

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4795138号
(P4795138)

(45) 発行日 平成23年10月19日(2011.10.19)

(24) 登録日 平成23年8月5日(2011.8.5)

(51) Int. Cl. F 1
G 0 6 T 1 / 2 0 (2 0 0 6 . 0 1) G O 6 T 1 / 2 0 C

請求項の数 10 (全 42 頁)

| | | | |
|-----------|------------------------------|-----------|-------------------|
| (21) 出願番号 | 特願2006-179258 (P2006-179258) | (73) 特許権者 | 000005496 |
| (22) 出願日 | 平成18年6月29日 (2006. 6. 29) | | 富士ゼロックス株式会社 |
| (65) 公開番号 | 特開2008-9697 (P2008-9697A) | | 東京都港区赤坂九丁目7番3号 |
| (43) 公開日 | 平成20年1月17日 (2008. 1. 17) | (73) 特許権者 | 306037311 |
| 審査請求日 | 平成20年11月11日 (2008. 11. 11) | | 富士フイルム株式会社 |
| | | | 東京都港区西麻布2丁目26番30号 |
| | | (74) 代理人 | 100079049 |
| | | | 弁理士 中島 淳 |
| | | (74) 代理人 | 100084995 |
| | | | 弁理士 加藤 和詳 |
| | | (74) 代理人 | 100085279 |
| | | | 弁理士 西元 勝一 |
| | | (74) 代理人 | 100099025 |
| | | | 弁理士 福田 浩志 |

最終頁に続く

(54) 【発明の名称】 画像処理装置及びプログラム

(57) 【特許請求の範囲】

【請求項1】

自モジュールの前段から画像データを取得し、取得した画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する画像処理モジュールとして各々機能すると共に、前記画像処理の種類又は内容が互いに異なる複数種の画像処理モジュールのプログラムを各々記憶する記憶手段と、

前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第1の画像処理部を構築し、前記第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する並列処理制御手段と、

前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御

10

20

を行わない第2のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第2の画像処理部を構築し、前記第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御する逐次処理制御手段と、

前記並列処理制御手段及び逐次処理制御手段を選択的に動作させ、処理対象の画像に対する画像処理を前記第1の画像処理部又は前記第2の画像処理部によって行わせる選択手段と、

を含み、

前記並列処理制御手段及び前記逐次処理制御手段は、前記第1の画像処理部又は第2の画像処理部の実行単位プログラムから状態通知が入力される毎に所定の処理を行って前記第1の画像処理部又は前記第2の画像処理部の状態を管理する状態管理手段を各々備え、前記並列処理制御手段に設けられている状態管理手段は、前記第1の画像処理部の実行単位プログラムから入力された状態通知を排他的に処理すると共に、エラーの発生を通知する状態通知が入力された場合に、エラー発生通知元以外の実行単位プログラムの実行を中止させる画像処理装置。

10

【請求項2】

前記記憶手段は、前記第1のバッファモジュールのプログラム及び前記第2のバッファモジュールのプログラムも各々記憶しており、前記並列処理制御手段は、前記記憶手段に記憶されている前記第1のバッファモジュールのプログラムを用いて前記第1の画像処理部を構築し、前記逐次処理制御手段は、前記記憶手段に記憶されている前記第2のバッファモジュールのプログラムを用いて前記第2の画像処理部を構築することを特徴とする請求項1記載の画像処理装置。

20

【請求項3】

前記画像処理モジュールは、画像処理の実行指示が入力される毎に、自モジュールの前段から画像データを所定量取得し、取得した所定量の画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する構成であり、

前記並列処理制御手段は、前記第1の画像処理部を構成する画像処理モジュールのうちの一部又は全ての画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、前記第1の画像処理部を構成する個々の画像処理モジュールで互いに並列に画像処理が行われるように、前記第1の画像処理部を構成する個々の画像処理モジュールへ画像処理の実行指示を入力し、

30

前記逐次処理制御手段は、前記第2の画像処理部を構成する画像処理モジュールのプログラムを、複数の画像処理モジュールのプログラムを実行単位プログラムとしてプログラム実行リソースにより実行させると共に、前記個々の画像処理モジュールで画像処理が逐次行われるように、前記第2の画像処理部を構成する個々の画像処理モジュールへ画像処理の実行指示を入力することを特徴とする請求項1記載の画像処理装置。

【請求項4】

前記並列処理制御手段として、

前記第1の画像処理部を構成する個々の画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、個々の実行単位プログラムの実行優先度を固定的に設定する第1の並列処理制御手段、

40

及び、前記第1の画像処理部を構成する個々の画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、個々の実行単位プログラムの実行優先度を、前記第1の画像処理部における画像処理の進行度合に応じて変更する第2の並列処理制御手段

の少なくとも一方を備えていることを特徴とする請求項1記載の画像処理装置。

【請求項5】

前記画像処理モジュールは、画像処理の実行指示が入力される毎に、自モジュールの前段から画像データを所定量取得し、取得した所定量の画像データに所定の画像処理を行い

50

、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する構成であり、

前記逐次処理制御手段として、

前記処理対象の画像データのうちの所定量の画像データに対する画像処理を、前記第2の画像処理部を構成する画像処理モジュールのうち前記パイプライン形態又は有向非循環グラフ形態における前段側の画像処理モジュールから順に行うことが、前記処理対象の画像データに対する画像処理が終了する迄繰り返されるように、画像処理の実行指示を入力する第1の逐次処理制御手段、

及び、前記第2の画像処理部を構成する画像処理モジュールのうち特定の画像処理モジュールで前記処理対象の画像データに対する画像処理が終了する迄、前記特定の画像処理モジュールへ画像処理の実行指示を繰り返し入力することを、パイプライン形態又は有向非循環グラフ形態における前段側の画像処理モジュールから順に行う第2の逐次処理制御手段

の少なくとも一方を備えていることを特徴とする請求項1記載の画像処理装置。

【請求項6】

前記選択手段は、利用者によって予め入力された選択結果又は選択条件に基づいて、前記並列処理制御手段を動作させるか逐次処理制御手段を動作させるかを選択することを特徴とする請求項1記載の画像処理装置。

【請求項7】

前記選択手段は、前記並列処理制御手段を動作させるか前記逐次処理制御手段を動作させるかを、前記第1の画像処理部又は前記第2の画像処理部が動作する環境に応じて選択することを特徴とする請求項1記載の画像処理装置。

【請求項8】

前記選択手段は、前記第1の画像処理部又は前記第2の画像処理部が動作する環境として、前記第1の画像処理部又は前記第2の画像処理部を構成する各モジュールのプログラムを実行するプログラム実行リソースの数、及び、前記第1の画像処理部又は前記第2の画像処理部を構成する画像処理モジュールの数の少なくとも一方に応じて、前記並列処理制御手段を動作させるか逐次処理制御手段を動作させるかを選択することを特徴とする請求項7記載の画像処理装置。

【請求項9】

前記記憶手段は、前記並列処理制御手段のプログラム及び前記逐次処理制御手段のプログラムを各々記憶すると共に、外部から入力された新たな前記並列処理制御手段又は前記逐次処理制御手段のプログラムを、追加記憶又は既に記憶しているプログラムに上書き記憶可能とされ、前記選択手段は、前記記憶手段にプログラムが記憶されている全ての並列処理制御手段及び全ての逐次処理制御手段の中から動作させる処理制御手段を選択することを特徴とする請求項1乃至請求項8の何れか1項記載の画像処理装置。

【請求項10】

自モジュールの前段から画像データを取得し、取得した画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する画像処理モジュールとして各々機能すると共に、前記画像処理の種類又は内容が互いに異なる複数種の画像処理モジュールのプログラムを各々記憶する記憶手段を備えたコンピュータを、

前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第1の画像処理部を構築し、前記第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する

10

20

30

40

50

並列処理制御手段、

前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行わない第2のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第2の画像処理部を構築し、前記第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御する逐次処理制御手段、

10

及び、前記並列処理制御手段及び逐次処理制御手段を選択的に動作させ、前記処理対象の画像に対する画像処理を前記第1の画像処理部又は前記第2の画像処理部によって行わせる選択手段

として機能させ、

前記並列処理制御手段及び前記逐次処理制御手段は、前記第1の画像処理部又は第2の画像処理部の実行単位プログラムから状態通知が入力される毎に所定の処理を行って前記第1の画像処理部又は前記第2の画像処理部の状態を管理する状態管理手段を各々備え、前記並列処理制御手段に設けられている状態管理手段は、前記第1の画像処理部の実行単位プログラムから入力された状態通知を排他的に処理すると共に、エラーの発生を通知する状態通知が入力された場合に、エラー発生通知元以外の実行単位プログラムの実行を中止させる画像処理プログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像処理装置及びプログラムに係り、特に、画像処理モジュールの前段及び後段の少なくとも一方にバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る画像処理部を備えた画像処理装置、及び、コンピュータを前記画像処理装置として機能させるための画像処理プログラムに関する。

【背景技術】

30

【0002】

入力された画像データに対して画像処理を行う画像処理装置や、画像を取扱可能なDTP(デスクトップ・パブリッシング)システム、入力された画像データが表す画像を記録材料に記録するプリントシステム等では、入力された画像データに対して拡大・縮小、回転、アフィン変換、色変換、フィルタ処理、画像合成等の各種の画像処理が行われる。これらの装置やシステムにおいて、入力される画像データの属性や画像データに対する画像処理の内容・手順・パラメータ等が固定されている場合には、専用に設計したハードウェアによって画像処理を行わせる場合もあるが、例えば色空間や1画素当たりのビット数が異なる様々な画像データが入力されたり、画像処理の内容や手順・パラメータ等が様々に変更される場合には、実行する画像処理をより柔軟に変更可能な構成が必要となる。

40

【0003】

このような要求を満たすために、例えば特許文献1には、プログラマブルな処理モジュールをパイプライン形態やDAG(Directed Acyclic Graph: 有向非循環グラフ)形態に接続して、所望の画像処理を行うことを可能とする技術が提案されている。特許文献1に記載の技術では、複数のプログラマブル演算処理部の各々における演算処理の内容と、ネットワーク部による各プログラマブル演算処理部の接続形態を、ホストコントロール手段を通じて外部から自在に設定できるように構成することで、高速かつ高度な演算処理が可能で、機能変更や系統変更に対する自由度が高いデジタル映像信号処理装置を実現している。

【0004】

50

また、画像処理にも適用可能な技術として、特許文献2には、複数の記憶手段と、時分割された処理の一つである実行単位について、複数の記憶手段のうちの入力用の単一の記憶手段から入力データを受け取り、出力用の単一の記憶手段に処理結果を記憶するデータ処理手段と、記憶手段のデータ記憶量の情報から実行単位の実行状況を求め、実行単位の起動優先度を決定する起動優先度決定手段を備えたプロセッサが開示されている。

【特許文献1】特開平5-260373号公報

【特許文献2】特開2004-287883号公報

【発明の開示】

【発明が解決しようとする課題】

【0005】

画像処理の実行にあたって特許文献1に記載の技術を適用し、複数の画像処理モジュールを組み合わせて画像処理部を構築し所望の画像処理を行う場合の処理方式としては、個々の画像処理モジュールにおける画像処理を並列に行わせる並列処理方式と、常に単一の画像処理モジュールで画像処理を行わせると共に、画像処理を行わせる画像処理モジュールを逐次切り替える逐次処理方式が考えられる。このうち、並列処理方式は、例えばメモリ等のリソースが比較的潤沢な動作環境では逐次処理方式よりも処理速度の高速化が可能な一方、逐次処理方式は、メモリ等のリソースが比較的乏しい動作環境でも安定して動作する等、互いに異なる特徴を備えているため、装置構成や動作環境、同時に実行される他のアプリケーションプログラムとの兼ね合い等の様々な要因を考慮して処理方式を切替可能であることが望ましい。

【0006】

ここで、並列処理と逐次処理を切替可能とすることは、例えば並列処理用のプログラムと逐次処理用のプログラムを各々用意することで実現できるが、この場合、コンピュータを画像処理装置として機能させるためのプログラムの開発負荷が増大するという問題がある。特に、上記のプログラムのうち、画像処理モジュールとして機能するプログラムについては、既存のアルゴリズムの改良や、新たな画像処理を行う画像処理モジュールの追加等の更新・追加が比較的高い頻度で行われるが、画像処理モジュールのプログラムとして並列処理用のプログラムと逐次処理用のプログラムを各々用意することは、プログラム開発負荷の大幅な増大に繋がる。

【0007】

また、並列処理用のプログラムと逐次処理用のプログラムを共通化することも考えられるが、並列処理と逐次処理は、逐次処理では不要な排他制御が並列処理では必要となる点で相違しており、並列処理用のプログラムと逐次処理用のプログラムを単に共通化したとすると、逐次処理時にも不要な排他制御が行われることで、処理速度の低下やリソースの無駄な消費等の不都合が生ずる。また、並列処理方式では、個々の画像処理モジュールを独立したスレッドとして動作させると共に、各スレッドの実行優先度を切り替えることで処理効率を向上させることも可能であるが、並列処理用のプログラムと逐次処理用のプログラムを単に共通化したとすると、並列処理時に上記のような実行優先度の制御を行うことも困難である。

【0008】

本発明は上記事実を考慮して成されたもので、並列処理で動作させるか逐次処理で動作させるかを切替可能とすることを、プログラムの開発負荷の増大を抑制しつつ実現できる画像処理装置及び画像処理プログラムを得ることが目的である。

【課題を解決するための手段】

【0009】

上記目的を達成するために請求項1記載の発明に係る画像処理装置は、自モジュールの前段から画像データを取得し、取得した画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する画像処理モジュールとして各々機能すると共に、前記画像処理の種類又は内容が互いに異なる複数種の画像処理モジュールのプログラムを各々記憶する記憶手段と、前記記憶手段にプロ

