを挿入することができる。これは、デコード682がプログラム可能ではなく、着信したメディアデータサンプルのピクチャ順序カウンタ(POC)の復号をサポートしないとき、必要であり得る。他の場合、デコード682がプログラム可能であり、POCをサポートする場合、パイプラインマネージャ638は、表示順序の代わりに、PICまたは復号順序に従ってメディアデータサンプルを出力するように、デコード682を構成することができる。復号順序でサンプルを出力するようにプログラマブルデコード682を構成すると、デコード682へのあらゆる入力サンプルは、復号が完了するとすぐに出力されることが保証される。このようにして、デコード682での失速および遅延は、削減または除去され得る。加えて、パイプラインマネージャ638は、デコード682とレンダラ684との間のレンダリングキュー694に含まれるバッファの数を最小化して、シンクデバイス660内の待ち時間をさらに削減することができる。たとえば、レンダリングキュー694は、レンダラ684によるレンダリングの前にメディアデータを保持するバッファを、4つまたはそれより少なく含む場合がある。

30

40

【0102】

[0112]加えて、本開示の技法によれば、パイプラインマネージャ638は、ソースデバイスから受信されたメディアデータについてのアプリケーション認識に基づいて、シンクデバイス660でカスタマイズされたバッファリングを提供するように、処理パイプライン640を構成する。具体的には、ソースデバイス660のデコード682は、受信されたメディアデータ用のアプリケーションのタイプを知り、パイプラインマネージャ638は、処理パイプライン650内のバッファのサイズを調整するように処理パイプライン650を構成する。このようにして、シンクデバイス660は、メディアデータ用のアプリケーションのタイプについての滑らかさと待ち時間との間の適切なバランスを実現することができる。パイプラインマネージャ638は、レンダリングキュー694、フレームバ

50

ッファ 696、または、メディアデータ用のアプリケーション内の表示の前にデータを保持する、処理パイプライン 650 に含まれる任意の他のバッファのサイズを調整することができる。

【0103】

[0113]たとえば、メディアデータがビデオ再生アプリケーション用であるとき、再生の品質または滑らかさがシンクデバイス 660 で最も優先度が高く、上述された低遅延技法は目に見えるジッタをもたらす場合がある。この場合、パイプラインマネージャ 638 は、ビデオ再生アプリケーション内のメディアデータの滑らかさを増大させるために処理パイプライン 650 内のバッファのサイズを増大させることができる。反対に、メディアデータがユーザインターフェース (UI) アプリケーションまたはゲームアプリケーション用であるとき、低遅延がシンクデバイス 660 で最も優先度が高い。この場合、パイプラインマネージャ 638 は、UI アプリケーションまたはゲームアプリケーションについての待ち時間を削減するために処理パイプライン 650 内のバッファのサイズを減少させることができる。バッファサイズを削減することによって、パイプラインマネージャ 638 は、UI アプリケーションまたはゲームアプリケーション用のメディアデータが、失速または遅延なしにシンクデバイス 660 内の処理パイプライン 650 を通り抜けることを可能にする。

【0104】

[0114]場合によっては、メディアデータ用のアプリケーションのタイプは、ソースデバイスで検出され、WD システム内のシンクデバイス 660 に通知され得る。たとえば、シンクデバイス 660 は、ソースデバイスから受信された指示に基づいて、受信されたメディアデータ用のアプリケーションを検出することができる。ソースデバイスは、着信したオーディオレートとオーディオデータの画面更新レートとを検出することによって、メディアデータがビデオ再生アプリケーション用であると判断することができる。ソースデバイスは、ビデオ再生エンジン用のメディアデータの開始の指示をシンクデバイス 660 に送り、ビデオ再生アプリケーション用のメディアデータの終了の別の指示をシンクデバイス 660 に送ることができる。シンクデバイス 660 は、メディアプレーヤプロトコルまたは別のプロプライエタリなプロトコルで、メディアデータとともにフラグまたは他のインジケータを受信することができる。

【0105】

[0115]他の場合、メディアデータ用のアプリケーションのタイプは、シンクデバイス 660 で検出され得る。シンクデバイス 660 にあるデコーダ 682 は、一定の間隔で受信されたメディアデータに含まれるオーディオタイムスタンプに基づいて、受信されたメディアデータ用のアプリケーションのタイプを判断することができる。メディアデータが一定の間隔で対応するオーディオデータを有するビデオデータを含むとき、メディアデータはビデオ再生アプリケーション用であるとみなされ得る。したがって、デコーダ 682 は、ビデオ再生アプリケーション用のメディアデータを受信しながら、一定の間隔で発生するオーディオタイムスタンプを検出することができる。

【0106】

[0116]いずれの場合も、受信されたメディアデータがビデオ再生アプリケーション用であることをシンクデバイス 660 のデコーダ 682 が知ったとき、パイプラインマネージャ 638 は、ビデオ再生アプリケーション内の表示の前にメディアデータを保持するために使用されるバッファ、たとえば、レンダリングキュー 694 またはフレームバッファ 696 のサイズを増大させる。たとえば、パイプラインマネージャ 638 は、レンダリングキュー 694 内に 4 つ以上のバッファを含むように、かつ 4 つ以上のフレームバッファ 696、たとえば、2 つのピンポンバッファと少なくとも 2 つの保持バッファとを含むように、処理パイプライン 650 を構成することができる。バッファが大量のメディアデータを保持することが可能であるとき、フレームを落とすリスクは減少し、再生内のジッタの数も減少する。

【0107】

[0117]一方、受信されたメディアデータがビデオ再生アプリケーション用ではなく、UIアプリケーションまたはゲームアプリケーション用であることをシンクデバイス660のデコーダ682が知ったとき、パイプラインマネージャ638は、UIアプリケーションまたはゲームアプリケーション内の表示の前にメディアデータを保持するために使用されるバッファのサイズを減少させる。たとえば、パイプラインマネージャ638は、レンダリングキュー694内に4つ未満のバッファを含むように、かつ4つ以下のフレームバッファ696、たとえば、2つのピンポンバッファと多くとも2つの保持バッファとを含むように、処理パイプライン650を構成することができる。このようにして、メディアデータは、シンクデバイス660の処理パイプライン650を通り抜けて、待ち時間を改善する。

10

【0108】

[0118]場合によっては、パイプラインマネージャ638は、レンダリングキュー694にさらなるメディアデータサンプルを記憶するために、ある時間期間の間レンダラ684を休止することができる。たとえば、パイプラインマネージャ638は、しきい値の数のサンプルがレンダリングキュー694に記憶されるまでレンダラ684を休止し、次いで、レンダリングを再開してビデオ再生の滑らかさを増大することができる。場合によっては、しきい値の数のサンプルは、8個のサンプル、16個のサンプル、または場合によっては16個より多いサンプルに設定され得る。

【0109】

[0119]加えて、シンクデバイス660は、ソースデバイスにソースデバイスとの通信チャネルのトランスポート状態を記述するフィードバック情報を供給することができる。ソースデバイスは、図5に関してより詳細に上述されたように、フィードバック情報に基づいてその処理パイプラインを修正することができる。たとえば、フィードバック情報は、シンクデバイス660で判断され、ソースデバイスに通信された誤り率またはパケット損失を含む場合がある。場合によっては、ワイヤレスモデム634は、WDシステム内のシンクデバイス660とソースデバイスとの間で確立されたUIBCを介して、シンクデバイス660からソースデバイスにフィードバック情報を送ることができる。他の場合、ワイヤレスモデム634は、異なるソケットリンクを介して、シンクデバイス660からソースデバイスにフィードバック情報を送ることができる。加えて、ワイヤレスモデム634は、RTPまたは他の何らかのカスタマイズされたシグナリングを使用して、フィードバック情報を通信することができる。

20

30

【0110】

[0120]図7は、シンクデバイスで取得されたユーザ入力データおよび/またはフィードバックデータをソースデバイスに配信するために使用され得る例示的なデータパケット700を示す概念図である。図1を参照してデータパケット700の態様が説明されるが、論じられる技法はWDシステムのさらなるタイプに適用可能であり得る。データパケット700は、その後にペイロードデータ750が続くデータパケットヘッダ710を含む場合がある。データパケット700は、たとえば、シンクデバイス160で受信されたユーザ入力データをシグナリングするため、または、通信チャネル150のトランスポート状態を記述するフィードバック情報をシグナリングするために、シンクデバイス160からソースデバイス120に送信され得る。

40

【0111】

[0121]ペイロードデータ750に含まれるデータのタイプ、たとえば、ユーザ入力データまたはフィードバックデータは、データパケットヘッダ710内で識別され得る。このようにして、データパケットヘッダ710のコンテンツに基づいて、ソースデバイス120は、データパケット700のペイロードデータ750を構文解析して、シンクデバイス160からのユーザ入力データまたはフィードバックデータを識別することができる。本開示で使用する「構文解析」および「構文解析する」という用語は、一般に、ビットストリームを解析してビットストリームからデータを抽出するプロセスを指す。データを抽出することは、たとえば、ビットストリーム内の情報がどのようにフォーマットされている

