

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4519082号
(P4519082)

(45) 発行日 平成22年8月4日(2010.8.4)

(24) 登録日 平成22年5月28日(2010.5.28)

(51) Int. Cl.			F I		
HO4N	5/92	(2006.01)	HO4N	5/92	H
HO4N	7/08	(2006.01)	HO4N	7/08	Z
HO4N	7/081	(2006.01)	HO4N	7/13	Z
HO4N	7/26	(2006.01)	GO6F	9/46	452G
GO6F	9/48	(2006.01)			

請求項の数 18 (全 16 頁)

(21) 出願番号	特願2006-37890 (P2006-37890)
(22) 出願日	平成18年2月15日(2006.2.15)
(65) 公開番号	特開2007-221323 (P2007-221323A)
(43) 公開日	平成19年8月30日(2007.8.30)
審査請求日	平成19年2月7日(2007.2.7)

(73) 特許権者	395015319 株式会社ソニー・コンピュータエンタテインメント 東京都港区南青山二丁目6番21号
(74) 代理人	100105924 弁理士 森下 賢樹
(72) 発明者	山本 竜司 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内

審査官 清水 正一

最終頁に続く

(54) 【発明の名称】 情報処理方法、動画サムネイル表示方法、復号化装置、および情報処理装置

(57) 【特許請求の範囲】

【請求項1】

複数の入力データストリームのそれぞれに所定の処理を施して生成した複数の出力データを並列に出力する情報処理方法であって、

各入力データストリームに第1の処理を施して中間データを生成し、当該入力データストリームに対応づけられたバッファに格納するスレッドと、

前記中間データを前記対応づけられたバッファから読み出し第2の処理を施して出力データを生成するスレッドと、を前記複数の入力データストリームのそれぞれに対して生成するステップと、

各入力データストリームに対応する、前記格納するスレッドと前記出力データを生成するスレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理するステップと、

を含み、

前記処理するステップは、

一の入力データストリームに対応する前記格納するスレッドの処理開始時に、当該入力データストリームに対応づけられたバッファに格納され、読み出されていない前記中間データの量を検出するステップと、

前記中間データの量が当該バッファの容量より小さい規定値より大きいとき、当該格納するスレッドの処理を休止し、前記処理待ち行列に格納したうえ、別の入力データストリームに対応する前記格納するスレッドを含む前記処理待ち行列から、次に処理すべきスレ

10

20

ッドを取得して処理を開始するステップと、
を含むことを特徴とする情報処理方法。

【請求項 2】

前記処理するステップは、一の入力データストリームに対応づけられたバッファに格納され、読み出されていない前記中間データの量が前記規定値より大きいとき、当該格納するスレッドのイールド処理を行い、別の入力データストリームに対応づけられたバッファに前記中間データを格納するスレッドの処理回数を増加させることを特徴とする請求項 1 に記載の情報処理方法。

【請求項 3】

前記検出するステップは、前記処理するステップにおける、前記格納するスレッド内の処理において実施し、前記格納するスレッドは、前記中間データの量が前記規定値より大きいとき、自らの処理を休止して他のスレッドへ処理を譲るイールド処理を行うことを特徴とする請求項 1 または 2 に記載の情報処理方法。

10

【請求項 4】

各入力データストリームに対応する格納するスレッドはその他のスレッドとともに、オペレーティングシステムが決定したスケジュールおよび優先度に基づく順序で、前記処理待ち行列に格納されたうえ前記処理するステップにおいて順次処理が開始されることを特徴とする請求項 1 から 3 のいずれかに記載の情報処理方法。

【請求項 5】

前記入力データストリームごとに生成した前記格納するスレッドは、処理の再開に対して同一の優先度を有することを特徴とする請求項 1 から 4 のいずれかに記載の情報処理方法。

20

【請求項 6】

前記処理するステップは、前記中間データの量が前記規定値以下となったとき、およびそれ以後、前記バッファに格納され、読み出されていない前記中間データが所定の量に達するまでは前記中間データの量が前記規定値より大きくても、当該格納するスレッドの処理を続行することを特徴とする請求項 1 から 5 のいずれかに記載の情報処理方法。

【請求項 7】

圧縮符号化された複数の動画ファイルをそれぞれ復号化して描画した複数の動画を並列に表示する動画サムネイル表示方法であって、

30

各動画ファイルを復号化して生成したテキストデータを、当該動画ファイルに対応づけられたテキストバッファに格納する処理を行う復号化スレッドと、前記テキストデータを前記対応づけられたテキストバッファから読み出し、画像フレームデータを生成して対応するフレームバッファに格納する処理を行う描画スレッドとを前記複数の動画ファイルのそれぞれに対し生成するステップと、

各動画ファイルに対応する、前記復号化スレッドと前記描画スレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理するステップと、

複数の前記動画ファイルの前記画像フレームデータを、対応する前記フレームバッファからそれぞれ読み出し、並列に表示装置に出力するステップと、

を含み、

40

前記処理するステップは、

一の動画ファイルに対応する前記復号化スレッドの処理開始時に、当該動画ファイルに対応づけられたテキストバッファに格納され、読み出されていない前記テキストデータの量を検出するステップと、

前記テキストデータの量が当該テキストバッファの容量より小さい規定値より大きいとき、前記復号化スレッドの復号化に係る処理を休止し、当該スレッドを前記処理待ち行列に格納したうえ、別の動画ファイルに対応する前記復号化スレッドを含む前記処理待ち行列から、次に処理すべきスレッドを取得して処理を開始するステップと、

を含むことを特徴とする動画サムネイル表示方法。

【請求項 8】

50

複数のデータストリームのそれぞれに所定の処理を施して生成した複数の出力データを並列に出力する復号化装置であって、

前記複数のデータストリームを入力する入力部と、

前記入力部に入力した各データストリームに第 1 の処理を施して中間データを生成し、当該データストリームに対応して設けられたバッファに一時保存するスレッドと、前記対応して設けられたバッファに保存された前記中間データを読み出し、第 2 の処理を施して出力データを生成するスレッドと、を前記複数のデータストリームのそれぞれに対し生成し、各データストリームに対応する、前記一時保存するスレッドと前記出力データを生成するスレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理するプロセッサユニットと、