10

20

30

40

50

グラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第1の画像処理部を構築し、前記第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する並列処理制御手段と、前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行わない第2のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第2の画像処理部を構築し、前記第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御する逐次処理制御手段と、前記並列処理制御手段及び逐次処理制御手段を選択的に動作させ、処理対象の画像に対する画像処理を前記第1の画像処理部又は前記第2の画像処理部によって行わせる選択手段と、を含み、前記並列処理制御手段及び前記逐次処理制御手段は、前記第1の画像処理部又は第2の画像処理部の実行単位プログラムから状態通知が入力される毎に所定の処理を行って前記第1の画像処理部又は前記第2の画像処理部の状態を管理する状態管理手段を各々備え、前記並列処理制御手段に設けられている状態管理手段は、前記第1の画像処理部の実行単位プログラムから入力された状態通知を排他的に処理すると共に、エラーの発生を通知する状態通知が入力された場合に、エラー発生通知元以外の実行単位プログラムの実行を中止させる。

【0010】

請求項1記載の発明では、自モジュールの前段から画像データを取得し、取得した画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する画像処理モジュールとして各々機能すると共に、前記画像処理の種類又は内容が互いに異なる複数種の画像処理モジュールのプログラムが記憶手段に各々記憶されている。また、請求項1記載の発明に係る並列処理制御手段は、記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第1の画像処理部を構築し、第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する。

【0011】

また、請求項1記載の発明に係る逐次処理制御手段は、記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、バッファへのアクセスに対する排他制御を行わない第2のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第2の画像処理部を構築し、第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御する。そして、請求項1記載の発明に係る選択手段は、並列処理制御手段及び逐次処理制御手段を選択的に動作させ、処理対象の画像に対する画像処理を

第1の画像処理部又は第2の画像処理部によって行わせる。

【0012】

このように、請求項1記載の発明では、選択手段によって並列処理制御手段が選択された場合には、並列処理制御手段によって第1の画像処理部が構築され、第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御される一方、選択手段によって逐次処理制御手段が選択された場合には、逐次処理制御手段によって第2の画像処理部が構築され、第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御されるので、画像処理部を並列処理で動作させるか逐次処理で動作させるかを切替えることができる。

【0013】

また、請求項1記載の発明では、並列処理用の第1の画像処理部の構築及び逐次処理用の第2の画像処理部の構築にあたり、記憶手段に記憶されている画像処理モジュールのプログラムが共通に使用されるので、画像処理モジュールのプログラムとして、並列処理用のプログラムと逐次処理用のプログラムを別に用意する必要がなくなり、並列処理で動作させるか逐次処理で動作させるかを切替可能とすることに伴って、プログラム（特に画像処理モジュールのプログラム）の開発負荷が増大することを抑制することができる。

【0014】

更に、請求項1記載の発明では、第1の画像処理部を構築する際には、画像処理モジュールの前段及び後段の少なくとも一方に、バッファへのアクセスに対する排他制御を行う第1のバッファモジュールを連結し、第2の画像処理部を構築する際には、画像処理モジュールの前段及び後段の少なくとも一方に、上記の排他制御を行わない第2のバッファモジュールを連結しているため、第2の画像処理部によって逐次処理を行う際に不要な排他制御が行われることはなく、処理速度の低下やリソースの無駄な消費等の不都合が生ずることも防止することができる。

また、請求項1記載の発明において、並列処理制御手段及び逐次処理制御手段は、第1の画像処理部又は第2の画像処理部の実行単位プログラムから状態通知が入力される毎に所定の処理を行って第1の画像処理部又は第2の画像処理部の状態を管理する状態管理手段を各々備えており、並列処理制御手段に設けられている状態管理手段は、第1の画像処理部の実行単位プログラムから入力された状態通知を排他的に処理すると共に、エラーの発生を通知する状態通知が入力された場合に、エラー発生通知元以外の実行単位プログラムの実行を中止させる。上記のように、状態管理手段を並列処理制御手段及び逐次処理制御手段に組み込むことで、選択手段が選択する処理制御手段が切り替わると（並列処理で動作させるか逐次処理で動作させるかが切り替わると）、状態管理手段も同時に切り替わることになり、並列処理で動作させるか逐次処理で動作させるかの切替時の処理が簡単になる。

【0015】

なお、請求項1記載の発明において、例えば請求項2に記載したように、記憶手段には第1のバッファモジュールのプログラム及び第2のバッファモジュールのプログラムも各々記憶することができ、並列処理制御手段は、記憶手段に記憶されている第1のバッファモジュールのプログラムを用いて第1の画像処理部を構築し、逐次処理制御手段は、記憶手段に記憶されている第2のバッファモジュールのプログラムを用いて第2の画像処理部を構築するように構成することができる。

【0016】

また、請求項1記載の発明において、画像処理モジュールとしては、画像処理の実行指示が入力される毎に、自モジュールの前段から画像データを所定量取得し、取得した所定量の画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は画像処理の処理結果を自モジュールの後段へ出力する構成を採用することができる。この場合、例えば請求項3に記載したように、並列処理制御手段は、第1の画像処理部を構成する画像処理モジュールのうちの一部又は全ての画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、第1の画

10

20

30

40

50

像処理部を構成する個々の画像処理モジュールで互いに並列に画像処理が行われるように、第1の画像処理部を構成する個々の画像処理モジュールへ画像処理の実行指示を入力するよう構成することができ、逐次処理制御手段は、第2の画像処理部を構成する画像処理モジュールのプログラムを、複数の画像処理モジュールのプログラムを実行単位プログラムとしてプログラム実行リソースにより実行させると共に、個々の画像処理モジュールで画像処理が逐次行われるように、第2の画像処理部を構成する個々の画像処理モジュールへ画像処理の実行指示を入力するよう構成することができる。

【0017】

また、請求項1記載の発明において、例えば請求項4に記載したように、並列処理制御手段として、第1の画像処理部を構成する個々の画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、個々の実行単位プログラムの実行優先度を固定的に設定する第1の並列処理制御手段、及び、第1の画像処理部を構成する個々の画像処理モジュールのプログラムを、互いに独立した実行単位プログラムとしてプログラム実行リソースにより実行させると共に、個々の実行単位プログラムの実行優先度を、第1の画像処理部における画像処理の進行度合に応じて変更する第2の並列処理制御手段の少なくとも一方を設けてもよい。特に、上記の第2の並列処理制御手段を設け、第1の画像処理部における画像処理の進行度合に応じて個々の実行単位プログラムの実行優先度を変更した場合、第1の画像処理部による画像処理の処理効率を向上させることができる。また、並列処理制御手段として、上記のように第1の画像処理部に行わせる並列処理の内容が異なる複数種の並列処理制御手段を各々設けた場合、選択手段が選択する並列処理制御手段を単に切り替えることで、第1の画像処理部に行わせる並列処理の内容を切り替えることも可能となる。

【0018】

また、請求項1記載の発明において、画像処理モジュールとして、画像処理の実行指示が入力される毎に、自モジュールの前段から画像データを所定量取得し、取得した所定量の画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は画像処理の処理結果を自モジュールの後段へ出力する構成をした場合、例えば請求項5に記載したように、逐次処理制御手段として、処理対象の画像データのうちの所定量の画像データに対する画像処理を、第2の画像処理部を構成する画像処理モジュールのうちパイプライン形態又は有向非循環グラフ形態における前段側の画像処理モジュールから順に行うことが、処理対象の画像データに対する画像処理が終了する迄繰り返されるように、画像処理の実行指示を入力する第1の逐次処理制御手段、及び、第2の画像処理部を構成する画像処理モジュールのうちの特定の画像処理モジュールで処理対象の画像データに対する画像処理が終了する迄、特定の画像処理モジュールへ画像処理の実行指示を繰り返し入力することを、パイプライン形態又は有向非循環グラフ形態における前段側の画像処理モジュールから順に行う第2の逐次処理制御手段の少なくとも一方を設けてもよい。また、特に逐次処理制御手段として、上記のように第2の画像処理部に行わせる逐次処理の内容が異なる複数種の並列処理制御手段を各々設けた場合、選択手段が選択する逐次処理制御手段を単に切り替えることで、第2の画像処理部に行わせる逐次処理の内容を切り替えることも可能となる。

【0020】

また、請求項1記載の発明において、選択手段は、例えば請求項6に記載したように、利用者によって予め入力された選択結果（何れの処理制御手段を動作させるかが利用者によって選択された結果）又は選択条件（個々の処理制御手段をどのような場合に選択するか（動作させるか）を規定する、利用者によって設定された条件）に基づいて、並列処理制御手段を動作させるか逐次処理制御手段を動作させるかを選択するよう構成することができる。なお、利用者による選択結果又は選択条件の入力は、例えば利用者が設定可能でかつ選択手段が参照可能な形態の情報（例えば所定の動作定義ファイルやレジストリ等）に選択結果又は選択条件を設定することで行うことができ、選択手段は、上記情報を参照することで入力された選択結果又は選択条件を認識することができる。この場合、利用

10

20

30

40

50

者が所望する処理方式（並列処理又は逐次処理）で画像処理を行わせることができる。

【0021】

また、請求項1記載の発明において、選択手段は、例えば請求項7に記載したように、並列処理制御手段を動作させるか逐次処理制御手段を動作させるか（すなわち画像処理部を並列処理で動作させるか逐次処理で動作させるか）を、第1の画像処理部又は第2の画像処理部が動作する環境に応じて選択するように構成することが好ましい。なお、上記の動作環境（を表す変数）としては、例えば請求項8に記載したように、第1の画像処理部又は第2の画像処理部を構成する各モジュールのプログラムを実行するプログラム実行リソースの数や、第1の画像処理部又は第2の画像処理部を構成する画像処理モジュールの数を適用することができ、上記の動作環境を表す変数を閾値（この閾値は請求項6に記載の選択条件として利用者によって設定された値であってもよい）と比較し、その大小関係に基づいて並列処理制御手段を動作させるか逐次処理制御手段を動作させるかを選択するように選択手段を構成することができる。これにより、画像処理部を並列処理で動作させるか逐次処理で動作させるかを、第1の画像処理部又は第2の画像処理部が動作する環境に応じて適切に選択することができる。

10

【0022】

また、請求項4に記載したように並列処理の内容には複数種のバリエーションがあり、請求項5に記載したように逐次処理の内容にも複数種のバリエーションがあることを考慮すると、請求項1乃至請求項8の何れかに記載の発明において、例えば請求項9に記載したように、記憶手段を、並列処理制御手段のプログラム及び逐次処理制御手段のプログラムを各々記憶すると共に、外部から入力された新たな並列処理制御手段又は逐次処理制御手段のプログラムを、追加記憶又は既に記憶しているプログラムに上書き記憶可能に構成し、選択手段を、記憶手段にプログラムが記憶されている全ての並列処理制御手段及び全ての逐次処理制御手段の中から動作させる処理制御手段を選択するように構成することが好ましい。これにより、並列処理制御手段や逐次処理制御手段のプログラムを後から追加したり更新することが可能となり、例えば個々の画像処理装置に対するニーズの相違等に応じて、様々な内容の並列処理や逐次処理の中から動作させる処理を選択したり、記憶手段に記憶させる処理制御手段のプログラムの数を必要最小限とすることで記憶手段の記憶容量を節減する等、画像処理装置の構成（機能）をニーズ等に応じて柔軟に変更することが可能となる。

20

30

【0023】

請求項10記載の発明に係る画像処理プログラムは、自モジュールの前段から画像データを取得し、取得した画像データに所定の画像処理を行い、当該画像処理を経た画像データ又は前記画像処理の処理結果を自モジュールの後段へ出力する画像処理モジュールとして各々機能すると共に、前記画像処理の種類又は内容が互いに異なる複数種の画像処理モジュールのプログラムを各々記憶する記憶手段を備えたコンピュータを、前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結されるように、個々のモジュールをパイプライン形態又は有向非循環グラフ形態で連結して成る第1の画像処理部を構築し、前記第1の画像処理部の個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する並列処理制御手段、前記記憶手段にプログラムが記憶されている複数種の画像処理モジュールの中から1つ以上の画像処理モジュールを選択し、前記選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、自モジュールの前段のモジュールから出力される画像データをバッファに書き込ませ、前記バッファに記憶されている画像データを自モジュールの後段のモジュールによって読み出させると共に、前記バッファへのアクセスに対する排他制御を行わない第2のバッファモジュールが連結されるように、個々のモジュールをパイプ

40

50

イン形態又は有向非循環グラフ形態で連結して成る第2の画像処理部を構築し、前記第2の画像処理部の個々の画像処理モジュールで画像処理が逐次行われるように制御する逐次処理制御手段、及び、前記並列処理制御手段及び逐次処理制御手段を選択的に動作させ、前記処理対象の画像に対する画像処理を前記第1の画像処理部又は前記第2の画像処理部によって行わせる選択手段として機能させ、前記並列処理制御手段及び前記逐次処理制御手段は、前記第1の画像処理部又は第2の画像処理部の実行単位プログラムから状態通知が入力される毎に所定の処理を行って前記第1の画像処理部又は前記第2の画像処理部の状態を管理する状態管理手段を各々備え、前記並列処理制御手段に設けられている状態管理手段は、前記第1の画像処理部の実行単位プログラムから入力された状態通知を排他的に処理すると共に、エラーの発生を通知する状態通知が入力された場合に、エラー発生通知元以外の実行単位プログラムの実行を中止させる。