50

かを識別することを含む場合がある。以下でより詳細に記載されるように、データパケットヘッダ 710 は、ペイロードデータ 750 用の多くの可能なフォーマットのうちの 1 つを規定することができる。データパケットヘッダ 710 を構文解析することによって、ソースデバイス 120 は、ペイロードデータ 750 がどのようにフォーマットされているかと、ユーザ入力コマンドまたはフィードバック情報を抽出するためにペイロードデータ 750 をどのように構文解析すべきかとを判断することができる。

【0112】

[0122]いくつかの例では、データパケットヘッダ 710 は、図 7 に示されたようにフォーマットされた 1 つまたは複数のフィールド 720 を含む場合がある。フィールド 720 に隣接する番号 0 ~ 15 ならびにビットオフセット 0、16、および 32 は、データパケットヘッダ 710 内のビット位置を識別するものであり、データパケットヘッダ 710 内に含まれている情報を実際に表すものではない。データパケットヘッダ 710 は、バージョンフィールドと、タイムスタンプフラグと、予備フィールドと、入力カテゴリフィールドと、長さフィールドと、オプションのタイムスタンプフィールドとを含む。図 7 の例では、バージョンフィールドは、シンクデバイス 160 によって実装されている特定の通信プロトコルのバージョンを示すことができる 3 ビットのフィールドである。バージョンフィールド内の値は、データパケットヘッダ 710 の残りをどのように構文解析すべきか、ならびにペイロードデータ 750 をどのように解析すべきかを、ソースデバイス 120 に通知することができる。

【0113】

[0123]図 7 の例では、タイムスタンプフラグ (T) は、タイムスタンプフィールドが存在するデータパケットヘッダ 710 内に存在するか否かを示す 1 ビットのフィールドである。存在する場合、タイムスタンプフィールドは、ソースデバイス 120 によって生成され、シンクデバイス 160 に送信されたマルチメディアデータに基づくタイムスタンプを含む 16 ビットのフィールドである。タイムスタンプは、たとえば、ビデオのフレームがシンクデバイス 160 に送信されるより前に、ソースデバイス 120 によってそのフレームに割り当てられる連続値であり得る。データパケットヘッダ 710 を構文解析し、タイムスタンプフィールドが存在するかどうかを判定すると、ソースデバイス 120 は、タイムスタンプフィールドに含まれるタイムスタンプを処理する必要があるかどうかを知る。図 7 の例では、予備フィールドは、バージョンフィールド内で識別される特定のプロトコルの将来のバージョン用に確保された 8 ビットのフィールドである。

【0114】

[0124]図 7 の例では、入力カテゴリフィールドは、ペイロードデータ 750 に含まれているデータについての入力カテゴリを識別する 4 ビットのフィールドである。たとえば、シンクデバイス 160 は、ユーザ入力データを分類して入力カテゴリを特定することができる。ユーザ入力データは、コマンドの受信元のデバイスに基づいて、またはコマンド自体の特性に基づいて、分類され得る。シンクデバイス 160 はまた、フィードバック情報を分類して入力カテゴリを特定することができる。フィードバック情報は、シンクデバイス 160 で決定されたフィードバック情報のタイプ、またはフィードバック情報に基づいてソースデバイス 120 で要求された動作のタイプに基づいて、分類され得る。入力カテゴリフィールドの値は、場合によってはデータパケットヘッダ 710 の他の情報とともに、ペイロードデータ 750 がどのようにフォーマットされているかをソースデバイス 120 に対して識別する。このフォーマットに基づいて、ソースデバイス 120 は、ペイロードデータ 750 を構文解析して、ユーザ入力コマンドまたはフィードバック情報を抽出することができる。

【0115】

[0125]長さフィールドは、データパケット 700 の長さを示す 16 ビットのフィールドを備えることができる。データパケット 700 が 16 ビットのワードでソースデバイス 120 によって構文解析されるとき、データパケット 700 は 16 ビットの整数までパディングされ得る。長さフィールドに含まれている長さに基づいて、ソースデバイス 120 は

、ペイロードデータ750の終了（すなわち、データパケット700の終了）と新しい後続のデータパケットの開始とを識別することができる。

【0116】

[0126]図7の例で提供されたフィールドの様々なサイズは単に説明用のものであり、それらのフィールドは、図7に示されたものとは異なる数のビットを使用して実装され得るものである。加えて、データパケットヘッダ710は、上記で説明されたすべてのフィールドよりも少ないフィールドを含む場合があるか、または上記で説明されていないさらなるフィールドを使用する場合があることも考えられる。実際、本開示の技法は、パケットの様々なデータフィールドに使用される実際のフォーマットの点でフレキシブルであり得る。

10

【0117】

[0127]図8は、処理パイプライン内のメディアデータの低遅延のフレーム取込みとバッファリングとをサポートすることが可能なソースデバイスの例示的な動作を示すフローチャートである。図示された動作は、図5からのソースデバイス520に含まれる処理パイプライン550に関して記載される。他の例では、図示された動作は、図2からのソースデバイス220、図1からのソースデバイス120、またはWDシステムの別のソースデバイスによって実行され得る。

【0118】

[0128]ソースデバイス520は、最初にWDシステム内の1つまたは複数のシンクデバイスと通信セッションを確立する（800）。ソースデバイス520は、シンクデバイスとの能力交渉に従って通信セッションを確立することができる。本開示の技法によれば、メディアデータの低遅延のフレーム取込みとバッファリングとをサポートするために、ソースデバイス520のパイプラインマネージャ538は、最小サイズのバッファを含むように処理パイプライン550を構成する（802）。具体的には、パイプラインマネージャ538は、WBバッファ540のサイズを低減して2つのピンポンバッファのみを含むように、処理パイプライン550を構成することができる。処理パイプライン550に沿って通り抜けるようにメディアデータ用のバッファの数を低減すると、ソースデバイス520での待ち時間が削減される。加えて、パイプラインマネージャ538は、メディアデータの少なくとも直近のフレーム更新を保持することが可能な最小サイズのフレームバッファにフレームバッファ542のサイズを低減するように、処理パイプライン550を構成することができる。

20

30

【0119】

[0129]さらに、本技法によれば、パイプラインマネージャ538は、ハードウェア加速器536を使用して処理パイプライン550内のメディアデータの直接メモリアクセスを実行するように、処理パイプライン550を構成することができる（804）。標準メモリコピーを使用してメモリ532および処理パイプライン550内のバッファからメディアデータのフレーム更新を取り出すと、CPU負荷が過剰に使用される。メディアデータのフレームレートを増大させ、待ち時間を削減するために、処理パイプライン550は、代わりにハードウェア加速器536を使用して、フレーム更新にアクセスし、CPUから独立した処理パイプライン550の処理ユニット間を移動させる。

40

【0120】

[0130]処理パイプライン550がソースデバイス520内で構成された後、ソースデバイス520は、WDシステム内のシンクデバイスへの送信用のメディアデータの処理を開始することができる。ディスプレイプロセッサ535は、メディアソースからメディアデータのフレーム更新を取り込む（806）。メディアソースは、ソースデバイス520内に含まれるカメラまたは他のメディア取込みデバイスであり得るか、または、有線接続もしくはワイヤレス接続のいずれかを介してソースデバイス520に接続された外部メディアソースであり得る。フレーム更新は、メディアデータの全フレーム、フレームの更新された部分のみを表す部分フレーム、または2つの何らかの組合せを含む場合がある。

【0121】

50

[0131] フレーム更新を取り込んだ後、ディスプレイプロセッサ 535 は、最小サイズの WB バッファ 540 が一杯かどうかを判定するよう求める要求を WB バッファ 540 に送る (808)。最小サイズの WB バッファ 540 が一杯である場合 (808 の YES 分岐)、ディスプレイプロセッサ 535 から要求を受信すると、WB バッファ 540 は、WB バッファ 540 に保持されたより古いフレーム更新のうちの 1 つまたは複数を削除するようにトリガされる (810)。より古いフレーム更新を削除した後、WB バッファ 540 は、ディスプレイプロセッサ 535 によって取り込まれた直近のフレーム更新をバッファリングする (812)。ディスプレイプロセッサ 525 から要求を受信したとき最小サイズの WB バッファ 540 が一杯でない場合 (808 の NO 分岐)、WB バッファ 540 は、ディスプレイプロセッサ 535 によって取り込まれた直近のフレーム更新を直ちにバッファリングすることができる (812)。フレームバッファ 542 などの処理パイプライン 550 に沿った他の最小サイズのバッファ内のフレーム更新をバッファリングするために、同様のプロセスが使用され得る。直近のフレーム更新が WB バッファ 540 に保持されると、次いで、フレーム更新は、ハードウェア加速器 536 を介して WB バッファ 540 からメモリ 532 に書き込まれ得る。