前記プロセッサユニットがデータストリームごとに生成した出力データを並列に出力する出力部と、

を備え、

前記プロセッサユニットは、一のデータストリームに対応する前記一時保存するスレッドの処理開始時に、当該データストリームに対応して設けられたバッファに格納され、読み出されていない前記中間データの量を検出し、前記中間データの量が当該バッファの容量より小さい規定値より大きいときは、当該一時保存するスレッドの処理を休止し、前記処理待ち行列に格納したうえ、別のデータストリームに対応する前記一時保存するスレッドを含む前記処理待ち行列から、次に処理すべきスレッドを取得して処理を開始することを特徴とする復号化装置。

【請求項 9】

前記プロセッサユニットは、一のデータストリームに対応して設けられたバッファに格納され、読み出されていない前記中間データの量が前記規定値より大きいとき、当該一時保存するスレッドのイールド処理を行い、別のデータストリームに対応して設けられたバッファに前記中間データを一時保存するスレッドの処理回数を増加させることを特徴とする請求項 8 に記載の復号化装置。

【請求項 10】

前記プロセッサユニットは、前記中間データの量を検出する処理を、前記一時保存するスレッド内の処理において実施し、当該一時保存するスレッドは、前記読み出されていない中間データの量が前記規定値より大きいとき、自らの処理を休止して前記プロセッサユニットを開放することを特徴とする請求項 8 に記載の復号化装置。

【請求項 11】

各データストリームに対応する前記一時保存するスレッドはその他のスレッドとともに、オペレーティングシステムが決定したスケジュールおよび優先度に基づく順序で、前記処理待ち行列に格納されたうえ前記プロセッサユニットにおいて順次処理が開始されることを特徴とする請求項 8 から 10 のいずれかに記載の復号化装置。

【請求項 12】

圧縮符号化された複数の動画ファイルをそれぞれ復号化して描画した複数の動画像を並列に表示する情報処理装置であって、

前記動画ファイルのそれぞれに対応した複数のバッファと、

前記動画ファイルを復号化し、生成したテクスチャデータを対応する前記バッファに格納する復号化スレッドと、前記テクスチャデータを前記対応するバッファから読み出し、前記動画ファイルに対応した画像フレームデータを生成する描画スレッドと、を前記複数の動画ファイルのそれぞれに対し生成し、各動画ファイルに対応する、前記復号化スレッドと前記描画スレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理するプロセッサユニットと、

前記プロセッサユニットにより生成された、複数の前記動画ファイルの前記画像フレームデータを並列に表示する画像表示部と、

を備え、

前記プロセッサユニットは、一の動画ファイルに対応する前記復号化スレッドの処理開

始時に、当該動画ファイルに対応づけられたバッファに格納され、読み出されていない前記テクスチャデータの量が当該バッファの容量より小さい規定値より大きいとき、前記復号化スレッドの復号化に係る処理を休止し、当該スレッドを前記処理待ち行列に格納したうえ、別の動画ファイルに対応する前記復号化スレッドを含む前記処理待ち行列から、次に処理すべきスレッドを取得して処理を開始することを特徴とする情報処理装置。

【請求項 13】

前記プロセッサユニットは、前記テクスチャデータの量を検出する処理を、前記復号化スレッド内の処理において実施し、当該復号化スレッドは、前記読み出されていないテクスチャデータの量が前記規定値より大きいとき、自らの処理を休止して前記プロセッサユニットを開放することを特徴とする請求項 12 に記載の情報処理装置。

10

【請求項 14】

前記プロセッサユニットが処理するスレッドは、前記復号化スレッドおよび前記描画スレッド以外のスレッドを含むことを特徴とする請求項 12 または 13 に記載の情報処理装置。

【請求項 15】

前記複数の動画ファイルごとに生成された前記復号化スレッドは、前記プロセッサユニットによる処理に対し、同一の優先度を有することを特徴とする請求項 12 から 14 のいずれかに記載の情報処理装置。

【請求項 16】

前記プロセッサユニットは、前記テクスチャデータの量が前記規定値以下となったとき、およびそれ以後、前記バッファに格納され、読み出されていない前記テクスチャデータが所定の量に達するまでは前記テクスチャデータの量が前記規定値より大きくても、前記復号化スレッドの復号化に係る処理を続行することを特徴とする請求項 12 から 15 のいずれかに記載の情報処理装置。

20

【請求項 17】

複数のスレッドを切り替えて処理するコンピュータにおいて実行され、複数の入力データストリームからそれぞれ生成した複数の出力データを並列に出力する機能をコンピュータに実現させるコンピュータプログラムであって、

前記複数の入力データストリームを取得する機能と、

各入力データストリームに第 1 の処理を施して中間データを生成し、当該入力データストリームに対応づけられたバッファに格納するスレッドと、前記中間データを前記対応づけられたバッファから読み出し第 2 の処理を施して出力データを生成するスレッドと、を前記複数の入力データストリームのそれぞれに対し生成する機能と、

30

各入力データストリームに対応する、前記格納するスレッドと前記出力データを生成するスレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理する機能と、

一の入力データストリームに対応する前記格納するスレッドの処理開始時に、当該入力データストリームに対応づけられたバッファに格納され、読み出されていない前記中間データの量を検出し、前記中間データの量が当該バッファの容量より小さい規定値より大きいときは、当該格納するスレッドの処理を休止してプロセッサを開放し、別の入力データストリームに対応する前記格納するスレッドを含む他のスレッドへ処理を譲るイールド処理を行う機能と、

40

を前記コンピュータに実現させることを特徴とするコンピュータプログラム。

【請求項 18】

圧縮符号化された複数の動画ファイルをそれぞれ復号化して描画した複数の動画像を並列に表示する情報処理装置であって、

各動画ファイルを復号化して生成したテクスチャデータを、当該動画ファイルに対応づけられたテクスチャバッファに格納する処理を行う復号化スレッドと、前記テクスチャデータを前記対応づけられたテクスチャバッファから読み出し、画像フレームデータを生成して対応するフレームバッファに格納する処理を行う描画スレッドと、を前記複数の動画