10

【0024】

請求項10記載の発明に係る画像処理プログラムは、上記の記憶手段を備えたコンピュータを、上記の状態管理手段を備えた並列処理制御手段、状態管理手段を備えた逐次処理制御手段及び選択手段として機能させるためのプログラムであるので、上記のコンピュータが請求項10記載の発明に係る画像処理プログラムを実行することにより、上記のコンピュータが請求項1に記載の画像処理装置として機能することになり、請求項1記載の発明と同様に、並列処理で動作させるか逐次処理で動作させるかを切替可能とすることを、プログラムの開発負荷の増大を抑制しつつ実現することができ、並列処理で動作させるか逐次処理で動作させるかの切替時の処理も簡単になる。

20

【発明の効果】

【0025】

以上説明したように本発明は、複数種の画像処理モジュールの中から選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、バッファへのアクセスに対する排他制御を行う第1のバッファモジュールが連結して第1の画像処理部を構築し、個々の画像処理モジュールで互いに並列に画像処理が行われるように制御する並列処理制御手段、及び、複数種の画像処理モジュールの中から選択した個々の画像処理モジュールの前段及び後段の少なくとも一方に、バッファへのアクセスに対する排他制御を行わない第2のバッファモジュールを連結して第2の画像処理部を構築し、個々の画像処理モジュールで画像処理が逐次行われるように制御する逐次処理制御手段を設け、並列処理制御手段及び逐次処理制御手段を選択的に動作させるようにしたので、並列処理で動作させるか逐次処理で動作させるかを切替可能とすることを、プログラムの開発負荷の増大を抑制しつつ実現できる、という優れた効果を有する。

30

【発明を実施するための最良の形態】

【0026】

以下、図面を参照して本発明の実施形態の一例を詳細に説明する。図1には、本発明に係る画像処理装置として機能することが可能なコンピュータ10が示されている。なお、このコンピュータ10は、複写機、プリンタ、ファクシミリ装置、これらの機能を兼ね備えた複合機、スキャナ、写真プリンタ等のように内部で画像処理を行う必要のある任意の画像取扱機器に組み込まれていてもよいし、パーソナル・コンピュータ(PC)等の独立したコンピュータであってもよく、更にPDA(Personal Digital Assistant)や携帯電話機等の携帯機器に組み込まれたコンピュータであってもよい。

40

【0027】

コンピュータ10はCPU12、メモリ14、表示部16、操作部18、記憶部20、画像データ供給部22及び画像出力部24を備えており、これらはバス26を介して互いに接続されている。コンピュータ10が上述したような画像取扱機器に組み込まれている場合、表示部16や操作部18としては、画像取扱機器に設けられたLCD等から成る表示パネルやテンキー等を適用することができる。また、コンピュータ10が独立したコンピュータである場合、表示部16や操作部18としては、当該コンピュータに接続されたディスプレイやキーボード、マウス等を適用することができる。また、記憶部20として

50

はHDD(Hard Disk Drive)が好適であるが、これに代えてフラッシュメモリ等の他の不揮発性記憶手段を用いることも可能である。

【0028】

また、画像データ供給部22は処理対象の画像データを供給できるものであればよく、例えば紙や写真フィルム等の記録材料に記録されている画像を読み取って画像データを出力する画像読取部、通信回線を介して外部から画像データを受信する受信部、画像データを記憶する画像記憶部(メモリ14又は記憶部20)等を適用することができる。また、画像出力部24は画像処理を経た画像データ又は該画像データが表す画像を出力するものであればよく、例えば画像データが表す画像を紙や感光材料等の記録材料に記録する画像記録部、画像データが表す画像をディスプレイ等に表示する表示部、画像データを記録メディアに書き込む書込装置、画像データを通信回線を介して送信する送信部を適用することができる。また、画像出力部24は画像処理を経た画像データを単に記憶する画像記憶部(メモリ14又は記憶部20)であっても構わない。

10

【0029】

図1に示すように、記憶部20には、CPU12によって実行される各種のプログラムとして、メモリ14等のリソースの管理やCPU12によるプログラムの実行の管理、コンピュータ10と外部との通信等を司るオペレーティングシステム30のプログラム、コンピュータ10を本発明に係る画像処理装置として機能させるための画像処理プログラム群34、CPU12が上記画像処理プログラム群を実行することで実現される画像処理装置に対して所望の画像処理を行わせる各種のアプリケーション32のプログラム(図1ではアプリケーションプログラム群32と表記)が各々記憶されている。

20

【0030】

画像処理プログラム群34は、前述した各種の画像取扱機器や携帯機器を開発する際の開発負荷を軽減したり、PC等で利用可能な画像処理プログラムを開発する際の開発負荷を軽減することを目的として、各種の画像取扱機器や携帯機器、PC等の各種機器(プラットフォーム)で共通に使用可能に開発されたプログラムであり、本発明に係る画像処理プログラムに対応している。画像処理プログラム群34によって実現される画像処理装置は、アプリケーション32からの構築指示に従い、アプリケーション32が指示した画像処理を行う画像処理部を構築し、アプリケーション32からの実行指示に従い、前記画像処理部によって画像処理を行うが(詳細は後述)、画像処理プログラム群34は、所望の画像処理を行う画像処理部(所望の構成の画像処理部)の構築を指示したり、構築された画像処理部による画像処理の実行を指示するためのインタフェースをアプリケーション32に提供している。このため、内部で画像処理を行う必要のある任意の機器を新規開発する等の場合にも、前記画像処理を行うプログラムの開発に関しては、当該機器で必要とされる画像処理を上記のインタフェースを利用して画像処理プログラム群34に行わせるアプリケーション32を開発するのみで済み、実際に画像処理を行うプログラムを新たに開発する必要が無くなるので、開発負荷を軽減することができる。

30

【0031】

また、画像処理プログラム群34によって実現される画像処理装置は、前述のように、アプリケーション32からの構築指示に従い、アプリケーション32が指示した画像処理を行う画像処理部を構築し、構築した画像処理部によって画像処理を行うので、例えば画像処理対象の画像データの色空間や1画素当たりのビット数が不定であったり、実行すべき画像処理の内容や手順・パラメータ等が不定である場合にも、アプリケーション32が画像処理部の再構築を指示することで、画像処理装置(画像処理部)によって実行される画像処理を、処理対象の画像データ等に応じて柔軟に変更することができる。

40

【0032】

以下、画像処理プログラム群34について説明する。図1に示すように、画像処理プログラム群34はモジュールライブラリ36と、処理構築部42のプログラムと、処理管理部ライブラリ47に大別される。詳細は後述するが、本実施形態に係る処理構築部42は、アプリケーションからの指示により、例として図3に示すように、予め定められた画像

50

処理を行う1つ以上の画像処理モジュール38と、個々の画像処理モジュール38の前段及び後段の少なくとも一方に配置され画像データを記憶するためのバッファを備えたバッファモジュール40と、がパイプライン形態又はDAG(Directed Acyclic Graph:有向非循環グラフ)形態で連結されて成る画像処理部50を構築する。画像処理部50を構成する個々の画像処理モジュールの実体はCPU12によって実行されCPU12で所定の画像処理を行わせるための第1プログラム、又は、CPU12によって実行されCPU12により図1に図示されていない外部の画像処理装置(例えば専用画像処理ボード等)に対する処理の実行を指示するための第2プログラムであり、上述したモジュールライブラリ36には、予め定められた互いに異なる画像処理(例えば入力処理やフィルタ処理、色変換処理、拡大・縮小処理、スキュー角検知処理、画像回転処理、画像合成処理、出力処理等)を行う複数種の画像処理モジュール38のプログラムが各々登録されている。以下では、説明を簡単にするために、画像処理部50を構成する個々の画像処理モジュールの実体が上記の第1プログラムであるものとして説明する。

10

【0033】

個々の画像処理モジュール38は、例として図4(A)にも示すように、画像データに対する画像処理を所定の単位処理データ量ずつ行う画像処理エンジン38Aと、画像処理モジュール38の前段及び後段のモジュールとの画像データの入出力及び画像処理エンジン38Aの制御を行う制御部38Bから構成されている。個々の画像処理モジュール38における単位処理データ量は、画像の1ライン分、画像の複数ライン分、画像の1画素分、画像1面分等を含む任意のバイト数の中から、画像処理エンジン38Aが行う画像処理の種類等に応じて予め選択・設定されており、例えば色変換処理やフィルタ処理を行う画像処理モジュール38では単位処理データ量が1画素分とされ、拡大・縮小処理を行う画像処理モジュール38では単位処理データ量が画像の1ライン分又は画像の複数ライン分とされ、画像回転処理を行う画像処理モジュール38では単位処理データ量が画像1面分とされ、画像圧縮伸長処理を行う画像処理モジュール38では単位処理データ量が実行環境に依存するNバイトとされている。

20

【0034】

また、モジュールライブラリ36には、画像処理エンジン38Aが実行する画像処理の種類が同一でかつ実行する画像処理の内容が異なる画像処理モジュール38も登録されている(図1では、この種の画像処理モジュールを「モジュール1」「モジュール2」と表記して示している)。例えば拡大・縮小処理を行う画像処理モジュール38については、入力された画像データを1画素おきに間引くことで50%に縮小する縮小処理を行う画像処理モジュール38、入力された画像データに対して指定された拡大・縮小率で拡大・縮小処理を行う画像処理モジュール38等の複数の画像処理モジュール38が各々用意されている。また、例えば色変換処理を行う画像処理モジュール38については、RGB色空間をCMY色空間へ変換する画像処理モジュール38やその逆へ変換する画像処理モジュール38、L*a*b*色空間等の他の色空間変換を行う画像処理モジュール38が各々用意されている。

30

【0035】

また、画像処理モジュール38の制御部38Bは、画像処理エンジン38Aが単位処理データ量ずつ処理するために必要な画像データを入力するために、自モジュールの前段のモジュール(例えばバッファモジュール40)から画像データを単位読出データ量ずつ取得し、画像処理エンジン38Aから出力される画像データを単位書込データ量ずつ後段のモジュール(例えばバッファモジュール40)へ出力する(画像処理エンジン38Aで圧縮等のデータ量の増減を伴う画像処理が行われなければ単位書込データ量=単位処理データ量となる)か、画像処理エンジン38Aによる画像処理の結果を自モジュールの外部へ出力する(例えば画像処理エンジン38Aがスキュー角検知処理等の画像解析処理を行う場合、画像データに代えてスキュー角検知結果等の画像解析処理結果が出力されることがある)処理を行うが、モジュールライブラリ36には、画像処理エンジン38Aが実行する画像処理の種類及び内容が同一で、上記の単位処理データ量や単位読出データ量、単位書

40

50

込データ量が異なる画像処理モジュール38も登録されている。例えば、画像回転処理を行う画像処理モジュール38についても、前述のように単位処理データ量が画像1面分の画像処理モジュール38のプログラム以外に、単位処理データ量が画像の1ライン分や画像の複数ライン分の画像処理モジュール38のプログラムがモジュールライブラリ36に登録されていても良い。

【0036】

また、モジュールライブラリ36に登録されている個々の画像処理モジュール38のプログラムは、画像処理エンジン38Aに相当するプログラムと制御部38Bに相当するプログラムから構成されているが、制御部38Bに相当するプログラムは部品化されており、個々の画像処理モジュール38のうち単位読出データ量及び単位書込データ量が同一の画像処理モジュール38は、画像処理エンジン38Aで実行される画像処理の種類や内容に拘わらず、制御部38Bに相当するプログラムが共通化されている(制御部38Bに相当するプログラムとして同一のプログラムが用いられている)。これにより、画像処理モジュール38のプログラムの開発にあたっての開発負荷が軽減される。

【0037】

なお、画像処理モジュール38の中には、入力される画像の属性が未知の状態では単位読出データ量及び単位書込データ量が確定しておらず、入力画像データの属性を取得し、取得した属性を所定の演算式に代入して演算することで単位読出データ量や単位書込データ量が確定するモジュールが存在しているが、この種の画像処理モジュール38については、単位読出データ量と単位書込データ量が互いに同一の演算式を用いて導出される画像処理モジュール38について、制御部38Bに相当するプログラムを共通化するようにすればよい。また、本実施形態に係る画像処理プログラム群34は、前述のように各種機器に実装可能であるが、画像処理プログラム群34のうちモジュールライブラリ36に登録する画像処理モジュール38の数や種類等については、画像処理プログラム群34を実装する各種機器で必要とされる画像処理に応じて、適宜追加・削除・入替等が可能であることは言うまでもない。

【0038】

また、画像処理部50を構成する個々のバッファモジュール40は、例として図4(B)にも示すように、コンピュータ10に設けられたメモリ14からオペレーティングシステム30を通じて確保されたメモリ領域で構成されるバッファ40Aと、バッファモジュール40の前段及び後段のモジュールとの画像データの入出力及びバッファ40Aの管理を行うバッファ制御部40Bから構成されている。個々のバッファモジュール40のバッファ制御部40Bもその実体はCPU12によって実行されるプログラムであり、モジュールライブラリ36にはバッファ制御部40Bのプログラムも登録されている(図1ではバッファ制御部40Bのプログラムを「バッファモジュール」と表記して示している)。

【0039】

なお、詳細は後述するが、画像処理部50における画像処理の処理方式としては、画像処理部50の個々の画像処理モジュール38で互いに並列に画像処理を行わせる並列処理方式と、画像処理部50の画像処理モジュール38のうちの常に単一の画像処理モジュール38で画像処理を行わせると共に、画像処理を行わせる画像処理モジュール38を逐次切り替える逐次処理方式があり、本実施形態では、並列処理方式で画像処理を行わせる場合には並列処理用の画像処理部50を構築し、逐次処理方式で画像処理を行わせる場合には逐次処理用の画像処理部50を構築することで、画像処理部50における画像処理の処理方式を切り替えるようになっている。