【0122】

[0132] 加えて、ハードウェア加速器 536 は、WB バッファ 540 からフレーム更新を取り出し、処理パイプライン 550 に沿った処理ユニットとバッファとの間を移動させる (814)。次いで、取り出されたフレーム更新は、WD システム内のシンクデバイスへの送信用に処理パイプライン 550 内で処理される (816)。たとえば、プロセッサ 531 内で、ビデオ処理エンジン (VPE) 560 は、フレームバッファ 542 内にフレーム更新をレンダリングし、エンコーダ 562 は、フレームバッファ 542 からのフレーム更新を符号化する。符号化されたフレーム更新は、トランスポートユニット 533 内のパケット化 564 でフレーム更新をパケット内に形成する前に、コード化ピクチャバッファ (CPB) 544 にバッファリングされ得る。本技法によれば、符号化されたフレーム更新のメモリコピーを CPB 544 にバッファリングする代わりに、符号化されたフレーム更新のメモリコピーを作成せずに、メモリ 532 に記憶された符号化されたフレーム更新へのポインタが CPB 544 にバッファリングされ得る。

【0123】

[0133] 次いで、ワイヤレスモデム 534 は、WD システム内の通信セッションを介して、1 つまたは複数のシンクデバイスに処理されたフレーム更新を送信する (818)。処理されたフレーム更新を送信した後、ワイヤレスモデム 534 は、新しいフレーム更新が送信用に受信されるまで、所定の時間期間後に、処理されたフレーム更新を再送信する (すなわち、重複プッシュを実行する) ことができる。重複プッシュは、シンクデバイスが送信されたフレーム更新を受信し、削除されたパケットを待つ処理時間を浪費しないことを保証することによって、WD システム内のエンドツーエンドの待ち時間を削減することができる。

【0124】

[0134] 図 9 は、処理パイプライン内のカスタマイズされたビデオ再生をサポートすることが可能なシンクデバイスの例示的な動作を示すフローチャートである。図 6 に関して記載される。図示された動作は、図 6 からのシンクデバイス 660 に含まれる処理パイプライン 650 に関して記載される。他の例では、図示された動作は、図 3 からのシンクデバイス 360、図 1 からのシンクデバイス 160、または WD システムの別のシンクデバイスによって実行され得る。

【0125】

[0135] シンクデバイス 660 は、最初に WD システム内のソースデバイスと通信セッションを確立する (900)。シンクデバイス 660 は、ソースデバイスとの能力交渉に従って通信セッションを確立することができる。本開示の技法によれば、処理パイプライン 650 内でカスタマイズされたビデオ再生をサポートするために、シンクデバイス 660 は、通信セッションについての能力交渉期間の間にメディアデータのストリーム用のスト

リームヘッダ情報をソースデバイスと交換する(902)。次いで、パイプラインマネージャ638は、受信されたストリームヘッダ情報に基づいて、シンクデバイス660内の処理パイプライン650を構成する(904)。

【0126】

[0136]たとえば、ストリームヘッダ情報に基づいて、パイプラインマネージャ638は、ソースデバイスからメディアデータを受信するより前に、処理パイプライン650内のバッファサイズと、レンダリング開始タイマと、同期待ちタイマと、プログラマブルデコーダ設定などを構成することができる。処理パイプライン650がメディアデータのストリーム用にシンクデバイス660内で構成されると、ワイヤレスモデム634は、シンクデバイス660へのメディアデータのストリームの送信を開始しようとするソースデバイスに通知することができる(906)。このようにして、復号およびレンダリングされるべきメディアデータをソースデバイスから受信すると、シンクデバイス660は、処理パイプライン線650をセットアップするいかなる処理時間も浪費しない。

【0127】

[0137]処理パイプライン650が構成された後、シンクデバイス660は、ソースデバイスからのメディアデータの受信を開始する(908)。パーサ680は、符号化されたメディアデータをパケット化解除する。次いで、デコーダ682は、受信されたメディアデータを復号する(910)。受信されたメディアデータは、少なくともビデオデータを含む。復号すると、デコーダ682または処理パイプライン650内の他の何らかの処理ユニットは、復号されたメディアデータがオーディオデータを含むかどうかを検出する(912)。たとえば、デコーダ682は、メディアデータがオーディオデータを含むことを示す、受信されたメディアデータ内のオーディオタイムスタンプを検出することによって、メディアデータがオーディオデータを含むと判断することができる。同様に、デコーダ682は、受信されたメディアデータ内のオーディオタイムスタンプの不在に基づいて、メディアデータがビデオデータのみを含むと判断することができる。

【0128】

[0138]オーディオデータがメディアデータに含まれていない場合(914のNO分岐)、処理パイプライン650はビデオデータの加速されたレンダリングを実行する。より具体的には、パイプラインマネージャ638は、レンダラ684でビデオデータのオーディオデータとの同期を無効にする(916)。次いで、レンダラ684は、存在しないオーディオデータとの同期を待たずにビデオデータをレンダリングする(918)。場合によっては、メディアデータがビデオデータのみを含むとき、パイプラインマネージャ638は、明示的に同期を無効にすることはできないが、いかなるオーディオデータも無視し、存在しないオーディオデータとの同期を待たずにビデオデータをレンダリングするように、レンダラ684を構成することができる。加えて、パイプラインマネージャ638はまた、レンダラ684に適用可能なメディア時間に基づいて、ビデオデータのサンプル時間を増大させることができる。このようにして、レンダラ684は、増大されたサンプル時間に従って、ビデオデータの加速されたレンダリングを実行することができる。

【0129】

[0139]オーディオデータがメディアデータに含まれている場合(914のYES分岐)、パイプラインマネージャ638は、レンダラ684でのオーディオレンダリング開始タイマを削減することができる(920)。たとえば、いくつかの通信システムの場合、レンダラ684でのオーディオレンダリング開始タイマは、通信システムの初期化を補償するために高く保たれ得る。これはWDシステムに必要ではない。したがって、パイプラインマネージャ638は、オーディオデータがメディアデータに含まれているときでも、待ち時間を削減するために、WDシステム用のオーディオレンダリング開始タイマを削減することができる。次いで、レンダラ638は、ビデオデータをオーディオデータと同期し(922)、削減された開始タイマに従って、同期されたビデオデータとオーディオデータとをレンダリングする(924)ことができる。

【0130】

[0140] 場合によっては、ビデオデータのフレームはデコーダ 6 8 2 内で失速し、レンダラ 6 8 4 がデコーダ 6 8 2 から次のビデオフレームを受信することを待つ、シンクデバイス 6 6 0 でのさらなる待ち時間をもたらす場合がある。本技法によれば、デコーダ 6 8 2 が非プログラマブルデコーダであるとき、パイプラインマネージャ 6 3 8 は、レンダリング用にデコーダ 6 8 2 からビデオデータの復号されたサンプルをプッシュするためにシンクデバイス 6 6 0 でビデオデータにダミーフレームを挿入することができる。デコーダ 6 8 2 がプログラマブルデコーダであるとき、パイプラインマネージャ 6 3 8 は、復号が完了するとすぐにレンダリング用に復号順序でビデオデータの復号されたサンプルを出力するように、プログラマブルデコーダ 6 8 2 を構成することができる。通常、デコーダは、デフォルトではサンプルを表示順序で出力する。この場合、パイプラインマネージャ 6 3 8 は、デコーダ 6 8 2 を介して復号されたフレームをより早く移動させるように、デフォルト設定を変更することができる。

10

【 0 1 3 1 】

[0141] 加えて、ソースデバイスからメディアデータを受信すると、パイプラインマネージャ 6 3 8 は、受信されたメディアデータをレンダリングするより前に、受信されたメディアデータの 1 つまたは複数のサンプルをデコーダ 6 8 2 からフラッシュすることができる。場合によっては、ソースデバイスから送信された最初の数個のサンプルは、ソースデバイスでより古いメディアデータの失速したサンプルを含む場合がある。パイプラインマネージャ 6 3 8 は、これらのサンプルをシンクデバイス 6 6 0 からフラッシュして、古いメディアデータに対する処理時間およびリソースの消費を回避することができる。

20

【 0 1 3 2 】

[0142] 図 1 0 は、処理パイプライン内のメディアデータのアプリケーション認識に基づいて、カスタマイズされたバッファリングをサポートすることが可能なシンクデバイスの例示的な動作を示すフローチャートである。図示された動作は、図 6 からのソースデバイス 6 6 0 に含まれる処理パイプライン 6 5 0 に関して記載される。他の例では、図示された動作は、図 3 からのシンクデバイス 3 6 0、図 1 からのシンクデバイス 1 6 0、または W D システムの別のシンクデバイスによって実行され得る。