50

ファイルのそれぞれに対し生成する機能と、

各動画ファイルに対応する、前記復号化スレッドと前記描画スレッドを含む複数のスレッドの処理待ち行列から順次スレッドを取得して処理する機能と、

複数の前記動画ファイルの前記画像フレームデータを、対応する前記フレームバッファからそれぞれ読み出し、並列に表示装置に出力する機能と、

をコンピュータに実現させ、

前記処理する機能において、

一の動画ファイルに対応する前記復号化スレッドの処理開始時に、当該動画ファイルに対応づけられたテクスチャバッファに格納され、読み出されていない前記テクスチャデータの量を検出する機能と、

前記テクスチャデータの量が当該テクスチャバッファの容量より小さい規定値より大きいとき、前記復号化スレッドの復号化に係る処理を休止し、当該スレッドを前記処理待ち行列に格納したうえ、別の動画ファイルに対応する前記復号化スレッドを含む前記処理待ち行列から、次に処理すべきスレッドを取得して処理を開始する機能と、

をさらにコンピュータに実現させることを特徴とするコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、情報処理技術に関し、特に複数の入力データのそれぞれに処理を施し並列に出力する情報処理方法、動画サムネイル表示方法、およびそれを適用する復号化装置および情報処理装置に関する。

【背景技術】

【0002】

近年の情報処理技術の著しい進歩に伴い、高速演算を行う情報処理装置がより身近なものになってきた。同時に画像や音声など各種データの処理技術等が発展したことにより、ユーザは様々なデータを容易に取得し、再生したり所望の処理を施したりすることができるようになった。

【0003】

そのような状況において、同種の複数のデータを並列に処理し、並列に出力する技術は、膨大なデータの内容把握や、処理結果の比較といった目的で一般的に利用されるようになった。例えば動画再生装置における圧縮符号化された動画ファイルのサムネイル表示や、多重チャンネルでリアルタイムに取得された複数の画像、音声、温度などのデータの画像解析による比較映像など、そのような技術の利用分野は多岐にわたる。

【発明の開示】

【発明が解決しようとする課題】

【0004】

このようなデータの処理装置において並列処理を実現するためには、処理したいデータの数に応じてスレッドを生成し、CPUやメモリなどの限られたリソースを、時間的、空間的に分割して各スレッドに割り当てる必要がある。したがって、一度に表示するサムネイルの数を増やしたり、入力チャンネルの数を増やしてより多くの結果を比較したりしようとすると、スレッドの数が増え、装置の負荷が増大する。このため、例えばサムネイルとして複数の動画をそのまま再生する動画サムネイル表示時や、複数チャンネルのデータの経時変化を比較する場合など、リアルタイム性を要求される処理を並列して行う場合に所定の出力レートを確保するためには、並列に出力できるデータの数を増やすことが困難であった。

【0005】

本発明はこのような課題に鑑みてなされたものであり、その目的は多数のデータを並列に出力する場合であっても、出力を円滑に行うことのできる技術を提供することにある。

【課題を解決するための手段】

【0006】

10

20

30

40

50

本発明のある態様は情報処理方法に関する。この情報処理方法は、入力データストリームに所定の処理を施した中間データを対応するバッファに格納するスレッドを入力データストリームごとに生成し、切り替えて処理するステップと、複数の中間データを対応するバッファからそれぞれ読み出し、複数の出力データとして並列に出力するステップと、を含む。そして、切り替えて処理するステップは、スレッドの処理開始時に、バッファに格納され、読み出されていない中間データの量を検出するステップと、中間データの量が規定値より大きいとき、当該スレッドの処理を休止し、他のスレッドへ処理を切り替えるステップと、を含むことを特徴とする。

【0007】

ここで「入力データストリーム」は、ある期間に連続的に出力されるべきデータ、または、処理して出力されるべき時間がある許容範囲をもって定められている一続きの時系列データであってよく、圧縮符号化された動画ファイル、音声ファイルなどのデータおよびその組み合わせ、圧縮されていない各種データなどのいずれでもよい。従って「所定の処理」は圧縮符号化されたデータであれば復号化、その他のデータであれば画像解析、音声解析、2次元プロファイル形成など、入力データストリームの種類と所望の出力データによって決定され、同種、異種によらず複数単位の処理を含んでよい。出力データは中間データに必要な処理を施してもよいし、中間データをそのまま使用してもよい。

【0008】

「スレッド」は一般的にはアプリケーションに対応したプロセスを細分化したある機能を実現する処理単位であるが、ここではそれに限らず、プロセスそのものでも、ある機能を実現するために呼び出されるコンポーネントに対応した処理単位でもよく、何らかの処理または機能の単位に対応すればよい。

【0009】

本発明の別の態様は動画サムネイル表示方法に関する。この動画サムネイル表示方法は、圧縮符号化された動画ファイルを復号化して生成されたテキストデータを対応するテキストバッファに格納する処理を行う復号化スレッドと、テキストデータを対応するテキストバッファから読み出し、画像フレームデータを生成して対応するフレームバッファに格納する処理を行う描画スレッドとを動画ファイルごとに生成し、切り替えて処理するステップと、複数の動画ファイルの画像フレームデータを、対応するフレームバッファからそれぞれ読み出し、並列に表示装置に出力するステップと、を含み、切り替えて処理するステップは、復号化スレッドの処理開始時に、テキストバッファに格納され、読み出されていないテキストデータの量を検出するステップと、テキストデータの量が規定値より大きいとき、復号化スレッドの復号化に係る処理を休止し、他のスレッドへ処理を切り替えるステップと、を含むことを特徴とする。

【0010】

本発明のさらに別の態様は復号化装置に関する。この復号化装置は、符号化されたデータストリームを入力する入力部と、入力部に入力した複数の前記データストリームを切り替えて順に復号化する復号化部と、復号化部が復号化した中間データを一時保存する、前記データストリームのそれぞれに対応して設けられた複数のバッファと、複数のバッファに保存された複数の前記中間データを読み出し、並列に出力する出力部と、を備え、復号化部は、データストリームの復号化を行う前に、対応するバッファに格納され、読み出されていない中間データの量を検出し、中間データの量が規定値より大きいときは、データストリームの復号化を休止し、他の処理を開始することを特徴とする。