【0040】

ここで、逐次処理方式では画像処理を行っている画像処理モジュール38の数が常に1であるので、バッファモジュール40のバッファ40Aに対しても常に単一の画像処理モジュール38のみがアクセスするのに対し、並列処理方式では全ての画像処理モジュール38が並列に画像処理を行うので、バッファモジュール40のバッファ40Aに対して複数の画像処理モジュール38が同時にアクセスする可能性があり、バッファ40Aへのア

10

20

30

40

50

クセスに対して排他制御を行う必要がある。このため、本実施形態では、並列処理用として排他制御を行う機能が付加されたバッファモジュール40が、逐次処理用として排他制御を行う機能が付加されないバッファモジュール40が各々用意され、モジュールライブラリ36には、排他制御を行う機能が付加されたバッファモジュール40のプログラムと排他制御を行う機能が付加されていないバッファモジュール40のプログラムが各々登録されている。

【0041】

また、アプリケーション32からの指示に従って画像処理部50を構築する処理構築部42は、図1に示すように複数種のモジュール生成部44から構成されている。複数種のモジュール生成部44は互いに異なる画像処理に対応しており、アプリケーション32によって起動されることで、対応する画像処理を実現するための画像処理モジュール38及びバッファモジュール40から成るモジュール群を生成する処理を行う。なお、図1ではモジュール生成部44の一例として、モジュールライブラリ36に登録されている個々の画像処理モジュール38が実行する画像処理の種類に対応するモジュール生成部44を示しているが、個々のモジュール生成部44に対応する画像処理は、複数種の画像処理モジュール38によって実現される画像処理（例えばスキュー角検知処理と画像回転処理から成るスキュー補正処理）であってもよい。必要とされる画像処理が複数種の画像処理を組み合わせた処理である場合、アプリケーション32は複数種の画像処理の何れかに対応するモジュール生成部44を順次起動する。これにより、アプリケーション32によって順次起動されたモジュール生成部44により、必要とされる画像処理を行う画像処理部50が構築されることになる。

【0042】

また図1に示すように、処理管理部ライブラリ47には、処理管理部46のプログラムが複数登録されている。個々の処理管理部46は、画像処理部50における画像処理の実行を制御するワークフロー管理部46A、画像処理部50の各モジュールによるメモリ14や各種のファイル等のコンピュータ10のリソースの使用を管理するリソース管理部46B、及び、画像処理部50で発生したエラーを管理するエラー管理部46Cを含んで構成される。処理管理部ライブラリ47にプログラムが登録されている処理管理部46は、並列処理用の画像処理部50（バッファモジュール40として排他制御機能付きのバッファモジュール40を適用した画像処理部50）を構築させ、構築させた画像処理部50で並列処理方式によって画像処理が行われるように制御する並列処理管理部と、逐次処理用の画像処理部50（バッファモジュール40として排他制御機能無しのバッファモジュール40を適用した画像処理部50）を構築させ、構築させた画像処理部50で逐次処理方式によって画像処理が行われるように制御する逐次処理管理部に大別される。

【0043】

また、図1では並列処理管理部のプログラム及び逐次処理管理部のプログラムを各々1個のみ示しているが、処理管理部ライブラリ47には、並列処理管理部のプログラムとして、互いに内容の異なる並列処理を並列処理用の画像処理部50に行わせる複数種の並列処理管理部のプログラムを各々登録可能とされていると共に、逐次処理管理部のプログラムとして、互いに内容の異なる逐次処理を逐次処理用の画像処理部50に行わせる複数種の逐次処理管理部のプログラムを各々登録可能とされている。前述の処理構築部42は選択起動部45を備えており、選択起動部45は、アプリケーション32から画像処理部50の構築が指示された際に、処理管理部ライブラリ47に登録されている各処理管理部の中から何れかの処理管理部を選択的に起動する。

【0044】

なお、並列処理管理部は本発明に係る並列処理制御手段に対応しており、逐次処理管理部は本発明に係る逐次処理制御手段に対応している。

【0045】

次に本実施形態の作用を説明する。画像処理プログラム群34が実装されている機器において、何らかの画像処理を行う必要のある状況になると、この状況が特定のアプリケー

10

20

30

40

50

ション32によって検知される。なお、画像処理を行う必要のある状況としては、例えば画像データ供給部22としての画像読取部によって画像を読み取り、画像出力部24としての画像記録部により記録材料に画像として記録するか、画像出力部24としての表示部に画像として表示させるか、画像出力部24としての書込装置により画像データを記録メディアに書き込むか、画像出力部24としての送信部により画像データを送信するか、画像出力部24としての画像記憶部に記憶させるジョブの実行がユーザによって指示された場合、或いは、画像データ供給部22としての受信部によって受信されるか、画像データ供給部22としての画像記憶部に記憶されている画像データに対して、上記の記録材料への記録、表示部への表示、記録メディアへの書き込み、送信、画像記憶部への記憶の何れかを行うジョブの実行がユーザによって指示された場合が挙げられる。また、画像処理を行う必要のある状況は上記に限られるものではなく、例えばユーザからの指示に応じてアプリケーション32が実行可能な処理の名称等を表示部16に一覧表示している状態で、実行対象の処理がユーザによって選択された等の場合であってもよい。

10

【0046】

上記のように、何らかの画像処理を行う必要のある状況になったことを検知すると、アプリケーション32は処理構築部42の選択起動部45を起動し(図2のステップ150も参照)、アプリケーション32によって起動された選択起動部45は、まず処理方式選択情報を参照する(図2のステップ152も参照)。この処理方式選択情報は、利用者が、画像処理部50における処理方式としては並列処理と逐次処理の何れが適当か等を検討することで、処理管理部ライブラリ47にプログラムが登録されている各処理管理部46の中から起動対象の処理管理部46を選択した結果を表す情報(請求項6に記載の「選択結果」に相当)であり、利用者により、例えば動作定義ファイル(拡張子が".ini"のファイル等)やレジストリ等のように、利用者が設定可能でかつ選択手段が参照可能な情報に設定される。選択起動部45は処理方式選択情報を参照することで、起動対象として指示された処理管理部46を認識し、認識した処理管理部46を起動する(図2のステップ156も参照)。起動された処理管理部46は稼働状態となり、外部から何らかの要求や指示(後述するバッファモジュール生成要求や画像処理実行指示)が入力される迄待機する。

20

【0047】

なお、処理方式選択情報は、上記のように起動対象の処理管理部46を直接指定する情報に限られるものではなく、処理管理部ライブラリ47にプログラムが登録されている個々の処理管理部46を、画像処理部50がどのような動作環境である場合に起動対象として選択するかを規定する選択条件を記述した情報(請求項6に記載の「選択条件」に相当)であってもよい。この態様における環境変数としては、例えばコンピュータ10に設けられているプログラム実行リソース(例えばCPU12等)の数をを用いることができ、例えば「プログラム実行リソースの数がN個以上であれば並列処理管理部46を起動し、N個未満であれば逐次処理管理部46を起動する」等の条件を処理方式選択情報として記述することができる。

30

【0048】

処理方式選択情報が上記のように選択条件を記述した情報である場合、選択起動部45は、例えば処理方式選択情報を参照(図2のステップ152も参照)して上記の選択条件を認識した後に、認識した選択条件で使用されている環境変数(例えばCPU12の数等)をオペレーティングシステム30等を通じて取得し(図2のステップ154も参照)、取得した環境変数を選択条件に規定されている閾値(例えば前述のN)と比較することで、起動対象として選択すべき処理管理部46を判断し、起動対象として選択した処理管理部46を起動する(図2のステップ156も参照)。この場合、画像処理部50における処理方式を、画像処理部50の動作環境に応じて適切に選択することができる。

40

【0049】

また、上記の環境変数としては、例えば画像処理部50を構成する画像処理モジュール38の数等の他の変数を用いることも可能であり、例えば「プログラム実行リソースの数

50

「画像処理モジュールの数」と閾値（例えば1やその前後の値）との大小関係に応じて、起動対象として選択する処理管理部46を切り替えるようにしてもよい。また、処理方式選択情報は利用者によって設定されることに限られるものでもなく、事前に（画像処理プログラム群34の開発時に）設定しておくようにしてもよい。なお、上記処理を行う選択起動部45は本発明に係る選択手段に対応している。

【0050】

一方、アプリケーション32は選択起動部45の起動を完了すると、画像処理対象の画像データを供給する画像データ供給部22の種別を認識する（図2のステップ158も参照）。また、認識した種別がバッファ領域（メモリ14の一部領域）であった場合、アプリケーション32は、画像データ供給部22として指定されたバッファ領域を稼働中の処理管理部46へ通知し、画像データ供給部22として機能するバッファモジュール40の生成を処理管理部46へ要求する。この場合、処理管理部46はバッファ制御部40BのプログラムをCPU12が実行可能なようにメモリ14にロードすると共に、通知されたバッファ領域（画像データ供給部22として指定されたバッファ領域）を既に確保されたバッファ40Aとしてバッファ制御部40Bに認識させるパラメータを設定することで、画像データ供給部22として機能するバッファモジュール40を生成し（図2のステップ160も参照）、アプリケーション32へ応答を返す。なお、稼働中の処理管理部46が並列処理管理部46である場合は、上記のバッファモジュール40として排他制御機能付きのバッファモジュール40が生成され、稼働中の処理管理部46が逐次処理管理部46である場合は、上記のバッファモジュール40として排他制御機能無しのバッファモジュール40が生成される。

【0051】

続いてアプリケーション32は、画像処理を行った画像データの出力先としての画像出力部24の種別を認識する（図2のステップ162も参照）。また、認識した種別がバッファ領域（メモリ14の一部領域）であった場合、アプリケーション32は、画像出力部24として指定されたバッファ領域を稼働中の処理管理部46へ通知し、画像出力部24として指定されたバッファ領域を含むバッファモジュール40（画像出力部24として機能するバッファモジュール40）を処理管理部46によって生成させる（図2のステップ158も参照）。このときも、稼働中の処理管理部46が並列処理管理部46であれば、上記のバッファモジュール40として排他制御機能付きのバッファモジュール40が生成され、稼働中の処理管理部46が逐次処理管理部46であれば、上記のバッファモジュール40として排他制御機能無しのバッファモジュール40が生成される。

【0052】

次にアプリケーション32は、実行すべき画像処理の内容を認識し、実行すべき画像処理を、個々のモジュール生成部44に対応するレベルの画像処理の組み合わせに分解し、実行すべき画像処理を実現するために必要な画像処理の種類及び個々の画像処理の実行順序を判定する（図2のステップ166も参照）。なお、この判定は、例えば上記の画像処理の種類及び個々の画像処理の実行順序を、ユーザが実行を指示可能なジョブの種類と対応付けて予め情報として登録しておき、アプリケーション32は、実行が指示されたジョブの種類に対応する情報を読み出すことによって実現することができる。

【0053】

そしてアプリケーション32は、上記で判定した画像処理の種類及び実行順序に基づいて、特定の画像処理に対応するモジュール生成部44を起動し（図2のステップ168も参照）、起動したモジュール生成部44に対し、当該モジュール生成部44によるモジュール群の生成に必要な情報として、前記モジュール群に画像データを入力する入力モジュールを識別するための入力モジュール識別情報、前記モジュール群が画像データを出力する出力モジュールを識別するための出力モジュール識別情報、前記モジュール群に入力される入力画像データの属性を表す入力画像属性情報、実行すべき画像処理のパラメータを通知して対応するモジュール群の生成を指示する（図2のステップ170も参照）。また、必要とされる画像処理が複数種の画像処理を組み合わせた処理である場合、アプリケー

10

20

30

40

50

ション32は、指示したモジュール生成部44からモジュール群の生成完了が通知されると、個々の画像処理に対応する他のモジュール生成部44を起動してモジュール群の生成に必要な情報を通知する処理(図2のステップ168, 170)を個々の画像処理の実行順序の昇順に繰り返す。

【0054】

なお、上記の入力モジュールは、実行順序が1番目のモジュール群については画像データ供給部22が入力モジュールとなり、実行順序が2番目以降のモジュール群については前段のモジュール群の最終モジュール(通常はバッファモジュール40)が入力モジュールとなる。また、上記の出力モジュールについては、実行順序が最後のモジュール群では画像出力部24が出力モジュールとなるので、画像出力部24が出力モジュールとして指定されるが、その他のモジュール群では出力モジュールは未確定のためにアプリケーション32による指定は行われず、必要な場合はモジュール生成部44によって生成・設定される。また、入力画像属性や画像処理のパラメータについては、例えばユーザが実行を指示可能なジョブの種類と対応付けて予め情報として登録しておき、実行が指示されたジョブの種類に対応する情報を読み出すことでアプリケーション32が認識するようにしてもよいし、ユーザに指定させるようにしてもよい。

【0055】

一方、モジュール生成部44は、アプリケーション32によって起動されるとモジュール生成処理(図2のステップ172も参照)を行う。モジュール生成処理では、まず生成対象の画像処理モジュール38に入力される入力画像データの属性を表す入力画像属性情報を取得する。なお、入力画像データの属性を取得する処理は、生成対象の画像処理モジュール38の前段にバッファモジュール40が存在している場合、当該バッファモジュール40に画像データの書き込みを行う更に前段の画像処理モジュール38から出力画像データの属性を取得することによって実現できる。

【0056】

そして、取得した情報が表す入力画像データの属性に基づいて、生成対象の画像処理モジュール38の生成が必要か否か判定する。例えばモジュール生成部44が色変換処理を行うモジュール群を生成するモジュール生成部であり、画像処理のパラメータにより出力画像データの色空間としてCMY色空間がアプリケーション32から指定された場合、取得した入力画像属性情報に基づいて入力画像データがRGB色空間のデータであることが判明したときには、色空間処理を行う画像処理モジュール38としてRGB-CMYの色空間変換を行う画像処理モジュール38を生成する必要があるが、入力画像データがCMY色空間のデータであったときには、入力画像データの属性と出力画像データの属性が色空間に関して一致しているので、色空間変換処理を行う画像処理モジュール38は生成不要と判断する。