【 0 1 3 3 】

[0143] シンクデバイス 6 6 0 は、最初に W D システム内のソースデバイスと通信セッションを確立する (1 0 0 0)。シンクデバイス 6 6 0 は、ソースデバイスとの能力交渉に従って通信セッションを確立することができる。次いで、シンクデバイス 6 6 0 は、ソースデバイスからのメディアデータの受信を開始する (1 0 1 0)。本開示の技法によれば、処理パイプライン 6 5 0 内のメディアデータ用のアプリケーションのタイプに基づいてカスタマイズされたバッファリングをサポートするために、シンクデバイス 6 6 0 は、受信されたメディアデータ用のアプリケーションのタイプを知り、パイプラインマネージャ 6 3 8 は、アプリケーションのタイプに基づいて処理パイプライン 6 5 0 内のバッファのサイズを調整する。

30

【 0 1 3 4 】

[0144] 受信されたメディアデータ用のアプリケーションのタイプを知るために、パーサ 6 8 0 は、最初にメディアデータ用のアプリケーションのタイプの指示がメディアデータとともにソースデバイスから受信されたかどうかを判定する (1 0 2 0)。ソースデバイスから指示が受信された場合 (1 0 2 0 の Y E S 分岐)、シンクデバイス 6 6 0 のデコーダ 6 8 2 は、ソースデバイスからの指示に基づいて、メディアデータ用のアプリケーションのタイプを知る (1 0 3 0)。この場合、W D システム内のソースデバイスは、メディアデータ用のアプリケーションのタイプを検出し、メディアデータ用のアプリケーションのタイプの指示とともにメディアデータをシンクデバイス 6 6 0 に送信する。場合によっては、たとえば、シンクデバイス 6 6 0 は、ビデオ再生アプリケーション用のメディアデータのストリームの指示をソースデバイスから受信することができ、シンクデバイス 6 6 0 はまた、ビデオ再生アプリケーション用のメディアデータのストリームの終了の指示をソースデバイスから受信することができる。

40

50

【 0 1 3 5 】

[0145]ソースデバイスから指示が受信されなかった場合（ 1 0 2 0 の N O 分岐）、シンクデバイス 6 6 0 のデコーダ 6 8 2 は、メディアデータ内のオーディオタイムスタンプの存在または不在に基づいて、メディアデータ用のアプリケーションのタイプを知る（ 1 0 4 0 ）。たとえば、デコーダ 6 8 2 は、メディアデータ内の一定の間隔でオーディオタイムスタンプを検出することによって、メディアデータがビデオ再生アプリケーション用であると判断することができる。別の例として、デコーダ 6 8 2 は、受信されたメディアデータ内の一定の間隔でのオーディオタイムスタンプの不在に基づいて、メディアデータが U I アプリケーションまたはゲームアプリケーション用であると判断することができる。

【 0 1 3 6 】

[0146]受信されたメディアデータ用のアプリケーションのタイプが U I アプリケーションまたはゲームアプリケーションであり、ビデオ再生アプリケーションではないことをシンクデバイス 6 6 0 のデコーダ 6 8 2 が知ったとき（ 1 0 5 0 の N O 分岐）、パイプラインマネージャ 6 3 8 は、U I アプリケーションまたはゲームアプリケーション内の表示の前に、処理パイプライン 6 5 0 内でメディアデータを保持するバッファのサイズを減少させるように、処理パイプライン 6 5 0 を構成する（ 1 0 6 0 ）。たとえば、パイプラインマネージャ 6 3 8 は、バッファを通り抜けるメディアデータの待ち時間を削減するためにレンダリングキュー 6 9 4 のサイズおよび / またはフレームバッファ 6 9 6 のサイズを減少させることができる。U I アプリケーションまたはゲームアプリケーションの場合、W D システム内の主な関心事は低遅延のユーザ体験を提供することであり、ビデオの再生品質はより低い関心事である。

【 0 1 3 7 】

[0147]受信されたメディアデータ用のアプリケーションのタイプがビデオ再生アプリケーションであることをシンクデバイス 6 6 0 のデコーダ 6 8 2 が知ったとき（ 1 0 5 0 の Y E S 分岐）、パイプラインマネージャ 6 3 8 は、ビデオ再生アプリケーション内の表示の前に、処理パイプライン 6 5 0 内でメディアデータを保持するバッファのサイズを増大させることができる（ 1 0 7 0 ）。たとえば、パイプラインマネージャ 6 3 8 は、ビデオ再生アプリケーション内のビデオ再生の品質（すなわち、滑らかさ）を改善するためにレンダリングキュー 6 9 4 のサイズおよび / またはフレームバッファ 6 9 6 のサイズを増大させることができる。バッファサイズを増大させると待ち時間が増大する場合があるが、ビデオ再生アプリケーションの場合の主な関心事は、増大した待ち時間の何らかのトレードオフを伴う高い品質のビデオ再生ユーザ体験を提供することである。

【 0 1 3 8 】

[0148]レンダラ 6 8 4 は、メディアデータ用に示されたアプリケーションのタイプでの使用のために、レンダリングキュー 6 9 4 からのメディアデータを処理パイプライン 6 5 0 内のフレームバッファ 6 9 6 内にレンダリングする（ 1 0 8 0 ）。次いで、ディスプレイプロセッサ 6 3 5 は、フレームバッファ 6 9 6 からのレンダリングされたメディアデータを、シンクデバイス 6 6 0 で動作するメディアデータ用のアプリケーション内で表示する（ 1 0 9 0 ）。

【 0 1 3 9 】

[0149]場合によっては、メディアデータがビデオ再生アプリケーション用であるとき、パイプラインマネージャ 6 3 8 は、シンクデバイス 6 6 0 がメディアデータのさらなるサンプルを受信するために、レンダリングキュー 6 9 4 からのメディアデータのレンダリングを休止するように、レンダラ 6 8 4 に指示することができる。次いで、パイプラインマネージャ 6 3 8 は、メディアデータのしきい値の数のサンプルがレンダリングキュー 6 9 4 に記憶された後、レンダリングキュー 6 9 4 からのメディアデータのレンダリングを再開するようにレンダラ 6 8 4 に指示する。このようにして、ビデオ再生の品質または滑らかさは、レンダリングされる用意ができたメディアデータの複数のサンプルをレンダリングキュー 6 9 4 内に有することによって、さらに改善され得る。

【 0 1 4 0 】

[0150]図 1 1 は、W D システム内のオーディオデータの優先トランスポートをサポートすることが可能なソースデバイスおよびシンクデバイスの例示的な動作を示すフローチャートである。図示された動作は、図 5 からのソースデバイス 5 2 0 内の処理パイプライン 5 5 0 に関して、かつ図 6 からのシンクデバイス 6 6 0 に含まれる処理パイプライン 6 5 0 に関して記載される。他の例では、図示された動作は、図 2 からのソースデバイス 2 2 0、図 3 からのシンクデバイス 3 6 0、図 1 からのソースデバイス 1 2 0 およびシンクデバイス 1 6 0、または W D システムの他のソースデバイスおよびシンクデバイスによって実行され得る。

【 0 1 4 1 】

[0151]ソースデバイス 5 2 0 は、最初に W D システム内のシンクデバイス 6 6 0 と通信セッションを確立する (1 1 0 0)。ソースデバイス 5 2 0 およびシンクデバイス 6 6 0 は、能力交渉に基づいて通信セッションを確立することができる。本開示の技法によれば、オーディオデータの優先トランスポートを提供するために、ソースデバイス 5 2 0 内のパイプラインマネージャ 5 3 8 は、処理パイプライン 5 5 0 内のビデオ経路よりも多くのバッファリングを含むように、処理パイプライン 5 5 0 内のオーディオ経路を構成する (1 1 1 0)。加えて、ソースデバイス 5 2 0 内のパイプラインマネージャ 5 3 8 は、ワイヤレスモデムソケット 5 7 0 内のビデオトランスポートキューよりも高い優先度を有するワイヤレスモデムソケット 5 7 0 内のオーディオトランスポートキューを構成する (1 1 2 0)。

【 0 1 4 2 】

[0152]オーディオデータ用の優先トランスポート経路が構成されると、ソースデバイス 5 2 0 は、別々の処理パイプライン経路内で、オーディオデータとビデオデータの処理を開始する (1 1 3 0)。オーディオデータとビデオデータが処理されると、ワイヤレスモデム 5 3 4 は、ソケット 5 7 0 内の別々のトランスポートキューから、オーディオデータと関連するビデオデータとをシンクデバイス 6 6 0 にトランスポートする (1 1 4 0)。