【0011】

本発明のさらに別の態様は情報処理装置に関する。この情報処理装置は、圧縮符号化された複数の動画ファイルのそれぞれに対応した複数のバッファと、動画ファイルを復号化し、生成したテキストデータを対応するバッファに格納する復号化スレッドと、テキストデータを前記バッファから読み出し、動画ファイルに対応した画像フレームデータを生成する描画スレッドと、を動画ファイルごとに生成し、切り替えて処理するプロセッサユニットと、プロセッサユニットにより生成された、複数の動画ファイルの画像フレーム

10

20

30

40

50

データを並列に表示する画像表示部と、を備え、プロセッサユニットは、復号化スレッドの処理開始時に、バッファに格納され、読み出されていないテキストデータの量に応じて、復号化スレッドの復号化に係る処理の続行可否を判定することを特徴とする。

【 0 0 1 2 】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【 0 0 1 3 】

本発明によれば、再生装置や情報処理装置等において、複数のデータの並列出力を効率的に行うことができる。

10

【発明を実施するための最良の形態】

【 0 0 1 4 】

図1は本実施の形態における情報処理装置の構成を示している。情報処理装置100は装置全体を統括的に制御するとともに、圧縮符号化された動画ファイルの復号化処理などを行うプロセッサユニット20、実行中のプログラムや実行に必要なデータを記憶するメモリ70、ユーザによる入力指示を受け付ける入力装置75、動画ファイルを保存する記憶装置80、および復号化した動画ファイルのフレームを表示する表示装置90を含む。プロセッサユニット20、メモリ70、入力装置75、記憶装置80、表示装置90はバス95によって相互にデータ伝送を行う。

【 0 0 1 5 】

入力装置75はキーボード、マウス、トラックボールなど一般的に用いられる入力装置のいずれかでよい。記憶装置80は符号化された動画ファイルを記録したハードディスクや、DVD (Digital Versatile Disk)、CD (Compact Disk)などの記録媒体の読取装置などのいずれかでよい。

20

【 0 0 1 6 】

プロセッサユニット20は、情報処理装置100における処理を制御する管理部30、管理部30の管理下で処理を実行する処理部40を含む。管理部30はメモリ70、入力装置75、記憶装置80、表示装置90とプロセッサユニット20とのデータの入出力を制御するほか、処理部40における処理のスケジューリングを行い、複数のプロセスやスレッドを切り替えて処理するように制御する。

30

【 0 0 1 7 】

記憶装置80に保存された圧縮符号化された複数の動画ファイルは、ユーザによる入力装置75に対する動画サムネイル表示の指示により管理部30の制御のもと、メモリ70に格納される。処理部40はメモリ70に格納された複数の動画ファイルを並列して復号化し、最終的に出力すべき画像フレームデータを動画ファイルごとに生成する。そして管理部30が、生成された画像フレームデータを表示装置90の画面内で動画ファイルごとに設けた領域に順次出力するよう制御を行うことにより、表示装置90に動画サムネイルが複数表示される。

【 0 0 1 8 】

なお圧縮符号化はMPEG4、H.264 / AVC、SVC (Scalable Video Coding)など一般的な圧縮符号化手法のいずれに基づいていてもよい。

40

【 0 0 1 9 】

本実施の形態では1つの動画ファイルに対する動画サムネイル表示のための復号化処理を、1つのスレッドによって行う。従って複数の動画ファイルの動画サムネイルを表示させるためには、処理部40において複数の動画ファイルを復号化する複数のスレッドを並列に処理する。さらに処理部40は、復号化したデータに基づき画像フレームデータを生成するスレッドや、動画サムネイル表示以外のプロセスまたはスレッドも並列に処理する。処理部40は複数のサブプロセッサで構成されていてもよい。この場合も管理部30は、時分割したスレッドを各サブプロセッサに割り当てることにより、複数のスレッドを並列処理するように制御を行う。

50

【 0 0 2 0 】

この情報処理装置 1 0 0 では、情報処理装置 1 0 0 を効率よく使用するための機能、環境を提供し、装置全体を統括的に制御するオペレーティングシステム（以下、OS と略す）が実行される。その OS 上で複数のアプリケーションソフトウェア（以下、単にアプリケーションという）が実行される。例えば管理部 3 0 におけるスレッドのスケジューリングには、一般的なマルチスレッド対応の装置において導入されている OS をそのまま利用することができる。

【 0 0 2 1 】

図 2 は処理部 4 0 およびメモリ 7 0 のうち、動画サムネイル表示を行うための構成を示している。図 2 や図 1 のプロセッサユニット 2 0 などにおいて、様々な処理を行う機能ブロックとして記載される各要素は、ハードウェア的には、CPU、メモリ、その他の LSI で構成することができ、ソフトウェア的には、画像処理機能のあるプログラムなどによって実現される。したがって、これらの機能ブロックがハードウェアのみ、ソフトウェアのみ、またはそれらの組合せによっていろいろな形で実現できることは当業者には理解されるところであり、いずれかに限定されるものではない。

10

【 0 0 2 2 】

例えば図 2 に示した処理部 4 0 のブロックのうち破線で囲まれた要素は、ある動画ファイルの復号化を行うためのスレッドである、復号化スレッド 4 3 を処理するための機能ブロックを示している。上述のとおり本実施の形態では動画ファイルごとに復号化処理を行うスレッドが生成されるため、実際には復号化スレッド 4 3 と同様の構成を有する復号化スレッドが複数存在し、OS が割り当てたタイムスライスごとにそれぞれのスレッドに含まれる機能が発揮されることになる。描画部 5 0 も動画ファイルごとに描画スレッド 5 1 が生成され、タイムスライスごとに異なる動画ファイルの画像フレームデータが生成される。

20

【 0 0 2 3 】

処理部 4 0 では上述のとおり、図 2 で示した動画サムネイル表示を行うためのスレッド以外のスレッドを処理する場合もある。この際は、処理部 4 0 ではあるタイムスライスにおいてまた別の機能が発揮されることになるが、本実施の形態ではその機能を限定しないため、機能ブロックは同図では省略する。