【0057】

生成対象の画像処理モジュール38の生成が必要と判断した場合には、生成対象の画像処理モジュール38の後段にバッファモジュール40が必要か否かを判定する。この判定は、画像処理モジュールの後段が出力モジュール(画像出力部24)である場合(例えば図3(A)~(C)に示す画像処理部50における最後段の画像処理モジュール38を参照)や、例として図3(B)に示す画像処理部50においてスキュー角検知処理を行う画像処理モジュール38のように、画像処理モジュールが、画像データに対して解析等の画像処理を行いその結果を他の画像処理モジュール38へ出力するモジュールである場合は否定されるが、上記以外の場合は判定が肯定され、稼働中の処理管理部46に対して画像処理モジュール38の後段に連結するバッファモジュール40を生成を要求する。

【0058】

バッファモジュール40の生成が要求されると、処理管理部46はバッファ制御部40BのプログラムをCPU12が実行可能なようにメモリ14にロードすることで、バッファモジュール40を生成し(図2のステップ172も参照)、モジュール生成部44へ応答を返す。なお、稼働中の処理管理部46が並列処理管理部46であれば、上記のバッ

10

20

30

40

50

ァモジュール40として排他制御機能付きのバッファモジュール40が生成され、稼働中の処理管理部46が逐次処理管理部46であれば、上記のバッファモジュール40として排他制御機能無しのバッファモジュール40が生成される。

【0059】

続いてモジュール生成部44は、前段のモジュール(例えばバッファモジュール40)の情報、後段のバッファモジュール40の情報(後段にバッファモジュール40を生成した画像処理モジュール38のみ)、画像処理モジュール38に入力される入力画像データの属性、処理パラメータを与えて、モジュールライブラリ36に登録されており、画像処理モジュール38として利用可能な複数の候補モジュールの中から、先に取得した入力画像データの属性、及び、画像処理モジュール38で実行すべき処理パラメータに合致する画像処理モジュール38を選択し、選択した画像処理モジュール38のプログラムをCPU12が実行可能なようにメモリ14にロードすると共に、当該画像処理モジュール38の前段及び後段のモジュールを当該画像処理モジュール38の制御部38Bに認識させるパラメータを設定することで、画像処理モジュール38を生成する。

10

【0060】

例えばモジュール生成部44が色変換処理を行うモジュール群を生成するモジュール生成部であり、処理パラメータにより出力画像データの色空間としてCMY色空間が指定され、更に入力画像データがRGB色空間のデータであった場合には、モジュールライブラリ36に登録されている各種の色空間処理を行う複数種の画像処理モジュール38の中から、RGB CMYの色空間変換を行う画像処理モジュール38が選択・生成される。また、画像処理モジュールが拡大・縮小処理を行う画像処理モジュール38であり、指定された拡大縮小率が50%以外であれば、入力された画像データに対して指定された拡大・縮小率で拡大・縮小処理を行う画像処理モジュール38が選択・生成され、指定された拡大縮小率が50%であれば、拡大縮小率50%に特化した拡大縮小処理、すなわち入力された画像データを1画素おきに間引くことで50%に縮小する縮小処理を行う画像処理モジュール38が選択・生成される。

20

【0061】

なお、画像処理モジュール38の選択は上記に限られるものではなく、例えば画像処理エンジン38Aによる画像処理における単位処理データ量が異なる画像処理モジュール38をモジュールライブラリ36に複数登録しておき、画像処理部50へ割当可能なメモリ領域のサイズ等の動作環境に応じて、適切な単位処理データ量の画像処理モジュール38を選択する(例えば上記サイズが小さくなるに従って単位処理データ量の小さい画像処理モジュール38を選択する等)ようにしてもよいし、アプリケーション32或いはユーザに選択させるようにしてもよい。

30

【0062】

画像処理モジュール38の生成が完了すると、後段のバッファモジュール40のIDと生成した画像処理モジュール38のIDの組を稼働中の処理管理部46に通知する。このIDは、個々のモジュールを一意に判別できる情報であればよく、例えば個々のモジュールの生成順に付与した番号や、バッファモジュール40や画像処理モジュール38のオブジェクトのメモリ上でのアドレス等でも良い。またモジュール生成部44が、複数種の画像処理モジュール38によって実現される画像処理(例えばスキュー角検知処理を行う画像処理モジュール38と画像回転処理を行う画像処理モジュール38によって実現されるスキュー補正処理)を行うモジュール群を生成する場合には、上記処理が繰り返されて2個以上の画像処理モジュール38を含むモジュール群が生成される。アプリケーション32によって順次起動された個々のモジュール生成部44により、以上のモジュール生成処理が順次行われることで、例として図3(A)~(C)に示すように、必要とされる画像処理を行う画像処理部50が構築される。

40

【0063】

一方、アプリケーション32は、順次起動したモジュール生成部44によって前述のモジュール生成処理が順次行われることで、必要とされる画像処理を行う画像処理部50の

50

構築が完了すると、稼働中の処理管理部 4 6 に対して画像処理部 5 0 による画像処理の実行を指示する（図 2 のステップ 1 7 4 も参照）。処理管理部 4 6 は、アプリケーション 3 2 から画像処理の実行が指示されると、メモリ 1 4 にロードした画像処理部 5 0 の各モジュールのプログラムを、オペレーティングシステム 3 0 を通じてスレッドとして CPU 1 2 に実行させる。ここで、稼働中の処理管理部 4 6 が並列処理管理部である場合、処理管理部 4 6 は、画像処理部 5 0 の個々の画像処理モジュール 3 8 で互いに並列に画像処理を行わせるために、画像処理部 5 0 を構成する個々のモジュールのプログラムを互いに独立したスレッド（請求項 3 に記載の実行単位プログラムに相当）として CPU 1 2 に実行させる。また、稼働中の処理管理部 4 6 が逐次処理管理部である場合、処理管理部 4 6 は、画像処理部 5 0 を構成する個々のモジュールのプログラムを単一のスレッドとして CPU 1 2 に実行させる。なお、スレッドに代えて、プロセス又はオブジェクトとして CPU 1 2 に実行させるようにしてもよい。

10

【 0 0 6 4 】

画像処理モジュール 3 8 のプログラムがスレッドとして実行されると、個々の画像処理モジュール 3 8 の制御部 3 8 B は自モジュールの初期化を行う。画像処理モジュール 3 8 の初期化では、まずモジュール生成部 4 4 によって設定されたパラメータに基づいて自モジュールの前段のモジュールを判定する。自モジュールの前段にモジュールが存在していない場合には何ら処理を行わないが、前段のモジュールがバッファモジュール 4 0 以外、例えば画像データ供給部 2 2 や特定のファイル等である場合には、必要に応じてその初期化処理を行う。また、自モジュールの前段にバッファモジュール 4 0 が存在している場合

20

【 0 0 6 5 】

この単位読出データ量は、自モジュールの前段のバッファモジュール 4 0 の数が 1 個であれば 1 個だけであるが、例えば図 3 (C) に示す画像処理部 5 0 において画像合成処理を行う画像処理モジュール 3 8 のように、前段のバッファモジュール 4 0 の数が複数で、複数のバッファモジュール 4 0 から各々取得した画像データを用いて画像処理エンジン 3 8 A が画像処理を行う等の場合、前段の個々のバッファモジュール 4 0 に対応する単位読出データ量は、自モジュールの画像処理エンジン 3 8 A が行う画像処理の種類や内容、前段のバッファモジュール 4 0 の数等に応じて定まる。そして、認識した単位読出データ量を

30

【 0 0 6 6 】

次に、自モジュールの後段のモジュールを判定する。自モジュールの後段のモジュールがバッファモジュール 4 0 以外、例えば画像出力部 2 4 や特定のファイル等の場合には、必要に応じてその初期化処理（例えば後段のモジュールが画像出力部 2 4 であれば、単位書込データ量に相当するデータ量ずつ画像データを出力することを通知する処理等）を行う。また、後段のモジュールがバッファモジュール 4 0 であれば、1 回の画像データの書き込みにおける画像データのデータ量（単位書込データ量）を認識し、後段のバッファモジュールに当該単位書込データ量を設定（図 4 (A) の(2)も参照）する。そして、当該画像処理モジュール 3 8 の初期化の完了を処理管理部 4 6 通知する。

40

【 0 0 6 7 】

また、バッファモジュール 4 0 （のバッファ制御部 4 0 B）のプログラムがスレッドとして実行されると、個々のバッファモジュール 4 0 のバッファ制御部 4 0 B は自モジュールの初期化を行う。バッファモジュール 4 0 の初期化では、まず自モジュールの前段の画像処理モジュール 3 8 から単位書込データ量が通知されるか又は自モジュールの後段の画像処理モジュール 3 8 から単位読出データ量が通知される毎に、通知された単位書込データ量又は単位読出データ量を記憶する（図 3 (B) の(1), (2)も参照）。

【 0 0 6 8 】

50

自モジュールと連結されている全ての画像処理モジュール38から単位書込データ量又は単位読出データ量が通知されると、自モジュールと連結されている個々の画像処理モジュール38によって各々設定された単位書込データ量及び単位読出データ量に基づいて、自モジュールのバッファ40Aの管理単位である単位バッファ領域のサイズを決定し、決定した単位バッファ領域のサイズを記憶する。単位バッファ領域のサイズとしては、自モジュールに設定された単位書込データ量及び単位読出データ量のうちの最大値が好適であるが、単位書込データ量を設定してもよいし、単位読出データ量（自モジュールの後段に複数の画像処理モジュール38が連結されている場合は、個々の画像処理モジュール38によって各々設定された単位読出データ量の最大値）を設定してもよいし、単位書込データ量と単位読出データ量（の最大値）の最小公倍数を設定してもよいし、この最小公倍数が所定値未満であれば最小公倍数を、最小公倍数が所定値以上であれば別の値（例えば上述した単位書込データ量及び単位読出データ量のうちの最大値、単位書込データ量、単位読出データ量（の最大値）の何れか）を設定するようにしてもよい。

10

【0069】

また、自モジュールが画像データ供給部22又は画像出力部24として機能するバッファモジュール40であった場合には、自モジュールのバッファ40Aとして用いるメモリ領域が既に存在しているので、先に決定した単位バッファ領域のサイズを、自モジュールのバッファ40Aとして用いる既設のメモリ領域のサイズに変更する。更に、自モジュールの後段の個々の画像処理モジュール38に対応する有効データポイントを各々生成し、生成した有効データポイントを初期化する。この有効データポイントは、自モジュールの前段の画像処理モジュールによって自モジュールのバッファ40Aに書き込まれた画像データのうち、対応する後段の画像処理モジュール38によって読み出されていない画像データ（有効データ）の先頭位置（次の読出開始位置）と末尾位置を各々指し示すポイントであり、初期化時には通常、有効データが存在していないことを意味する特定の情報が設定されるが、自モジュールが画像データ供給部22として機能するバッファモジュール40であれば、自モジュールのバッファ40Aとして用いるメモリ領域には既に画像処理対象の画像データが書き込まれていることがあり、この場合は当該画像データの先頭位置及び末尾位置が後段の個々の画像処理モジュール38に対応する有効データポイントに各々設定される。以上の処理によりバッファモジュール40の初期化が完了し、バッファ制御部40Bは初期化の完了を処理管理部46へ通知する。

20

30

【0070】

処理管理部46は、画像処理部50を構成する全てのモジュールから初期化の完了が通知されると、ワークフロー管理部46Aのプログラムを実行するスレッド（又はプロセス又はオブジェクト）を起動し、ワークフロー管理部46Aに対して画像処理部50による画像処理の実行を指示する。ここで、処理管理部ライブラリ47にプログラムが登録されている個々の処理管理部46は、ワークフロー管理部46Aが行う処理が互いに異なっており、稼働中の処理管理部46が並列処理管理部である場合には、起動されたワークフロー管理部46Aにより、例えば図8に示す並列制御処理が行われ、稼働中の処理管理部46が逐次処理管理部である場合には、起動されたワークフロー管理部46Aにより、例えば図10に示すブロック単位逐次制御処理が行われる。これらの処理は、画像処理部50を構成する画像処理モジュール38に処理要求を入力することで、画像処理部50に画像処理を行わせるものであるが、以下では画像処理部50全体の動作説明に先立ち、個々のバッファモジュール40のバッファ制御部40Bによって行われる処理、個々の画像処理モジュール38の制御部38Bによって行われる処理について順に説明する。

40

【0071】

本実施形態では、画像処理モジュール38が後段のバッファモジュール40に画像データを書き込む場合には、画像処理モジュール38からバッファモジュール40へ書込要求が入力され、画像処理モジュール38が前段のバッファモジュール40から画像データを読み出す場合には、画像処理モジュール38からバッファモジュール40へ読出要求が入力される。排他制御機能付きのバッファモジュール40に前段の画像処理モジュール38

50

からの書込要求が入力された場合（及び、後述するタイマがタイムアウトした場合）は、以下で説明するデータ書込処理がバッファ制御部 40B によって実行される。

【0072】

排他制御機能付きのバッファモジュール 40 のバッファ制御部 40B によって行われるデータ書込処理では、まず自モジュールのバッファ 40A が既にアクセス中か否かが判定する。画像処理部 50 の個々の画像処理モジュール 38 が並列に画像処理を行う場合、バッファ 40A に対してデータの書き込みと非同期に読み出しも行われるので、バッファ 40A が既にアクセス中の場合は入力された書込要求情報をワークメモリ等に保管し、タイマをスタートさせてデータ書込処理を一旦終了する。なお、以降の処理では、入力された書込要求情報を処理対象の情報とするが、タイマがタイムアウトしてデータ書込処理が起動された場合には、過去に入力されてワークメモリ等に保管している書込要求情報をワークメモリ等から取り出し、取り出した書込要求情報を処理対象の情報として以降の処理を行う。