【 0 1 4 3 】

[0153]シンクデバイス 6 6 0 は、W D システム内の通信チャネルを介してソースデバイス 5 2 0 から、オーディオデータと関連するビデオデータとを受信する (1 1 5 0)。シンクデバイス 6 6 0 のプロセッサ 6 3 1 は、受信されたメディアデータに基づいて、通信チャネルのトランスポート状態を判断することができる (1 1 6 0)。たとえば、シンクデバイス 6 6 0 のプロセッサ 6 3 1 は、通信チャネルを介したトランスポート後にメディアデータについての誤り率とパケット損失の量とを判断することができる。シンクデバイス 6 6 0 のトランスポートユニット 6 3 1 は、パケットを生成して、通信チャネルのトランスポート状態を記述するフィードバック情報を、シンクデバイス 6 6 0 からソースデバイス 5 2 0 に送信することができる (1 1 7 0)。たとえば、ワイヤレスモデム 6 3 4 は、R T C P を使用して、U I B C または他のフィードバックチャネルを介してシンクデバイス 6 6 0 からソースデバイス 6 6 0 にパケットを送信することができる。

【 0 1 4 4 】

[0154]通信チャネルのトランスポート状態を記述するフィードバック情報を受信すると、ソースデバイス 5 2 0 のパイプラインマネージャ 5 3 8 は、シンクデバイス 6 6 0 からのフィードバック情報に基づいて、処理パイプライン 5 5 0 を修正することができる (1 1 8 0)。処理パイプライン 5 5 0 が通信チャネル用に構成されると、ソースデバイス 5 2 0 は、別々の処理パイプライン経路内でのオーディオデータとビデオデータの処理 (1 1 3 0) と、ソケット 5 7 0 内の別々のトランスポートキューからのオーディオデータおよび関連するビデオデータのシンクデバイス 6 6 0 へのトランスポート (1 1 4 0) とを再開する。

【 0 1 4 5 】

[0155] 1 つまたは複数の例では、記載された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せに実装することができる。ソフトウェアに実装される場合、機能は、 1 つまたは複数の命令またはコードとしてコンピュータ可読媒体

10

20

30

40

50

上に記憶されるか、あるいはコンピュータ可読媒体を介して送信することができる。コンピュータ可読媒体は、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む、コンピュータデータ記憶媒体またはコンピュータデータ通信媒体を含むことができる。いくつかの例では、コンピュータ可読媒体は、非一時的コンピュータ可読媒体を備えることができる。データ記憶媒体は、本開示に記載された技法の実装のための命令、コードおよび/またはデータ構造を取り出すために、1つもしくは複数のコンピュータまたは1つもしくは複数のプロセッサによってアクセスできる任意の利用可能な媒体であり得る。

【0146】

[0156]限定ではなく例として、そのようなコンピュータ可読媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気ストレージデバイス、フラッシュメモリなどの非一時的媒体、または、命令もしくはデータ構造の形態の所望のプログラムコードを搬送もしくは記憶するために使用できる、コンピュータによってアクセスできる、任意の他の媒体を備えることができる。また、いかなる接続もコンピュータ可読媒体と適切に呼ばれる。たとえば、ソフトウェアが、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。本明細書で使用するディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)およびブルーレイディスク(disc)を含み、ディスク(disk)は、通常、データを磁氣的に再生し、ディスク(disc)は、データをレーザーで光学的に再生する。上記の組合せもコンピュータ可読媒体の範囲内に含まれるべきである。

【0147】

[0157]コードは、1つもしくは複数のデジタル信号プロセッサ(DSP)などの1つもしくは複数のプロセッサ、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブル論理アレイ(FPGA)、または他の等価な集積回路もしくはディスクリート論理回路によって実行することができる。したがって、本明細書で使用する「プロセッサ」という用語は、前述の構造、または本明細書に記載された技法の実装に適した他の構造のいずれをも指す。加えて、いくつかの態様では、本明細書に記載された機能は、符号化および復号のために構成された専用のハードウェアおよび/もしくはソフトウェアモジュールの内部に提供できるか、または複合コーデックに組み込むことができる。また、本技法は、1つまたは複数の回路または論理要素の中に完全に実装することができる。

【0148】

[0158]本開示の技法は、ワイヤレスハンドセット、集積回路(IC)またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置に実装することができる。開示された技法を実行するように構成されたデバイスの機能的態様を強調するために、様々な構成要素、モジュール、またはユニットが本開示に記載されたが、それらの構成要素、モジュール、またはユニットは、必ずしも異なるハードウェアユニットによって実現する必要があるとは限らない。むしろ、上述されたように、様々なユニットは、適切なソフトウェアおよび/またはファームウェアとともに、上述された1つまたは複数のプロセッサを含む、コーデックハードウェアユニットにおいて組み合わされるか、または相互動作ハードウェアユニットの集合によって提供することができる。

【0149】

[0159]本発明の様々な実施形態が記載された。これらおよび他の実施形態は、以下の特許請求の範囲内に入る。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

[C 1] ワイヤレスディスプレイ (W D) システムにおいて 1 つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、

前記ソースデバイスでメディアデータのフレーム更新を取り込むことと、

前記ソースデバイスの処理パイプラインに含まれる最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも直近の 1 つをバッファリングすることと、

ハードウェア加速を使用して前記最小サイズのバッファ内の前記フレーム更新を取り出すことと、

前記 1 つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファから取り出された前記フレーム更新を処理することとを備える、方法。

10

[C 2] メディアデータのフレーム更新を取り込むことが、前記ソースデバイスのメモリにフレーム更新を記憶することを備える、C 1 に記載の方法。

[C 3] 前記フレーム更新のうちの少なくとも直近の 1 つをバッファリングすることが、前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つへのポインタのうちの 1 つをバッファリングすることを備える、C 2 に記載の方法。

[C 4] 前記フレーム更新のうちの少なくとも直近の 1 つをバッファリングすることが、

前記最小サイズのバッファが一杯であるとき、前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除することと、

20

前記最小サイズのバッファに前記フレーム更新のうちの前記直近の 1 つをバッファリングすることとを備える、C 1 に記載の方法。

[C 5] ハードウェア加速を使用して前記フレーム更新を取り出すことが、前記ソースデバイスの中央処理装置 (C P U) を使用せずに、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させることを備える、C 1 に記載の方法。

[C 6] 前記 W D システム内で前記通信セッションを確立すると、前記最小サイズのバッファを含むように、および前記処理パイプライン内の待ち時間を削減するようにハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成すること

30

をさらに備える、C 1 に記載の方法。

[C 7] 前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にビデオ処理エンジンとマルチメディアディスプレイプロセッサとの間の 2 つのライトバック (W B) バッファを備える、C 1 に記載の方法。

[C 8] 前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にエンコーダとビデオ処理エンジンとの間の 4 つのフレームバッファを備える、C 1 に記載の方法。

[C 9] 前記処理パイプラインを用いて前記フレーム更新を処理することが、

前記処理パイプラインに含まれる前記 W D システム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、

40

前記 1 つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを備える、C 1 に記載の方法。

[C 1 0] 前記ソースデバイスのコード化ピクチャバッファに、前記 1 つまたは複数のシンクデバイスへの送信より前に、前記符号化されたフレーム更新をバッファリングすることをさらに備える、C 9 に記載の方法。

[C 1 1] 前記符号化されたフレーム更新をバッファリングすることが、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングすることを備える、C 1 0 に記載の方法。

[C 1 2] 前記 1 つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信

50

することと、

新しいフレーム更新を受信せずにある時間期間後に、前記1つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとをさらに備える、C1に記載の方法。

[C13] 前記1つまたは複数のシンクデバイスへのビデオデータトランスポートよりもオーディオデータトランスポートを優先させるように、前記ソースデバイスの前記処理パイプラインを構成することとをさらに備える、C1に記載の方法。

[C14] 前記シンクデバイスのうちの少なくとも1つからトランスポートチャネル状態を記述するフィードバック情報を受信することと、

前記フィードバック情報に基づいて前記ソースデバイスの前記処理パイプラインを修正することとをさらに備える、C1に記載の方法。

[C15] 前記ソースデバイスで前記通信セッションを確立することが、前記通信セッション用の前記メディアデータを複数のシンクデバイスに広告することと、前記1つまたは複数のシンクデバイスから前記通信セッションへの参加の要求を受信することとを備える、C1に記載の方法。

[C16] ワイヤレスディスプレイ(WD)システムにおいて1つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立することと、メディアデータのフレーム更新を取り込むことと、処理パイプラインに含まれる最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、前記1つまたは複数のシンクデバイスへの送信用に前記最小サイズのバッファからの前記フレーム更新を処理することとを行うように構成された、1つまたは複数の処理ユニットを含む前記処理パイプラインと、