【 0 0 2 4 】

メモリ 7 0 は、圧縮符号化された動画ファイルを格納するファイル格納部 4 7 と、各フレームのテクスチャデータを保存するテクスチャバッファ 4 8 と、表示装置 9 0 に表示させる画像フレームデータを保存するフレームバッファ 5 2 を含む。テクスチャバッファ 4 8 およびフレームバッファ 5 2 は図示しないグラフィックメモリなどを別に設け、その中に領域を確保してもよい。テクスチャバッファ 4 8 とフレームバッファ 5 2 は動画ファイルごとに領域が割り当てられ、復号化スレッド 4 3 や描画スレッド 5 1 は、それぞれに対応した動画ファイルに割り当てられたテクスチャバッファ 4 8 およびフレームバッファ 5 2 の領域を使用する。以後の説明では、テクスチャバッファ 4 8 およびフレームバッファ 5 2 内の個別の領域も、テクスチャバッファ 4 8、フレームバッファ 5 2 と呼ぶ。

30

【 0 0 2 5 】

処理部 4 0 は動画ファイルの復号化スレッド 4 3 として、メモリ 7 0 のファイル格納部 4 7 から圧縮符号化された動画ファイルを取得する初期処理部 4 2、符号化された動画ファイルを復号化する復号化部 4 4、復号化したデータをテクスチャデータとしてテクスチャバッファ 4 8 に保存するバッファ書込部 4 6 を含む。処理部 4 0 はさらに、テクスチャデータをテクスチャバッファ 4 8 から読み出し画像フレームデータを生成してフレームバッファ 5 2 に書き込む描画部 5 0 を、描画スレッド 5 1 として含む。

40

【 0 0 2 6 】

初期処理部 4 2 は後述するように、テクスチャバッファ 4 8 に格納されたテクスチャデータのフレーム数に基づき、以後の処理を続行するかどうかの決定も行う。バッファ書込部 4 6 は復号化部 4 4 において復号化したデータに対し RGB 変換など必要な処理を施し

50

でテクスチャデータとしてもよい。描画部50は動画ファイルの種類によって、ポリゴンにテクスチャを貼り付けるなどの必要な処理を適宜行い画像フレームデータを生成する。画像フレームデータの生成に必要なデータのうち、テクスチャデータ以外のデータは、描画部50がメモリ70より適宜読み込んでよい。

【0027】

次に上記の構成によって本実施の形態を実現するための動作について説明する。本実施の形態では複数の動画サムネイル表示処理を効率化するために、処理部40における復号化スレッド43の処理時間を、スレッド自身によって変化させる。本実施の形態の効果を明らかにするため、まず一般的なOSによって割り当てられたタイムスライスをそのまま適用した場合について説明する。

10

【0028】

図3は処理部40におけるスレッド切り替えの例を模式的に示している。同図横方向が時間軸であり、第1復号化スレッド43a、第1描画スレッド51a、第2復号化スレッド43b、第2描画スレッド51b、動画サムネイル表示以外のスレッドである他スレッド55が、各タイムスライスごとに順番に処理されている。実際には第1復号化スレッド43aの後に他スレッド55が処理されるなど、優先度に基づくOSのスケジューリングによって順番は様々となる。

【0029】

割り当てられた処理時間が終了したら、各スレッドはCPUの割り当てを開放し、次の処理時間が到来するまで実行待ちの行列に並んで待機状態となる。復号化スレッド43および描画スレッド51は第1および第2の2つに限らず、サムネイル表示を行いたい動画ファイルの数だけ発生する。全てのスレッドは図示した時間以降も、繰り返し処理が行われる。

20

【0030】

第1復号化スレッド43aのバッファ書込部46は、テクスチャバッファ48内の割り当てられた領域である第1テクスチャバッファ48aに動画のテクスチャデータをフレームごとに格納する。第1復号化スレッド43aが待ち行列に並んでいる間に、第1描画スレッド51aの処理が開始されると、描画部50は第1テクスチャバッファ48a内のテクスチャデータをフレームごとに消費して、当該動画の画像フレームデータを生成し、フレームバッファ52内の割り当てられた領域、第1フレームバッファ52aへ格納する。これらの処理を繰り返すことにより、1つの動画ファイルの動画サムネイルが表示装置90に表示される。

30

【0031】

同様に、第2復号化スレッド43bが第2テクスチャバッファ48bへテクスチャデータを格納し、第2描画スレッド51bがそれを消費して第2フレームバッファ52bに画像フレームデータを格納することによって、別の動画ファイルの動画サムネイルが表示装置90に表示される。

【0032】

各描画スレッド51が一度の処理時間で消費しきれなかったテクスチャバッファ48内のテクスチャデータはそのままテクスチャバッファ48に残り、各復号化スレッド43のバッファ書込部46は、新たに生成したテクスチャデータをテクスチャバッファ48に追加して格納していく。

40

【0033】

バッファ書込部46はテクスチャデータをテクスチャバッファ48に格納するとともに、テクスチャバッファ48内に格納されているテクスチャデータのフレーム数をカウントするカウンタ(不図示)を動作させる。あるタイムスライスにおいて処理部40が復号化スレッド43を処理中、バッファ書込部46がテクスチャバッファ48の容量が満たされた、または満たされつつあることを当該カウンタにより検知すると、復号化スレッド43はタイムスライス中でもCPUの割り当てを開放して実行待ちの行列に並ぶ。後に述べるように、本実施の形態ではカウンタは初期処理部42によっても参照される。

50

【 0 0 3 4 】

ところで復号化部 4 4 における動画ファイルの復号化の速度は、元の画像に依存して変化する。例えば基準フレームから得られる予測画像と実際の画像との差分画像を利用した予測符号化を行った動画ファイルにおいては、ほとんど動きのないシーンと、動きの早いシーンや色の変化の大きいシーンとで演算量が異なり、前者の方が復号化の速度が速い。復号化の速度は 1 つの動画ファイルにおいてもシーンによって変化する。さらに動画サムネイル表示のように一斉に複数の動画ファイルを再生した場合には、同一時刻において動画ファイル間で速度の大小が発生する。

【 0 0 3 5 】

図 4 はある時点で復号化の速度が異なる 2 つの動画ファイルの復号化スレッド、第 1 復号化スレッド 4 3 a および第 2 復号化スレッド 4 3 b がそれぞれ使用する、2 つのテキストバッファ、第 1 テキストバッファ 4 8 a および第 2 テキストバッファ 4 8 b におけるテキストデータの格納状況を模式的に示している。ここで第 1 復号化スレッド 4 3 a は復号化速度の速い動画ファイルの復号化スレッド、第 2 復号化スレッド 4 3 b は復号化速度の遅い動画ファイルの復号化スレッドとする。