10

【0073】

なお、上記のようにバッファ 40A が既にアクセス中か否かを判定し、アクセス中であれば書込要求情報を保管すると共にタイマをスタートさせ、タイマがタイムアウトするとバッファ 40A がアクセス中か否かを再度判定する処理は、本発明に係る「バッファへのアクセスに対する排他制御」に対応している。

【0074】

バッファ 40A がアクセス中でないと判定されると、続いてデータ書込処理では、確保すべきメモリ領域のサイズとして単位書込データ量をリソース管理部 46B に通知し、書込用として用いるメモリ領域（書込用バッファ領域：図 5(B)も参照）を稼働中の処理管理部 46 のリソース管理部 46B を介して取得する。次に、自モジュールのバッファ 40A を構成する保管用の単位バッファ領域の中に、単位書込データ量以上の空き領域が有る単位バッファ領域（単位書込データ量の画像データを書き込み可能な単位バッファ領域）が存在しているか否かが判定する。モジュール生成部 44 によって生成されたバッファモジュール 40 は、当初はバッファ 40A として用いるメモリ領域（単位バッファ領域）が確保されておらず、メモリ領域の不足が生ずる度に単位バッファ領域を単位として確保されるので、バッファモジュール 40 に最初に書込要求が入力されたときにはバッファ 40A として用いるメモリ領域（単位バッファ領域）が存在しておらず、この判定は否定される。また、後述する処理を経てバッファ 40A として用いる単位バッファ領域が確保された後も、当該単位バッファ領域への画像データの書込に伴って当該単位バッファ領域内の空き領域が単位書込データ量未満になった場合にも上記判定は否定される。

20

30

【0075】

単位書込データ量以上の空き領域が有る単位バッファ領域（単位書込データ量の画像データを書き込み可能な単位バッファ領域）が存在していないと判定された場合は、確保すべきメモリ領域のサイズ（単位バッファ領域のサイズ）をリソース管理部 46B に通知して、自モジュールのバッファ 40A として用いるメモリ領域（画像データの保管に用いる単位バッファ領域）をリソース管理部 46B を介して取得する。そして、先に取得した書込用バッファ領域を書込領域として、当該書込領域の先頭アドレスを書込要求元の画像処理モジュール 38 へ通知すると共に、書込対象の画像データを通知した先頭アドレスから順に書き込むよう要請する。これにより、書込要求元の画像処理モジュール 38 は、先頭アドレスが通知された書込用バッファ領域に画像データを書き込む（図 5(B)も参照）。

40

【0076】

例えば単位バッファ領域のサイズが単位書込データ量の整数倍でない場合、バッファ 40A（単位バッファ領域）への単位書込データ量の画像データの書込が繰り返されることで、例として図 5(A)にも示すように、空き領域有りの単位バッファ領域における空き領域のサイズが単位書込データ量よりも小さい状態が生ずる。この場合、単位書込データ量の画像データが書き込まれる領域が複数の単位バッファ領域に跨ることになるが、本実施形態では、バッファ 40A として用いるメモリ領域を単位バッファ領域を単位として確保

50

するので、異なるタイミングで確保した単位バッファ領域が実メモリ（メモリ14）上で連続する領域であることは保証されない。これに対して本実施形態では、画像処理モジュール38による画像データの書き込みを、保管用の単位バッファ領域と別に確保した書込用バッファ領域に対して行わせ、図5(C)に示すように、書込用バッファ領域に一旦書き込まれた画像データを保管用の単一又は複数の単位バッファ領域へ複写するので、画像データが書き込まれる領域が複数の単位バッファ領域に跨るか否かに拘わらず、書込要求元の画像処理モジュール38への書込領域の通知は、上記のようにその先頭アドレスを通知するのみで済み、画像処理モジュール38とのインタフェースが簡単になる。

【0077】

なお、自モジュールがアプリケーション32によって生成されたバッファモジュール40である場合、すなわちバッファ40Aとして用いるメモリ領域が既に確保されている場合には、既に確保されたメモリ領域のアドレスを画像処理モジュール38に書込領域のアドレスとして通知し、上記メモリ領域への画像データの書き込みを行わせる。前段の画像処理モジュール38による書込領域への画像データの書き込みが完了すると、書込用バッファ領域に書き込まれている画像データに属性情報を付加した後に、保管用バッファ領域にそのまま書き込む。なお、空き領域有りの単位バッファ領域における空き領域のサイズが単位書込データ量よりも小さい場合、書込用バッファ領域に書き込まれた画像データは、図5(C)に示すように、保管用の複数の単位バッファ領域へ分けて書き込まれることになる。

【0078】

そして、自モジュールの後段の個々の画像処理モジュール38に対応する有効データポインタのうち有効データの末尾位置を表すポインタを、該ポインタが指し示す有効データの末尾位置が単位書込データ量分だけ後へ移動するように更新する（図5(C)も参照）と共に、先に書込用バッファ領域として確保したメモリ領域をリソース管理部46Bによって解放させ、データ書込処理を一旦終了する。なお、書込用バッファ領域はバッファモジュール40の初期化時に確保し、バッファモジュール40の消去時に解放するように構成してもよい。

【0079】

上記で説明したデータ書込処理は、並列処理用の画像処理部50に組み込まれた排他制御機能付きのバッファモジュール40のバッファ制御部40Bによって行われるデータ書込処理であるが、逐次処理用の画像処理部50に組み込まれた排他制御機能無しのバッファモジュール40のバッファ制御部40Bによって行われるデータ書込処理は、排他制御に相当する処理、すなわちバッファ40Aが既にアクセス中か否かを判定し、アクセス中であれば書込要求情報を保管すると共にタイマをスタートさせ、タイマがタイムアウトするとバッファ40Aがアクセス中か否かを再度判定する処理を行わない点以外は上記で説明したデータ書込処理と同一である。排他制御機能無しのバッファモジュール40におけるデータ書込処理は、逐次処理では不要な排他制御に相当する処理が省略されていることで、処理効率を向上させることができる。

【0080】

続いて、排他制御機能付きのバッファモジュール40に後段の画像処理モジュール38から読出要求が入力された場合（及び、後述するタイマがタイムアウトした場合）に、バッファモジュール40のバッファ制御部40Bによって実行されるデータ読出処理について説明する。

【0081】

排他制御機能付きのバッファモジュール40のバッファ制御部40Bによって行われるデータ読出処理では、まず今回のデータ読出処理の起動要因が、後段の画像処理モジュールからの読出要求の入力による起動か否か判定し、判定が肯定された場合は後段の画像処理モジュールから入力された読出要求情報を読出用の待ち行列の末尾に登録する。次に、自モジュールのバッファ40Aが既にアクセス中か否か判定する。バッファ40Aがアクセス中であれば、読出用の待ち行列に読出要求情報が登録されているか否か判定し、読出

10

20

30

40

50

要求情報が未登録であればそのままデータ読出処理を終了し、読出要求情報が登録されていればタイマをスタートさせた後にデータ読出処理を終了する。このタイマがタイムアウトするとデータ読出処理が再度起動され、読出用の待ち行列に登録されている未処理の読出要求（情報）が再度取り出され、当該読出要求に応じた処理が行われる。

【 0 0 8 2 】

なお、上記のように後段の画像処理モジュールからの読出要求情報を待ち行列の末尾に一旦登録し、バッファ 4 0 A が既にアクセス中か否かを判定し、アクセス中でかつ待ち行列に読出要求情報が登録されている場合はタイマをスタートさせ、タイマがタイムアウトするとバッファ 4 0 A がアクセス中か否かを再度判定する処理も、本発明に係る「バッファへのアクセスに対する排他制御」に対応している。データ読出処理及び前述のデータ書込処理における排他制御により、単一のバッファモジュール 4 0 に複数の要求が同時又は略同時に入力された場合の不都合の発生を回避できるので、コンピュータ 1 0 の CPU 1 2 が並列処理用の画像処理部 5 0 の個々のモジュールに対応するスレッドを並列に実行することが可能となる。

【 0 0 8 3 】

一方、自モジュールのバッファ 4 0 A がアクセス中でなければ、読出用の待ち行列から先頭に登録されている読出要求情報を取り出し、取り出した読出要求情報に含まれる要求元識別情報に基づいて読出要求元の画像処理モジュール 3 8 を認識し、読出要求元の画像処理モジュール 3 8 によって設定された単位読出データ量を認識すると共に、読出要求元の画像処理モジュール 3 8 に対応する有効データポイントに基づいて、読出要求元の画像処理モジュール 3 8 に対応する有効データのバッファ 4 0 A 上での先頭位置及び末尾位置を認識する。次に、認識した有効データの先頭位置及び末尾位置に基づいて、読出要求元の画像処理モジュール 3 8 に対応する有効データ（読出要求元の画像処理モジュール 3 8 が読出可能な画像データ）が単位読出データ量以上有るか否かを判定する。

【 0 0 8 4 】

読出要求元の画像処理モジュール 3 8 に対応する有効データが単位読出データ量未満であれば、読出要求元の画像処理モジュール 3 8 が読出可能な有効データの末尾が処理対象の画像データの末尾か否かを判定する。読出要求元の画像処理モジュール 3 8 に対応する有効データがバッファ 4 0 A に単位読出データ量以上記憶されているか、又は、バッファ 4 0 A に記憶されている読出要求元の画像処理モジュール 3 8 に対応する有効データが単位読出データ量未満であるものの、当該有効データの末尾が処理対象の画像データの末尾であった場合には、確保すべきメモリ領域のサイズとして読出要求元の画像処理モジュール 3 8 に対応する単位読出データ量をリソース管理部 4 6 B に通知すると共に、読出に用いるメモリ領域（読出用バッファ領域：図 6 (B) も参照）の確保をリソース管理部 4 6 B に要求し、リソース管理部 4 6 B を介して読出用バッファ領域を取得する。

【 0 0 8 5 】

次に、読出対象の有効データをバッファ 4 0 A から単位読出データ量分だけ読み出して読出用バッファ領域に書き込み、読出用バッファ領域の先頭アドレスを読出領域の先頭アドレスとして読出要求元の画像処理モジュール 3 8 へ通知すると共に、通知した先頭アドレスから画像データを順に読み出すよう要請する。これにより、読出要求元の画像処理モジュール 3 8 は、先頭アドレスが通知された読出領域（読出用バッファ領域）からの画像データの読み出しを行う。なお、読出対象の有効データが処理対象の画像データの末尾に相当するデータであった場合には、画像データの読出要求に際し、読出対象の画像データのサイズと共に、処理対象の画像データの末尾であることも読出要求元の画像処理モジュール 3 8 に通知する。また、自モジュールがアプリケーション 3 2 によって生成されたバッファモジュール 4 0 である場合は、バッファ 4 0 A として用いているメモリ領域（単位バッファ領域の集合体）は連続領域であるので、読出用バッファ領域の確保、読出対象の画像データの読出用バッファ領域への書き込みを省略し、後段の画像処理モジュール 3 8 が単位バッファ領域から直接画像データを読み出すようにしてもよい。

【 0 0 8 6 】

なお、例として図6(A)に示すように、有効データの先頭部分の画像データを記憶している単位バッファ領域に記憶されている有効データのデータ量が単位読出データ量未満であり、読出対象の有効データが複数の単位バッファ領域に跨っている場合には、今回の読出対象の有効データが実メモリ(メモリ14)上で連続する領域に記憶されているとは限らないが、上記のデータ読出処理では、図6(B),(C)に示すように、このような場合にも読出対象の画像データを読出用バッファ領域に一旦書き込んだ後に該読出用バッファ領域から画像データを読み出させるので、読出対象の画像データが複数の単位バッファ領域に跨って記憶されているか否かに拘わらず、読出要求元の画像処理モジュール38への読出領域の通知は、上記のようにその先頭アドレスを通知するのみで済み、画像処理モジュール38とのインタフェースが簡単になる。

10

【0087】

読出要求元の画像処理モジュール38による読出領域からの画像データの読み出し完了が通知されると、読出用バッファ領域として確保したメモリ領域の先頭アドレス及びサイズをリソース管理部46Bへ通知して、当該メモリ領域をリソース管理部46Bによって解放させる。この読出用バッファ領域についても、バッファモジュール40の初期化時に確保しておき、バッファモジュール40が消去される時に解放するよう構成してもよい。また、読出要求元の画像処理モジュール38に対応する有効データポインタのうち有効データの先頭位置を表すポインタを、該ポインタが指し示す有効データの先頭位置を単位読出データ量分だけ後へ移動させることで更新する(図6(C)も参照)。

【0088】

20

次に、後段の個々の画像処理モジュール38に対応する有効データポインタを各々参照し、先のポインタ更新により、バッファ40Aを構成する単位バッファ領域の中に、記憶している画像データの後段の各画像処理モジュール38による読み出しが全て完了した単位バッファ領域、すなわち有効データを記憶していない単位バッファ領域が出現したか否か判定する。判定が否定された場合は、前述した読出用の待ち行列のチェック処理(読出用の待ち行列に読出要求情報が登録されているか否かの判定)を経てデータ読出処理を終了するが、有効データを記憶していない単位バッファ領域が出現した場合は、当該単位バッファ領域をリソース管理部46Bによって解放させた後に読出用の待ち行列のチェック処理を経てデータ読出処理を終了する。