前記処理パイプラインによる処理用に前記最小サイズのバッファから前記フレーム更新を取り出すように構成されたハードウェア加速器とを備える、ソースデバイス。

[C17] 前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記ソースデバイスのメモリにフレーム更新を記憶するように構成された、C16に記載のソース。

[C18] 前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記メモリに記憶された前記フレーム更新のうちの前記直近の1つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の1つへのポインタのうちの1つをバッファリングするように構成された、C17に記載のソースデバイス。

[C19] 前記処理パイプラインの前記1つまたは複数の処理ユニットが、前記最小サイズのバッファが一杯であるとき、前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除することと、

前記最小サイズのバッファに前記フレーム更新のうちの前記直近の1つをバッファリングすることとを行うように構成された、C16に記載のソースデバイス。

[C20] 前記ハードウェア加速器が、前記ソースデバイスの中央処理装置(CPU)を使用せずに、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させるように構成された、C16に記載のソースデバイス。

[C21] 前記処理パイプライン用のパイプラインマネージャをさらに備え、前記WDシステム内で前記通信セッションを確立すると、前記パイプラインマネージャが、前記最小サイズのバッファを含むように、および前記処理パイプライン内の待ち時間を削減するようにハードウェア加速を使用するように前記処理パイプラインを構成する、C16に記載のソースデバイス。

[C22] 前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にビデオ処理エンジンとマルチメディアディスプレイプロセッサとの間の2つのライトバック(WB)バッファを備える、C16に記載のソースデバイス。

[C23] 前記最小サイズのバッファが、前記ソースデバイスの前記処理パイプライン内にエンコーダとビデオ処理エンジンとの間の4つのフレームバッファを備える、C16に記載のソースデバイス。

10

20

30

40

50

[C 2 4] 前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記処理パイプラインに含まれる前記 W D システム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、

前記 1 つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを行うように構成された、C 1 6 に記載のソースデバイス。

[C 2 5] 前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記ソースデバイスのコード化ピクチャバッファに、前記 1 つまたは複数のシンクデバイスへの送信より前に、前記符号化されたフレーム更新をバッファリングするように構成された、C 2 4 に記載のソースデバイス。

[C 2 6] 前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングするように構成された、C 2 5 に記載のソースデバイス。

[C 2 7] 前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記 1 つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信することと、

新しいフレーム更新を受信せずにある時間期間後に、前記 1 つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとを行うように構成された、C 1 6 に記載のソースデバイス。

[C 2 8] 前記処理パイプライン用のパイプラインマネージャをさらに備え、ここにおいて、前記パイプラインマネージャが、前記 1 つまたは複数のシンクデバイスへのビデオデータトランスポートよりもオーディオデータトランスポートを優先させるように前記処理パイプラインを構成する、C 1 6 に記載のソースデバイス。

[C 2 9] 前記処理パイプライン用のパイプラインマネージャをさらに備え、ここにおいて、前記パイプラインマネージャが、

前記シンクデバイスのうちの少なくとも 1 つからトランスポートチャネル状態を記述するフィードバック情報を受信することと、

前記フィードバック情報に基づいて前記ソースデバイスの前記処理パイプラインを修正することとを行う、C 1 6 に記載のソースデバイス。

[C 3 0] 前記処理パイプラインの前記 1 つまたは複数の処理ユニットが、前記通信セッション用の前記メディアデータを複数のシンクデバイスに広告することと、前記ソースデバイスで前記通信セッションを確立するために前記 1 つまたは複数のシンクデバイスから前記通信セッションへの参加の要求を受信することとを行うように構成された、C 1 6 に記載のソースデバイス。

[C 3 1] ワイヤレスディスプレイ (W D) システムにおいて 1 つまたは複数のシンクデバイスとソースデバイスで通信セッションを確立するための手段と、

メディアデータのフレーム更新を取り込むための手段と、

前記ソースデバイスの処理パイプラインに含まれる最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも直近の 1 つをバッファリングするための手段と、

ハードウェア加速を使用して前記最小サイズのバッファから前記フレーム更新を取り出すための手段と、

前記 1 つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファから取り出された前記フレーム更新を処理するための手段とを備える、ソースデバイス。

[C 3 2] 前記ソースデバイスのメモリにフレーム更新を記憶するための手段をさらに備える、C 3 1 に記載のソースデバイス。

[C 3 3] 前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の 1 つへのポインタのうちの 1 つをバッファリングするための手段をさらに備える、C 3 2 に記載のソースデバイス。

10

20

30

40

50

〔C 3 4〕 前記最小サイズのバッファが一杯であるとき、前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除するための手段と、

前記最小サイズのバッファに前記フレーム更新のうちの前記直近の1つをバッファリングするための手段とをさらに備える、C 3 1に記載のソースデバイス。

〔C 3 5〕 前記ソースデバイスの中央処理装置（CPU）を使用せずに、ハードウェア加速を使用して、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させるための手段をさらに備える、C 3 1に記載のソースデバイス。

〔C 3 6〕 前記WDシステム内で前記通信セッションを確立すると、前記最小サイズのバッファを含むように、および前記処理パイプライン内の待ち時間を削減するようにハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成するための手段をさらに備える、C 3 1に記載のソースデバイス。

〔C 3 7〕 前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングするための手段と、

前記1つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化するための手段とをさらに備える、C 3 1に記載のソースデバイス。

〔C 3 8〕 前記ソースデバイスのコード化ピクチャバッファに、前記1つまたは複数のシンクデバイスへの送信より前に、前記符号化されたフレーム更新をバッファリングするための手段をさらに備え、ここにおいて、前記符号化されたフレーム更新をバッファリングするための前記手段が、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングするための手段を備える、C 3 7に記載のソースデバイス。

〔C 3 9〕 前記1つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信するための手段と、

新しいフレーム更新を受信せずにある時間期間後に、前記1つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信するための手段とをさらに備える、C 3 1に記載のソースデバイス。

〔C 4 0〕 ソースデバイス内で実行されると、

ワイヤレスディスプレイ（WD）システムにおいて1つまたは複数のシンクデバイスと前記ソースデバイスで通信セッションを確立することと、

前記ソースデバイスでメディアデータのフレーム更新を取り込むことと、

前記ソースデバイスの処理パイプラインに含まれる最小サイズのバッファに前記取り込まれたフレーム更新のうちの少なくとも直近の1つをバッファリングすることと、

前記1つまたは複数のシンクデバイスへの送信用に前記処理パイプラインを用いて、前記最小サイズのバッファからの前記フレーム更新を処理することであって、ここにおいて、前記フレーム更新がハードウェア加速を使用して前記最小サイズのバッファから取り出される、処理することと、をプログラマブルプロセッサに行わせる命令を備える、コンピュータ可読媒体。

〔C 4 1〕 前記命令が前記プログラマブルプロセッサに、前記ソースデバイスのメモリにフレーム更新を記憶することを行わせる、C 4 0に記載のコンピュータ可読媒体。

〔C 4 2〕 前記命令が前記プログラマブルプロセッサに、前記メモリに記憶された前記フレーム更新のうちの前記直近の1つのメモリコピー、または前記メモリに記憶された前記フレーム更新のうちの前記直近の1つへのポインタのうちの1つをバッファリングすることを行わせる、C 4 1に記載のコンピュータ可読媒体。

〔C 4 3〕 前記命令が前記プログラマブルプロセッサに、

前記最小サイズのバッファが一杯であるとき、前記最小サイズのバッファから前記メディアデータのより古いフレーム更新を削除することと、

前記最小サイズのバッファに前記フレーム更新のうちの前記直近の1つをバッファリングすることとを行わせる、C 4 0に記載のコンピュータ可読媒体。

〔C 4 4〕 前記命令が前記プログラマブルプロセッサに、前記ソースデバイスの中央処

10

20

30

40

50

理装置（CPU）を使用せずに、ハードウェア加速を使用して、前記ソースデバイスの前記処理パイプライン内の処理ユニットと前記最小サイズのバッファとの間で前記フレーム更新を移動させることを行わせる、C 4 0に記載のコンピュータ可読媒体。

〔C 4 5〕 前記WDシステム内で前記通信セッションを確立すると、前記最小サイズのバッファを含むように、および前記処理パイプライン内の待ち時間を削減するようにハードウェア加速を使用するように前記ソースデバイスの前記処理パイプラインを構成することを前記プログラマブルプロセッサに行わせる命令をさらに備える、C 4 0に記載のコンピュータ可読媒体。

〔C 4 6〕 前記命令が前記プログラマブルプロセッサに、前記処理パイプラインに含まれる前記WDシステム用のフレームバッファ内に前記フレーム更新をレンダリングすることと、