【 0 0 3 6 】

各スレッドに割り当てられた第 1 テキストバッファ 4 8 a および第 2 テキストバッファ 4 8 b は、同図の例では 8 フレーム分のテキストデータを格納することができ、1 フレーム分を 1 つの矩形で示している。さらに 1 フレーム分を表す矩形内に示された「」は、テキストバッファ 4 8 に格納され、消費される前のテキストデータのフレーム数を表している。同図の例では第 1 テキストバッファ 4 8 a には 7 フレーム分、第 2 テキストバッファ 4 8 b には 1 フレーム分のテキストデータが格納されている。

【 0 0 3 7 】

一般的に OS は、各スレッドに均等に処理時間を割り当てる。そのため、1 回の処理時間において復号化速度の速い第 1 復号化スレッド 4 3 a が処理できるフレーム数は、第 2 復号化スレッド 4 3 b より多くなる。さらに第 1 復号化スレッド 4 3 a は、割り当てられた第 1 テキストバッファ 4 8 a の容量を満たしてタイムスライス満了を待たずに CPU の割り当てを開放する確率が高くなるため、複数のサブプロセッサで複数のスレッドを並列に処理する環境下では、待ち行列に並ぶ回数が第 2 復号化スレッド 4 3 b より多くなる。この場合も OS は、「動画ファイルの復号化」という共通のスレッドである第 1 復号化スレッド 4 3 a および第 2 復号化スレッド 4 3 b の双方に同一の優先度を与えるため、結果として第 1 復号化スレッド 4 3 a が処理される回数は第 2 復号化スレッド 4 3 b に比べて多くなる。

【 0 0 3 8 】

以上のことから、図 3 に示すように、第 1 復号化スレッド 4 3 a に割り当てられた第 1 テキストバッファ 4 8 a は、第 2 復号化スレッド 4 3 b に割り当てられた第 2 テキストバッファ 4 8 b より、格納されているテキストデータの量が恒常的に多くなる。さらにこのような状況下では、第 2 復号化スレッド 4 3 b に対応する動画ファイルのテキストデータの蓄積が少なく、対応する第 2 描画スレッド 5 1 b の処理が開始されても十分な数の画像フレームデータが生成できない可能性がある。結果として画像フレームデータが表示装置 9 0 における出力タイミングに間に合わず、動画サムネイルがコマ落ちする確率が高くなる。

【 0 0 3 9 】

動画サムネイルのコマ落ちを防止するために、復号化速度の遅い動画ファイルの復号化スレッドの優先度を上げるなど、スケジューリング時の優先度を動的に割り振るように OS 上で操作を行うと、動画サムネイル表示と関係しない他のプロセスの優先度が相対的に下がる場合が発生しうる。他の重要なプロセスの処理の優先度が不安定になる状況は望ましくないため、本実施の形態では一般的な OS が提供する優先度やスケジューリングをそのまま導入しながら、動画サムネイルのコマ落ち削減を実現する。

【 0 0 4 0 】

10

20

30

40

50

上述のとおり復号化速度の速い復号化スレッドと復号化速度の遅い復号化スレッドとは、テクスチャバッファ48におけるテクスチャデータの蓄積量に差が生じる。本実施の形態ではこの差を利用し、「復号化スレッド」のグループ間で復号化速度の遅い復号化スレッドを優先して処理するようにする。さらに復号化速度はシーンによりリアルタイムで変化するため、優先状況はそれに応じて変化させる。

【0041】

具体的にはテクスチャバッファ48におけるテクスチャデータの格納量に第1のしきい値F1および第2のしきい値F2 ($F1 < F2$) を設ける。ここで第2のしきい値F2はテクスチャバッファ48中の割り当てられた容量、またはそれに近い容量である。そしてテクスチャデータの格納量が一旦第2のしきい値F2を超えたら、テクスチャデータの格納量が第1のしきい値F1以下になるまで復号化を行わずにCPUの割り当てを開放して次のスレッドに処理を譲るイールド(Yield)処理を行う(以下、イールドと呼ぶ)。そして描画部50によってテクスチャデータが消費され、格納されているテクスチャデータが第1のしきい値F1以下になったら復号化処理を再開し、格納量が再び第2のしきい値F2を超えるまで処理を続ける。

10

【0042】

図5は本実施の形態の復号化スレッド43の割り当て処理時間における処理手順を示している。まずOSが当該復号化スレッド43に対して割り当てた時間が開始すると(S10)、初期処理部42は図示しないレジスタ等に記憶したフラグ値を確認する(S12)。フラグ値は、ユーザが動画サムネイル表示機能を起動させた直後など、当該復号化スレッド43の初期状態では「0」である。フラグが0であったら(S12のY)、初期処理部42はメモリ70から復号化対象となる動画ファイルを取得する(S20)。そして復号化部44およびバッファ書き込み部46が復号化およびテクスチャバッファ48への書き込み処理を通常通り行う(S22)。

20

【0043】

テクスチャデータを書き込んだ際、バッファ書き込み部46はカウンタによって、テクスチャバッファ48に格納されているテクスチャデータのフレーム数が第2のしきい値F2に到達したかを確認する(S24)。格納量が第2のしきい値F2より小さければ(S24のY)、当該スレッドの割り当て時間が満了するまで、復号化処理を続行する(S28のN、S20)。テクスチャデータの格納量が第2のしきい値F2となったら(S24のN)、フラグ値を「1」に設定し、OSの制御のもとレジスタの値をメモリ70に退避させるなどの処理を行いスレッド処理を終了させる(S30)。S30のステップは、テクスチャデータの格納量が第2のしきい値F2へ到達する前に割り当て時間が満了した場合(S28のY)も同様に行う。

30

【0044】

復号化スレッド43の処理開始時にフラグ値が「1」であった場合(S12のN)、初期処理部42はバッファ書き込み部46が管理するカウンタにより、テクスチャバッファ48に格納されているテクスチャデータのフレーム数を検出する。そして格納されているフレーム数が第1のしきい値F1より大きいかを確認する(S14)。