【0089】

30

一方、バッファ40Aに記憶されており読出要求元の画像処理モジュール38が読出可能な有効データのデータ量が単位読出データ量未満であり、かつ読出可能な有効データの末尾が処理対象の画像データの末尾でない場合(図4(B)の(4)で読出可能な有効データ無が検知された場合)には、新たな画像データを要求するデータ要求をワークフロー管理部46Aへ出力し(図4(B)の(5)も参照)、読出用の待ち行列から取り出した読出要求情報を元の待ち行列(の先頭又は末尾)に再度登録した後に、読出用の待ち行列のチェック処理を経てデータ読出処理を終了する。この場合、ワークフロー管理部46Aにより、自モジュールの前段の画像処理モジュール38に処理要求が入力されることになる。これにより、読出可能な有効データのデータ量が単位読出データ量以上になるか、読出可能な有効データの末尾が処理対象の画像データの末尾であることが検知される迄の間、対応する読出要求情報は読出用の待ち行列に保存されると共に定期的に取り出されて要求された処理の実行が繰り返し試行されることになる。

40

【0090】

詳細は後述するが、ワークフロー管理部46Aはバッファモジュール40からデータ要求が入力されると、データ要求元のバッファモジュール40の前段の画像処理モジュール38に処理要求を入力する(図4(B)の(6)も参照)。この処理要求の入力をトリガとして前段の画像処理モジュール38の制御部38Bで行われる処理により、前段の画像処理モジュール38がバッファモジュール40へ画像データを書込可能な状態になると、前段の画像処理モジュール38から書込要求が入力されることで前述したデータ書込処理が行われ、前段の画像処理モジュール38からバッファモジュール40のバッファ40Aに画

50

像データが書き込まれる(図4(B)の(7),(8)も参照)。これにより、後段の画像処理モジュール38によるバッファ40Aからの画像データの読出が行われることになる(図4(B)の(9)も参照)。

【0091】

なお、上記で説明したデータ読出処理は、並列処理用の画像処理部50に組み込まれた排他制御機能付きのバッファモジュール40のバッファ制御部40Bによって行われるデータ読出処理であるが、逐次処理用の画像処理部50に組み込まれた排他制御機能無しのバッファモジュール40のバッファ制御部40Bによって行われるデータ読出処理は、排他制御に相当する処理、すなわちバッファ40Aが既にアクセス中か否かを判定し、アクセス中であつ待ち行列に読出要求情報が登録されている場合はタイマをスタートさせ、タイマがタイムアウトするとバッファ40Aがアクセス中か否かを再度判定すると共に、単一の読出要求に対する処理が終了した後に待ち行列に読出要求情報が残っているかをチェックする処理を行わない点以外は上記で説明したデータ読出処理と同一である。排他制御機能無しのバッファモジュール40におけるデータ読出処理は、逐次処理では不要な排他制御に相当する処理が省略されていることで、処理効率を向上させることができる。

10

【0092】

続いて、画像処理部50を構成する個々の画像処理モジュール38に対してワークフロー管理部46Aから処理要求が入力される毎に、個々の画像処理モジュール38の制御部38Bによって各々行われる画像処理モジュール制御処理(図7)を説明する。なお、画像処理モジュール38の構成については、画像処理部50が並列処理用か逐次処理用かに拘わらず同一であり、以下では画像処理部50が並列処理用か逐次処理用かを区別することなく、画像処理モジュール制御処理を説明する。

20

【0093】

画像処理モジュール制御処理では、まずステップ219において、自モジュールの画像処理エンジン38Aが行う画像処理の種類や内容等に基づき、自モジュールが使用するメモリのサイズ及び自モジュールが使用する他のリソースの有無を認識する。なお、画像処理モジュール38が使用するメモリは、画像処理エンジン38Aが画像処理を行うために必要なメモリが主であるが、前段のモジュールが画像データ供給部22である場合や後段のモジュールが画像出力部24である場合には、前段又は後段のモジュールとの画像データの送受に際して画像データを一時記憶するためのバッファ用のメモリが必要となることもある。また、処理パラメータにテーブル等の情報が含まれている場合には、それを保持するためのメモリ領域が必要となることもある。そして、認識したサイズのメモリ領域の確保をリソース管理部46Bへ要求し、リソース管理部46Bによって確保されたメモリ領域をリソース管理部46Bから取得する。また、自モジュール(の画像処理エンジン38A)がメモリ以外の他のリソースを必要としていると認識した場合には、上記他のリソースの確保をリソース管理部46Bへ要求し、上記他のリソースをリソース管理部46Bから取得する。

30

【0094】

次のステップ220では、自モジュールの前段にモジュール(バッファモジュール40や画像データ供給部22、画像処理モジュール38等)が存在している場合に、当該前段のモジュールに対してデータ(画像データ又は解析等の画像処理の処理結果)を要求する。次のステップ222では前段のモジュールからデータが取得可能であるかを判定し、ステップ222の判定が否定された場合はステップ224で全体処理終了が通知されたか否かを判定する。ステップ224の判定が否定された場合はステップ222に戻り、前段のモジュールからデータを取得可能となる迄ステップ222, 224を繰り返す。ステップ222の判定が肯定された場合には、ステップ226で前段のモジュールからデータを取得し、取得したデータをステップ219で取得したメモリ領域のうちデータの一時保管用のメモリ領域に書き込むデータ取得処理を行う。

40

【0095】

ここで、自モジュールの前段のモジュールがバッファモジュール40である場合には、

50

先のステップ 220 でデータを要求すると（読出要求）、読出可能な有効データがバッファモジュール 40 のバッファ 40A に単位読出データ量以上記憶されているか、読出可能な有効データの末尾が処理対象の画像データの末尾に一致している状態であれば直ちに、当該状態でなければ、当該バッファモジュール 40 の前段の画像処理モジュール 38 が当該バッファモジュール 40 のバッファ 40A に画像データを書き込んだことに伴って前記状態へ変化した後に、バッファモジュール 40 から読出領域の先頭アドレスが通知されて画像データの読出が要請される。これにより、ステップ 222 の判定が肯定されてステップ 226 へ移行し、前段のバッファモジュール 40 より先頭アドレスが通知された読出領域から単位読出データ量（又はそれ未満のデータ量）の画像データを読み出し、一時保管用のメモリ領域に書き込むデータ取得処理を行う（図 3(A)の(3)も参照）。

10

【0096】

また、自モジュールの前段のモジュールが画像データ供給部 22 であれば、先のステップ 220 でデータ要求を出力すると画像データを取得可能な状態であることが前段の画像データ供給部 22 から直ちに通知されることで、ステップ 222 の判定が肯定されてステップ 226 へ移行し、前段の画像データ供給部 22 から単位読出データ量の画像データを取得し、一時保管用のメモリ領域に書き込む画像データ取得処理を行う。また、自モジュールの前段のモジュールが画像処理モジュール 38 であれば、先のステップ 220 でデータ要求（処理要求）を出力すると、前段の画像処理モジュール 38 が画像処理を実行可能な状態であれば書込要求が入力されることでデータ（画像処理結果）を取得可能な状態であることが通知されるので、ステップ 222 の判定が肯定されてステップ 226 へ移行し、前段の画像処理モジュール 38 によってデータを書き込ませる一時保管用のメモリ領域のアドレスを通知して書込を要請することで、前段の画像処理モジュール 38 から出力されるデータを一時保管用のメモリ領域に書き込ませるデータ取得処理を行う。

20

【0097】

次のステップ 228 では、自モジュールの前段に複数のモジュールが連結されているか否か判定する。判定が否定された場合には何ら処理を行うことなくステップ 232 へ移行するが、判定が肯定された場合はステップ 230 へ移行し、前段に連結されている全てのモジュールからデータを取得したか否か判定する。ステップ 230 の判定が否定された場合はステップ 220 に戻り、ステップ 230 の判定が肯定される迄ステップ 220 ~ ステップ 230 を繰り返す。前段のモジュールから取得すべきデータが全て揃うと、ステップ 228 の判定が否定されるかステップ 230 の判定が肯定されてステップ 232 へ移行する。

30

【0098】

次のステップ 232 では自モジュールの後段のモジュールに対してデータ出力用の領域を要求し、ステップ 232 でデータ出力領域が取得できる迄（データ出力領域の先頭アドレスが通知される迄）繰り返し判定を行う。なお、後段のモジュールがバッファモジュール 40 であれば、上記のデータ出力用領域の要求は当該バッファモジュール 40 に対して書込要求を出力することによって成される。データ出力領域（後段のモジュールがバッファモジュール 40 であれば当該バッファモジュール 40 から先頭アドレスが通知された書込領域）が取得できたら（図 3(A)の(4)も参照）、次のステップ 236 において、先のデータ取得処理で取得したデータ、後段のモジュールから取得したデータ出力領域（の先頭アドレス）、先のステップ 219 で取得したメモリ領域のうち画像処理エンジンによる画像処理用のメモリ領域（の先頭アドレス及びサイズ）を画像処理エンジン 38A に入力し、入力したデータに対し画像処理用のメモリ領域を使用して所定の画像処理を行わせる（図 3(A)の(5)も参照）と共に、処理後のデータをデータ出力領域に書き込ませる（図 3(A)の(6)も参照）。画像処理エンジン 38A への単位読出データ量のデータの入力が完了し、画像処理エンジン 38A から出力されたデータがデータ出力領域に全て書き込まれると、次のステップ 238 で出力が完了したことを後段のモジュールに通知する。

40

【0099】

上記のステップ 220 ~ ステップ 238 により画像処理モジュール 38 における単位処

50

理データ量のデータに対する処理（単位処理）が完了するが、ワークフロー管理部 4 6 A から画像処理モジュール 3 8 に入力される処理要求では、ワークフロー管理部 4 6 A によって単位処理の実行回数が指定されることがある。このためステップ 2 4 0 では、単位処理の実行回数が、入力された処理要求によって指示された実行回数に達したか否かが判定する。指示された単位処理の実行回数が 1 回の場合、この判定は無条件に肯定されるが、指示された単位処理の実行回数が 2 回以上の場合はステップ 2 2 0 に戻り、ステップ 2 4 0 の判定が肯定される迄ステップ 2 2 0 ~ ステップ 2 4 0 を繰り返す。ステップ 2 4 0 の判定が肯定されるとステップ 2 4 2 へ移行し、ワークフロー管理部 4 6 A へ処理完了通知を出力することで、入力された処理要求に対応する処理が完了したことをワークフロー管理部 4 6 A へ通知し、画像処理モジュール制御処理を終了する。

10

【 0 1 0 0 】

また、ワークフロー管理部 4 6 A から処理要求が入力される毎に上述した処理が繰り返されることで処理対象の画像データを末尾まで処理すると、前段のモジュールから処理対象の画像データの終了が通知されることで、ステップ 2 2 4 の判定が肯定されてステップ 2 4 4 へ移行し、処理対象の画像データ（なお、処理対象の画像データは 1 頁分の画像データであることが多いが、複数頁分の画像データであってもよい）に対する処理が終了したことを意味する全体処理終了通知をワークフロー管理部 4 6 A 及び後段のモジュールへ各々出力する。また、次のステップ 2 4 6 では取得していた全てのリソースの解放を要求して自モジュールを消去する処理を行い、画像処理モジュール制御処理を終了する。

【 0 1 0 1 】

20

一方、稼働中の処理管理部 4 6 が並列処理管理部である場合、ワークフロー管理部 4 6 A は、画像処理の実行が指示されると、図 8 (A) に示す並列制御処理 1 を行う。先にも述べたように、ワークフロー管理部 4 6 A による画像処理部 5 0 の個々の画像処理モジュール 3 8 への処理要求の入力では、単位処理の実行回数を指定可能とされているが、並列制御処理 1 のステップ 5 0 0 では、1 回の処理要求で指定する単位処理の実行回数を個々の画像処理モジュール 3 8 毎に決定する。この処理要求 1 回当たりの単位処理の実行回数は、例えば処理対象の画像データ全体を処理する間の個々の画像処理モジュール 3 8 への処理要求の入力回数が平均化されるように定めることができるが、他の基準に従って定めてもよい。そして次のステップ 5 0 4 において、画像処理部 5 0 のうち最後段の画像処理モジュール 3 8 に処理要求を入力し（図 9 の(1)も参照）、並列制御処理 1 を終了する。

30

【 0 1 0 2 】

ここで、図 9 に示す画像処理部 5 0 において、ワークフロー管理部 4 6 A から最後段の画像処理モジュール 3 8₄ に処理要求が入力されると、画像処理モジュール 3 8₄ の制御部 3 8 B は前段のバッファモジュール 4 0₃ に読出要求を入力する（図 9 の(2)参照）。このとき、バッファモジュール 4 0₃ のバッファ 4 0 A には画像処理モジュール 3 8₄ が読出可能な有効データ（画像データ）が記憶されていないので、バッファモジュール 4 0₃ のバッファ制御部 4 0 B はワークフロー管理部 4 6 A にデータ要求を入力する（図 9 の(3)参照）。

【 0 1 0 3 】

並列処理管理部のワークフロー管理部 4 6 A は、バッファモジュール 4 0 からデータ要求が入力される毎に、図 9 (B) に示す並列制御処理 2 を行う。この並列制御処理 2 では、ステップ 5 1 0 において、データ要求入力元のバッファモジュール 4 0（ここではバッファモジュール 4 0₃）の前段の画像処理モジュール 3 8（ここでは画像処理モジュール 3 8₃）を認識し、認識した前段の画像処理モジュール 3 8 に処理要求を入力（図 9 の(4)参照）して処理を終了する。

40

【 0 1 0 4 】

画像処理モジュール 3 8₃ の制御部 3 8 B は、処理要求が入力されると前段のバッファモジュール 4 0₂ に読出要求を入力し（図 9 の(5)参照）、バッファモジュール 4 0₂ のバッファ 4 0 A にも読出可能な画像データが記憶されていないので、バッファモジュール 4 0₂ のバッファ制御部 4 0 B はワークフロー管理部 4 6 A にデータ要求を入力する（図 9