前記1つまたは複数のシンクデバイスへの送信用に前記レンダリングされたフレーム更新を符号化することとを行わせる、C 4 0に記載のコンピュータ可読媒体。

〔C 4 7〕 前記ソースデバイスのコード化ピクチャバッファに、前記1つまたは複数のシンクデバイスへの送信より前に、前記符号化されたフレーム更新をバッファリングすることを前記プログラマブルプロセッサに行わせる命令をさらに備え、ここにおいて、前記命令が前記プログラマブルプロセッサに、前記符号化されたフレーム更新のメモリコピーを作成せずに、前記ソースデバイスのメモリに記憶された前記符号化されたフレーム更新へのポインタをバッファリングすることをさらに行わせる、C 4 6に記載のコンピュータ可読媒体。

〔C 4 8〕 前記1つまたは複数のシンクデバイスに前記処理されたフレーム更新を送信することと、

新しいフレーム更新を受信せずにある時間期間後に、前記1つまたは複数のシンクデバイスに前記ソースデバイスから前記処理されたフレーム更新を再送信することとを前記プログラマブルプロセッサに行わせる命令をさらに備える、C 4 0に記載のコンピュータ可読媒体。

10

20

【図 1】

図 1

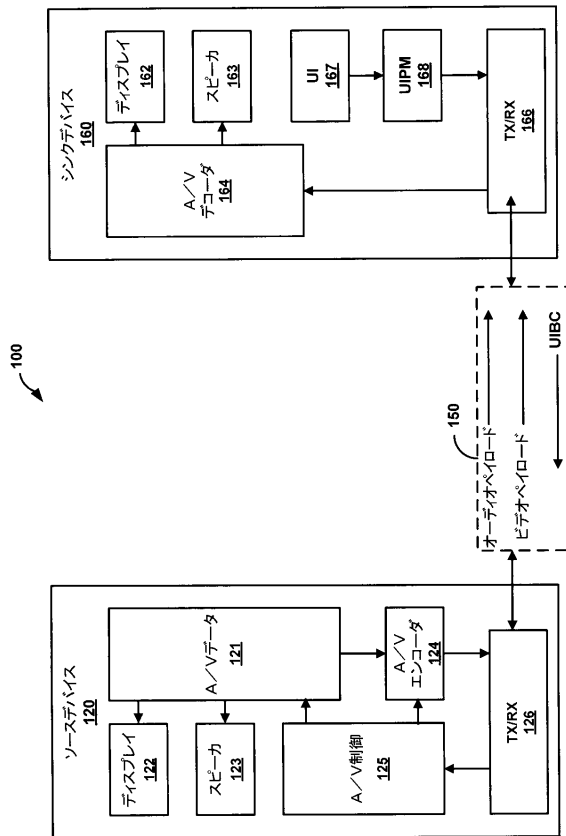


FIG. 1

【図 2】

図 2

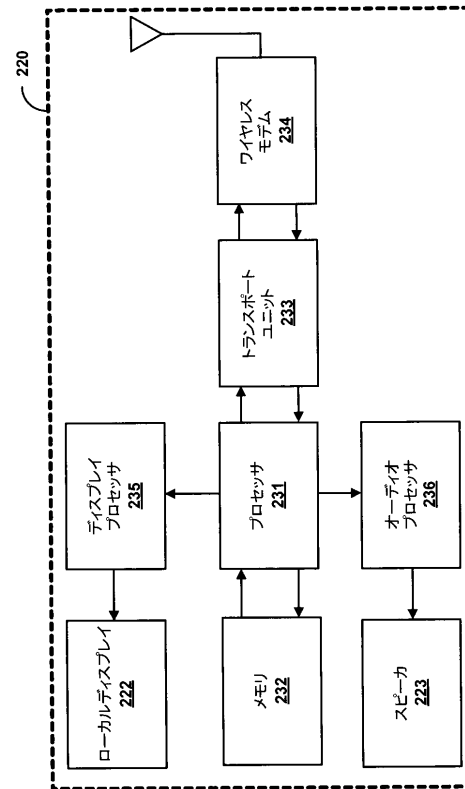


FIG. 2

【図 3】

図 3

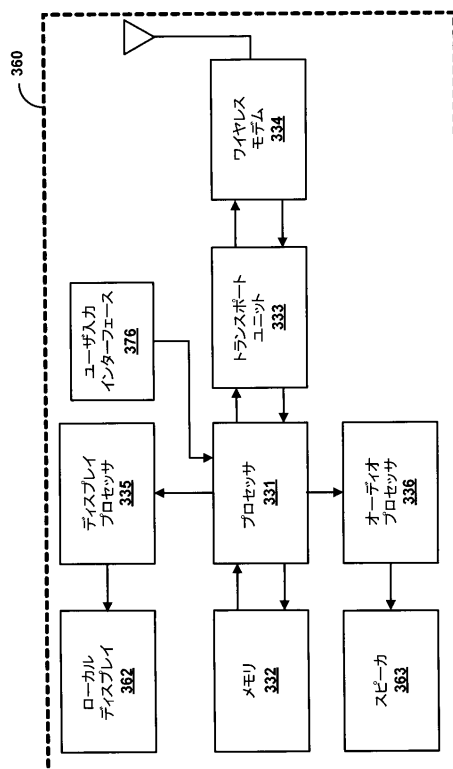


FIG. 3

【図 4】

図 4

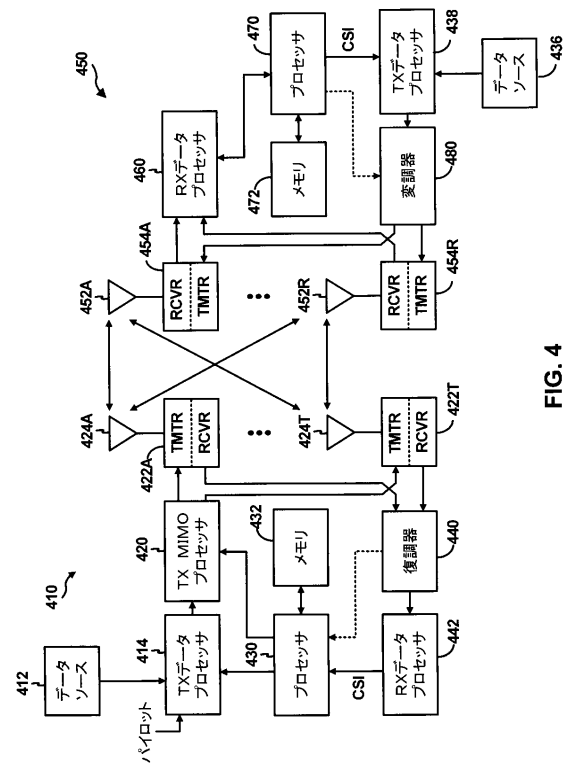


FIG. 4

【図 5】

図 5

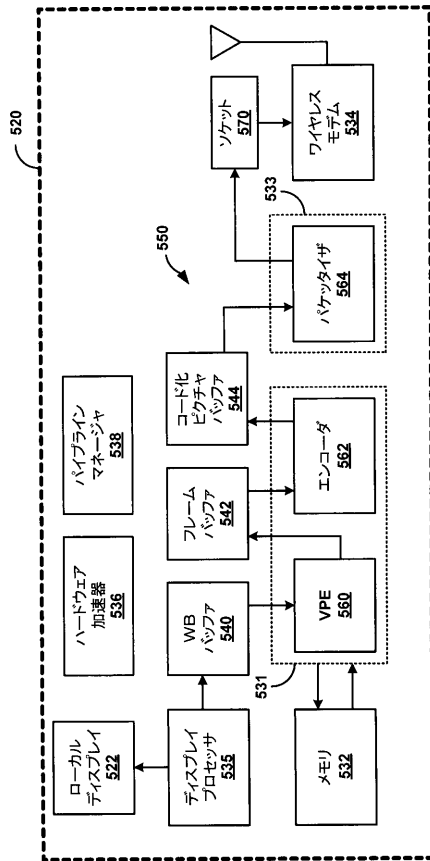


FIG. 5

【図 6】

図 6

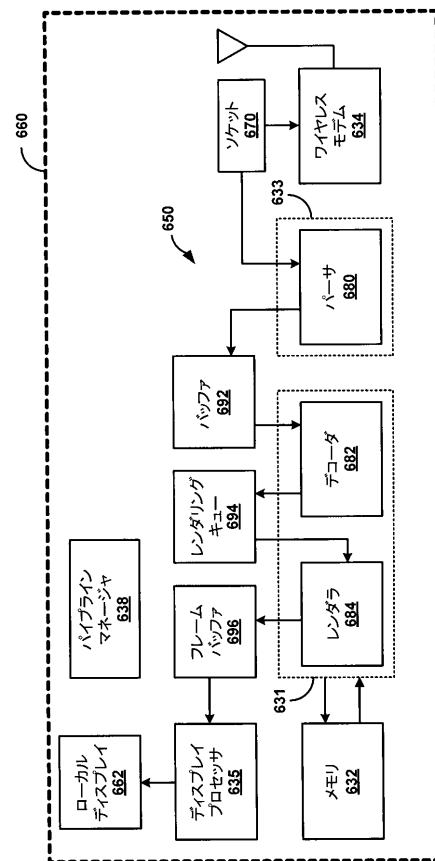


FIG. 6

【図 7】

図 7

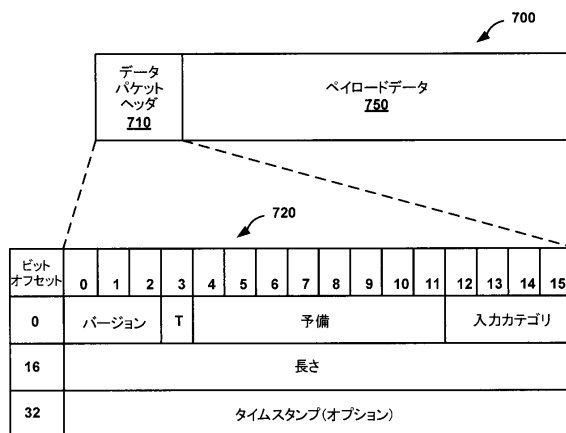


FIG. 7

【図 8】

図 8

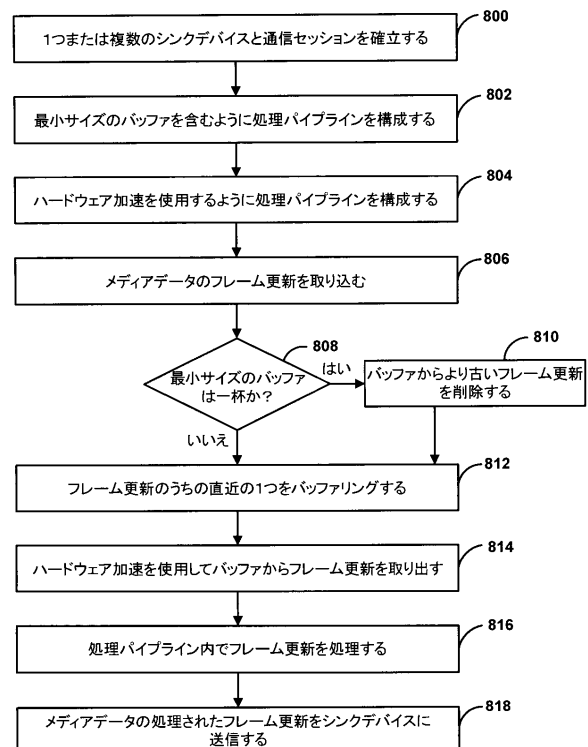


FIG. 8

【図 9】

図 9

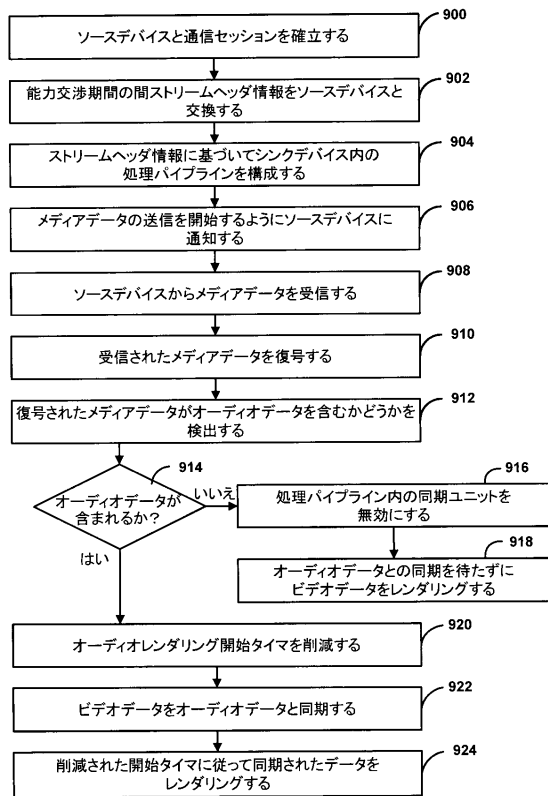


FIG. 9

【図 10】

図 10