【0045】

第1のしきい値F1としては、例えば図4で示したテクスチャバッファ48において、8フレーム分の記憶容量に対して2フレーム分など、復号する動画ファイルの符号化方法や、必要なフレームレートなどに応じて適宜計算あるいは実験によって求める。第1のしきい値F1を2フレーム分とすると、図4の例では、第1テクスチャバッファ48aは規定値より多いフレーム数のテクスチャデータが、第2テクスチャバッファ48bは規定値以下のフレーム数のテクスチャデータが格納されている。

40

【0046】

テクスチャデータの格納量が第1のしきい値F1より大きかったら(S14のY)、画像フレームデータの生成に十分なテクスチャデータの蓄積があるものとしてイールドする。この際は、通常のOSによるスレッド切り替え時と同様に、レジスタの値をメモリ70

50

に退避させるなどの処理を行う（S18）。この処理は実際にはOSに用意された機能呼び出すことによりOSが行ってもよい。

【0047】

テクスチャデータの格納量が第1のしきい値F1以下であったら（S14のN）、初期処理部42はフラグ値を「0」とする（S16）。そして上述のS20からS30のステップを行うことにより、テクスチャデータを貯めていく。

【0048】

以上の手順により、復号化速度の速い動画ファイルやあるシーンでは、復号化スレッド43の処理順が回ってきても、スレッドが自らイールドすることによって処理時間の消費が抑えられる。その分、他のスレッド、ひいては復号化速度の遅い他の復号化スレッドの単位時間あたりの処理回数が増加し、結果的には他の復号化スレッドの優先度を上げていることになる。このときOSが決定した優先度はそのままであるため、動画ファイルの復号化スレッド以外のスレッドとの相対的な優先度には影響せず、他のプロセスの動作が遅くなるなどの不安定要素は発生しない。

【0049】

そして1回の処理時間内で生成できるテクスチャデータのフレーム数が小さい、復号化に時間のかかる動画ファイルやシーンを復号化するスレッドは、単位時間あたりの処理回数が増加することによってテクスチャバッファ48におけるテクスチャデータの格納量を増加できる。復号化速度の速い復号化スレッドは、イールドすることによってテクスチャデータが消費されるのみになり、テクスチャバッファ48における格納量が一旦減少するが、第1のしきい値F1で設定した不足しない範囲の下限に達すると再び格納を開始する。

【0050】

そして格納量が第2のしきい値F2に到達するまでは、復号化速度の速い復号化スレッドも復号化処理を続けて行う。このため一旦イールドに転じた復号化スレッドは、テクスチャデータの貯蓄が十分であることから、割り当て時間が到来しても連続してイールドを行うことになり、まとまった時間を他のスレッドに譲ることができる。そして復号化速度の遅い復号化スレッドは譲られた時間を利用して効率よく復号化処理を行うことができる。

【0051】

結果として、描画部50が画像フレームデータの生成を行うのに十分な量のテクスチャデータがいずれの復号化スレッド43に割り当てられたテクスチャバッファ48にも格納されることになり、動画サムネイルのコマ落ちを削減することができる。

【0052】

実際に25個の動画サムネイルを表示した場合に本実施の形態を適用した。本実施の形態を適用せず、OSによる処理時間制御のみを行った場合、25個の動画において1秒間に合計20～30フレームのコマ落ちが確認されたのに対し、本実施の形態を適用することによって、1秒間に合計数フレームレベルのコマ落ちに抑えることができた。このようなレベルであれば、ユーザがコマ落ちを視認する可能性は極めて少ない。

【0053】

以上述べた本実施の形態によれば、動画サムネイルを複数表示させる際、復号化速度の速い動画の復号化スレッドと、復号化速度の遅い動画の復号化スレッドとを、テクスチャバッファにおけるテクスチャデータの格納量をリアルタイムで検出して判断する。そして復号化速度の速い復号化スレッドは、OSによる処理開始時間が開始しても復号化処理を行わずにイールドする。イールドはテクスチャデータが不足しない範囲で行われる。これにより、処理時間を有効に活用して、復号化速度の遅い動画の復号化スレッドの単位時間あたりの処理回数を増加させることができ、テクスチャデータの蓄積不足によるコマ落ちを削減することができる。

【0054】

本実施の形態はソフトウェアの改良により、実質上、動画サムネイル表示の処理速度を

10

20

30

40

50

増加させることができ、CPUやバスなどハードウェアの改良による処理速度向上と比較して安価なコストで導入することができる。またOSも一般的な動作を行うのみでよいため、携帯端末から大規模なシステムまで様々な装置に汎用的に適用することができる。

【0055】

以上、本発明を実施の形態をもとに説明した。上記実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【0056】

例えば本実施の形態では復号化スレッドがイールドを行うバッファ格納量のしきい値をあらかじめ設定したが、動画サムネイル表示をさせながら、動画ファイルの符号化方法や表示装置などに対して適応的にしきい値を自動的に決定するようにしてもよい。

10

【0057】

また本実施の形態では複数の動画サムネイル表示を行うために生成した、複数の復号化スレッド間で処理時間の譲渡を行ったが、割り当てられたバッファにそれぞれが出力を行う複数のスレッドと、バッファに格納されたデータをそれぞれ消費するスレッドと、を生成させるプロセスであれば同様の処理を適用することができる。例えば、カメラやマイク、温度センサなど外部デバイスから入力した複数のリアルタイムデータをそれぞれ画像化、音声処理などして出力するなどのマルチ入出力装置、複数台のネットワークコンピュータから取得した複数のデータから複数の出力を行う処理などに適用できる。

【0058】

20

このような処理を行う装置において、さらに別のプロセスが実行されている場合など、OSが決定した各スレッドの優先度を変化させることが困難な状況においては、本実施の形態は特に効果的である。

【0059】

また本実施の形態はOSが同一の処理時間を各スレッドに割り当てるプリエンティブ方式のマルチスレッドにおける処理手順を示しているが、各スレッドが所定の処理タイミングでCPUの割り当てを開放することによってスレッド処理の切り替えを行うノンプリエンティブ方式のマルチスレッドにおいても適用できる。この場合も復号化速度の遅速によって実行待ちの行列に並ぶ回数に差が生じるため、復号化速度の速い復号化スレッドがイールドすることによって、復号化速度の遅い復号化スレッドの処理回数を増加させることができ、本実施の形態と同様の効果を得ることができる。このときは、図5におけるS28の割り当て時間満了確認処理に代わり、切り替えタイミングの処理が行われたかの確認処理を行えばよい。あるいはS24Nの、テクスチャデータの格納量がF2となったタイミング自体を切り替えタイミングとしてもよい。

30

【図面の簡単な説明】

【0060】

【図1】本実施の形態における情報処理装置の構成を示す図である。

【図2】本実施の形態において動画サムネイルを表示する際の処理部の構成を示す図である。

【図3】本実施の形態の処理部におけるスレッド切り替えの様子を模式的に示す図である。

40

【図4】本実施の形態において2つのテクスチャバッファにおけるテクスチャデータの格納状況を説明する図である。

【図5】本実施の形態における復号化スレッドの割り当て処理時間における処理手順を示すフローチャートである。

【符号の説明】

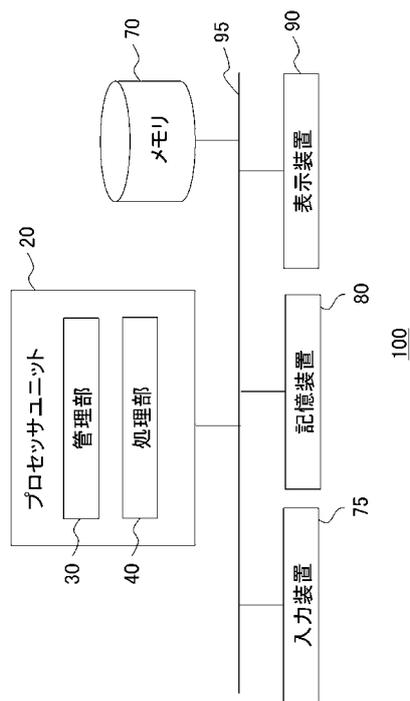
【0061】

20 プロセッサユニット、 30 管理部、 40 処理部、 42 初期処理部、
43 復号化スレッド、 44 復号化部、 46 バッファ書込部、 47 ファイル格納部、 48 テクスチャバッファ、 50 描画部、 51 描画スレッド、 5

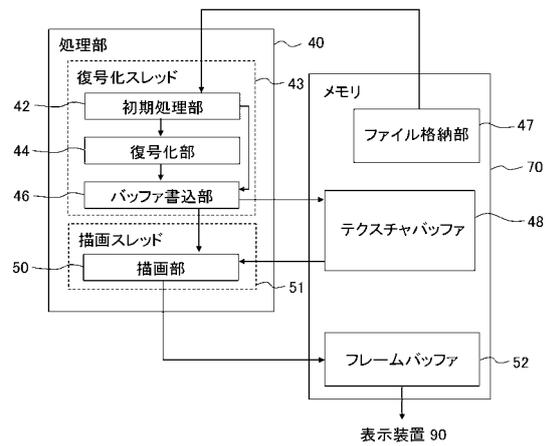
50

2 フレームバッファ、 70 メモリ、 75 入力装置、 80 記憶装置、 90 表示装置、 100 情報処理装置。

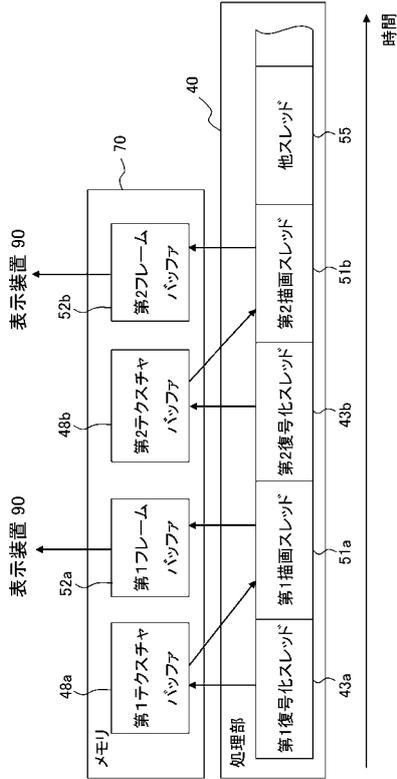
【図1】



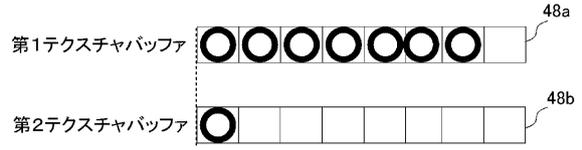
【図2】



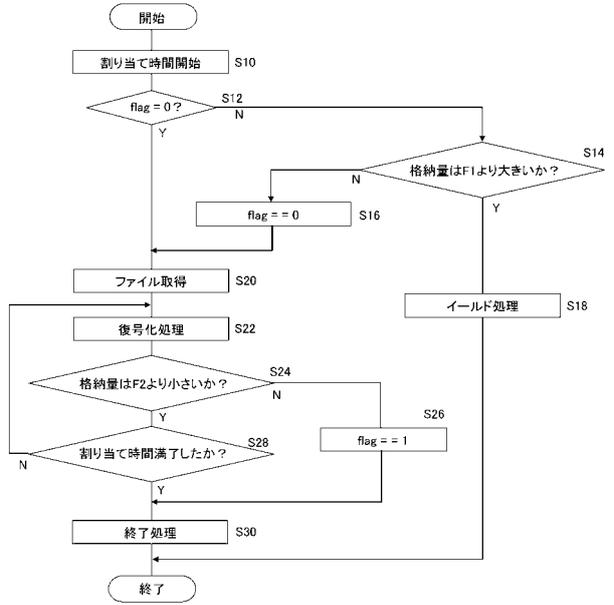
【図3】



【図4】



【図5】



フロントページの続き

- (56)参考文献 特開2006-012150(JP,A)
国際公開第2005/078580(WO,A1)
国際公開第2004/002145(WO,A1)
特開平08-107542(JP,A)
特開平09-282184(JP,A)
特開2000-353099(JP,A)
特開2002-032233(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04N	5/76	-	5/781
H04N	5/80	-	5/956
H04N	7/12		
H04N	7/26	-	7/32
G06F	9/46	-	9/54