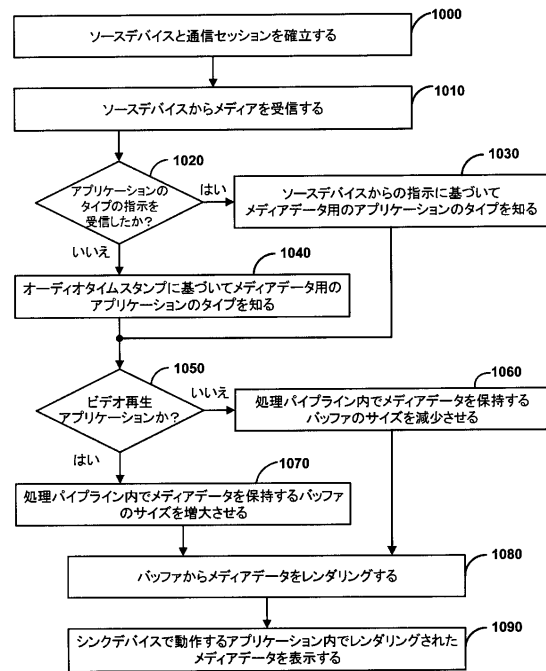


FIG. 10

【図 11】

図 11

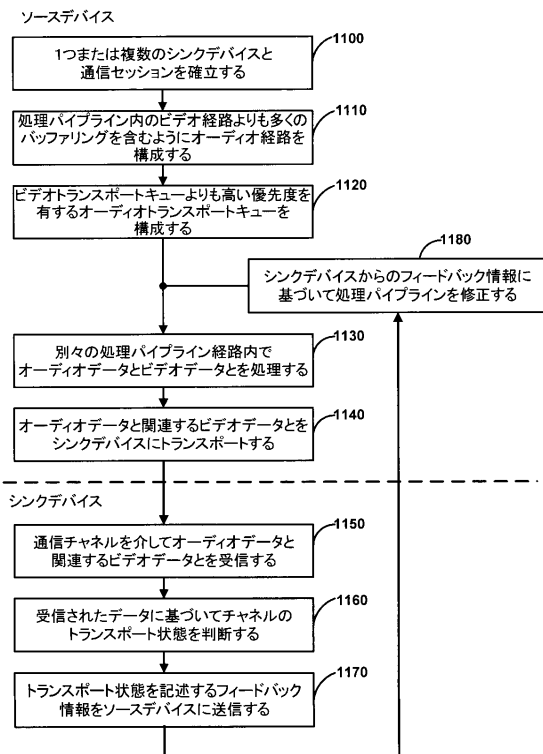


FIG. 11

フロントページの続き

- (31)優先権主張番号 61/604,090
 (32)優先日 平成24年2月28日(2012.2.28)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 61/604,094
 (32)優先日 平成24年2月28日(2012.2.28)
 (33)優先権主張国 米国(US)
 (31)優先権主張番号 13/633,328
 (32)優先日 平成24年10月2日(2012.10.2)
 (33)優先権主張国 米国(US)

早期審査対象出願

- (74)代理人 100153051
 弁理士 河野 直樹
 (74)代理人 100140176
 弁理士 砂川 克
 (74)代理人 100158805
 弁理士 井関 守三
 (74)代理人 100179062
 弁理士 井上 正
 (74)代理人 100124394
 弁理士 佐藤 立志
 (74)代理人 100112807
 弁理士 岡田 貴志
 (74)代理人 100111073
 弁理士 堀内 美保子
 (72)発明者 ワン、シャオドン
 アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 5775
 (72)発明者 シャウカット、ファワード
 アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 5775
 (72)発明者 ラビーン・ドラン、ピジャヤラクシュミ・アール
 アメリカ合衆国、カリフォルニア州 92121、サン・ディエゴ、モアハウス・ドライブ 5775

審査官 古川 哲也

- (56)参考文献 米国特許出願公開第2003/0027517(US,A1)
 米国特許出願公開第2011/0010607(US,A1)
 特開平02-203689(JP,A)
 米国特許出願公開第2011/0145879(US,A1)
 国際公開第2011/075468(WO,A1)

(58)調査した分野(Int.Cl., DB名)

H04N 21/00 - 21/858
 H04N 19/00 - 19/98
 G06F 15/00

G 0 6 F 1 3 / 0 0