50

の(6)参照)。ワークフロー管理部46Aは、バッファモジュール40₂からデータ要求が入力された場合も、前述の並列制御処理2を再度行うことで、その前段の画像処理モジュール38₂に処理要求を入力し(図9の(7)参照)、画像処理モジュール38₃の制御部38Bは前段のバッファモジュール40₁に読出要求を入力する(図9の(8)参照)。また、バッファモジュール40₁のバッファ40Aにも読出可能な画像データが記憶されていないので、バッファモジュール40₁のバッファ制御部40Bもワークフロー管理部46Aにデータ要求を入力し(図9の(9)参照)。ワークフロー管理部46Aは、バッファモジュール40₁からデータ要求が入力された場合も、前述の並列制御処理2を再度行うことで、その前段の画像処理モジュール38₁に処理要求を入力する(図9の(10)参照)。

【0105】

ここで、画像処理モジュール38₁の前段のモジュールは画像データ供給部22であるので、画像処理モジュール38₁の制御部38Bは、画像データ供給部22にデータ要求を入力することで画像データ供給部22から単位読出データ量の画像データを取得し(図9の(11)参照)、取得した画像データに対して画像処理エンジン38Aが画像処理を行うことで得られた画像データを、後段のバッファモジュール40₁のバッファ40Aに書き込む(図9の(12)参照)。

【0106】

また、バッファモジュール40₁のバッファ制御部40Bは、後段の画像処理モジュール38₂が読出可能な単位読出データ量以上の有効データが書き込まれると画像処理モジュール38₂に対して読出を要請し、これに伴い画像処理モジュール38₂の制御部38Bは、バッファモジュール40₁のバッファ40Aから単位読出データ量の画像データを読み出し(図9の(13)参照)、取得した画像データに対して画像処理エンジン38Aが画像処理を行うことで得られた画像データを、後段のバッファモジュール40₂のバッファ40Aに書き込む(図9の(14)参照)。バッファモジュール40₂のバッファ制御部40Bは、後段の画像処理モジュール38₃が読出可能な単位読出データ量以上の有効データが書き込まれると画像処理モジュール38₃へ読出を要請し、画像処理モジュール38₃の制御部38Bは、バッファモジュール40₂のバッファ40Aから単位読出データ量の画像データを読み出し(図9の(15)参照)、取得した画像データに対して画像処理エンジン38Aが画像処理を行うことで得られた画像データを、後段のバッファモジュール40₃のバッファ40Aに書き込む(図9の(16)参照)。

【0107】

更に、バッファモジュール40₃のバッファ制御部40Bは、後段の画像処理モジュール38₄が読出可能な単位読出データ量以上の有効データが書き込まれると画像処理モジュール38₄に対して読出を要請し、これに伴い画像処理モジュール38₄の制御部38Bは、バッファモジュール40₃のバッファ40Aから単位読出データ量の画像データを読み出し(図9の(17)参照)、取得した画像データに対して画像処理エンジン38Aが画像処理を行うことで得られた画像データを、後段のモジュールである画像出力部24へ出力する(図9の(18)参照)。

【0108】

また、個々の画像処理モジュール38の制御部38Bは、後段のバッファモジュール40のバッファ40Aへの画像データの書き込みを完了すると、ワークフロー管理部46Aへ処理完了通知を入力する。並列処理管理部のワークフロー管理部46Aは、画像処理モジュール38から処理完了通知が入力される毎に、図8(C)に示す並列制御処理3を行う。この並列制御処理3では、ステップ520で処理完了通知元の画像処理モジュール38に処理要求を再度入力して処理を終了する。

【0109】

このように、並列処理管理部のワークフロー管理部46Aによる並列制御処理では、任意の画像処理モジュール38から処理完了が通知される毎に、処理完了通知元の画像処理モジュール38へ処理要求を再度入力することで、処理対象の画像データが前段側のモジュールから後段側のモジュールへ画像1面分よりも小さいサイズ(ブロック)を単位とし

10

20

30

40

50

て順に引き渡されると共に、個々の画像処理モジュール38が互いに並列に画像処理を行う並列処理方式によって処理対象の画像データに対する画像処理が行われることになる。また、画像データ供給部22から供給される画像データが処理対象の画像データの末尾に達すると、個々の画像処理モジュール38からワークフロー管理部46Aへの全体処理終了通知の入力が、前段側の画像処理モジュール38から順次行われる。

【0110】

並列処理管理部のワークフロー管理部46Aは、画像処理モジュール38から全体処理終了通知が入力される毎に、図8(D)に示す並列制御処理4を行う。この並列制御処理4では、ステップ540において、全体処理終了通知入力元の画像処理モジュール38が最後段の画像処理モジュール38か否か判定する。判定が否定された場合は何ら処理を行うことなく処理を終了するが、処理対象の画像データに対して必要な画像処理が行われた画像データが画像出力部24へ全て出力されることで、最後段の画像処理モジュール38から全体処理終了通知が入力された場合には、ステップ540の判定が肯定されてステップ542へ移行し、アプリケーション32に対して画像処理の完了を通知し(図2のステップ178も参照)、並列制御処理4を終了する。そして、画像処理の完了が通知されたアプリケーション32は、ユーザに対して画像処理の完了を通知する(図2のステップ180も参照)。なお、上述した並列制御処理1~4を行う並列処理管理部は、詳しくは請求項4に記載の第1の並列処理制御手段に対応している。

10

【0111】

次に、稼働中の処理管理部46が逐次処理管理部である場合に、ワークフロー管理部46Aによって行われる処理を説明する。逐次処理管理部のワークフロー管理部46Aは、画像処理の実行が指示されると図10(A)に示すブロック単位逐次制御処理1を行い、バッファモジュール40からデータ要求が入力される毎に図10(B)に示すブロック単位逐次制御処理2を行い、画像処理モジュール38から処理完了通知が入力される毎に図10(C)に示すブロック単位逐次制御処理3を行い、画像処理モジュール38から全体処理終了通知が入力される毎に図10(D)に示すブロック単位逐次制御処理4を行うが、このうちブロック単位逐次制御処理1, 2, 4については、上記で説明した並列制御処理1, 2, 4と同一であるので説明を省略し、画像処理モジュール38から処理完了通知が入力される毎に行われるブロック単位逐次制御処理3について説明する。

20

【0112】

このブロック単位逐次制御処理3では、まずステップ518において、処理完了通知元の画像処理モジュール38が最後段の画像処理モジュール38か否か判定する。判定が否定された場合は何ら処理を行うことなくブロック単位逐次制御処理3を終了する。また、判定が肯定された場合はステップ520へ移行し、処理完了通知元の画像処理モジュール38に処理要求を再度入力して処理を終了する。

30

【0113】

従って、逐次処理管理部のワークフロー管理部46Aによるブロック単位逐次制御処理では、画像処理部50の最後段の画像処理モジュール38に入力された処理要求がより前段の画像処理モジュール38へ遡り、最前段の画像処理モジュール38に到達すると、画像1面分よりも小さいサイズ(ブロック)のデータに対する画像処理が最前段の画像処理モジュール38から逐次行われ(常に何れか1つの画像処理モジュール38でのみ画像処理が行われると共に、画像処理を行っている画像処理モジュール38が逐次切り替わり)、上記データに対する最後段の画像処理モジュール38による画像処理が終了すると、最後段の画像処理モジュール38に処理要求が再度入力されることが繰り返される逐次処理方式によって、処理対象の画像データに対する画像処理が行われることになる。なお、上述したブロック単位逐次制御処理1~4を行う逐次処理管理部は、詳しくは請求項5に記載の第1の逐次処理制御手段に対応している。

40

【0114】

また、上記のように画像処理部50で並列処理方式又は逐次処理方式で画像処理が行われるようにワークフロー管理部46Aが制御を行っている間、処理管理部46のエラー管

50

理部 46C も動作している。エラー管理部 46C は、画像処理部 50 の任意の画像処理モジュール 38 でエラーが発生した場合、エラーが発生した画像処理モジュール 38 からエラー発生通知が入力される。エラー管理部 46C は、エラー発生通知が入力されると、発生したエラーの種別等のエラー情報を取得し、画像処理プログラム群 34 がインストールされたコンピュータ 10 が組み込まれている機器の種別や構成等を表す装置環境情報を記憶部 20 等から取得し、取得した装置環境情報が表す装置環境に応じたエラー通知方法を判断し、判断したエラー通知方法でエラーの発生を通知するエラー通知処理を行う。

【0115】

ここで、画像処理部 50 が逐次処理方式で画像処理を行っている場合には、常に何れか 1 つの画像処理モジュール 38 でのみ画像処理が行われるので、複数の画像処理モジュール 38 からエラー管理部 46C へ同時又はほぼ同時にエラー発生通知が入力されることはなく、また、画像処理部 50 を構成する個々のモジュールのプログラムが単一のスレッドとして実行されているので、何れかの画像処理モジュール 38 で重度のエラーが発生して画像処理を停止すると、これに伴って画像処理部 50 で行っている画像処理全体も停止することになるが、画像処理部 50 が並列処理方式で画像処理を行っている場合には、個々の画像処理モジュール 38 で並列に画像処理が行われるので、複数の画像処理モジュール 38 からエラー管理部 46C へ同時又はほぼ同時にエラー発生通知が入力される可能性があり、また、画像処理部 50 を構成する個々のモジュールのプログラムが互いに独立したスレッドとして実行されているので、何れかの画像処理モジュール 38 で重度のエラーが発生して画像処理を停止しても、画像処理部 50 の他の画像処理モジュール 38 では画像処理が継続されることになる。

【0116】

このため、並列処理管理部 46 のエラー管理部 46C は、複数の画像処理モジュール 38 から同時又はほぼ同時にエラー発生通知が入力された場合に不都合が生じないように排他制御を行うと共に、何れかの画像処理モジュール 38 で重度のエラーが発生して画像処理が停止した場合に、画像処理部 50 の他の画像処理モジュール 38 へエラーの発生を通知して画像処理の実行を中止させ、画像処理部 50 の個々のモジュールに相当する全てのスレッドの実行を停止させる処理を行う。これにより、画像処理部 50 が並列処理方式で画像処理を行っている場合にも、エラー処理を支障なく行うことができる。

【0117】

このように、並列処理管理部 46 には並列処理方式に適したエラー処理を行うエラー管理部 46C が設けられていると共に、逐次処理管理部 46 には逐次処理方式に適したエラー処理を行うエラー管理部 46C が設けられており、選択起動部 45 が並列処理管理部 46 と逐次処理管理部 46 を選択的に起動することに伴い、動作するエラー管理部 46C が自動的に切り替わるので、画像処理部 50 を並列処理方式で動作させるか逐次処理方式で動作させるかを切り替える際の処理が簡単になる。

【0118】

なお、画像処理部 50 を並列処理方式で動作させる際のワークフロー管理部 46A の処理は、図 8 に示した並列制御処理 1 ~ 4 に限られるものではない。画像処理部 50 を並列処理方式で動作させる場合であっても、個々の画像処理モジュール 38 における画像処理の進行は、より前段に位置している画像処理モジュール 38 における画像処理の進行状況に依存し、特に画像処理部 50 における一連の画像処理の実行開始時やその付近の期間には、各画像処理モジュール 38 のうち、パイプライン形態又は有向非循環グラフ形態において前段側に位置している画像処理モジュールにおける画像処理を優先的に実行した方が処理効率が向上する。

【0119】

また、前段側の画像処理モジュール 38 よりも後段側の画像処理モジュール 38 の方が画像処理の進行が常に後になり、処理対象の画像データの残量も後段側の画像処理モジュール 38 の方が常に多くなるので、画像処理部における一連の画像処理の進行に伴って、後段側に位置している画像処理モジュールにおける画像処理の実行優先度を高くしていっ

10

20

30

40

50

た方が処理効率が向上し、特に画像処理部における一連の画像処理の実行終了時やその付近の期間には、全体処理が終了した画像処理モジュール38が前段側から徐々に増えてくることに伴い、後段側に位置している画像処理モジュールにおける画像処理の実行優先度をより高くすることが処理効率の点から望ましい。上記に基づき、画像処理部50を並列処理方式で動作させる際に、例として図11に示すように、ワークフロー管理部46Aが画像処理部50の個々の画像処理モジュール38に対応するスレッドの実行優先度を設定及び変更するようにしてもよい。

【0120】

すなわち、図11に示す並列制御処理では、ステップ500で1回の処理要求で指定する単位処理の実行回数を個々の画像処理モジュール38毎に決定した後に、次のステップ502において、個々の画像処理モジュール38に対応する個々のスレッドの実行優先度が、例として図12(A)に示すように、パイプライン形態又は有向非循環グラフ形態の連結形態における画像処理モジュール38の位置が前段側になるに従って高くなるように、前記個々のスレッドの実行優先度の初期設定を行う。

【0121】

なお、上記の「画像処理モジュール38の位置」は、例えば画像処理部がパイプライン形態であれば、図13(A)に示すように先頭(最前段)の画像処理モジュール38から昇順に付した位置値(或いは最後尾(最後段)の画像処理モジュール38から降順に付した位置値)に基づいて判断することができ、画像処理部が有向非循環グラフ形態であれば、図13(B)に示すように先頭(最前段)の画像処理モジュール38から昇順に(或いは最後尾(最後段)の画像処理モジュール38から降順に)位置値を付すと共に、バッファモジュールを介して複数の画像処理モジュールから画像データを取得する画像処理モジュール38(図13(B)の例では画像処理モジュールE)については、前段の複数の画像処理モジュールに付した位置値の最大値(又は最小値)に基づいて位置値を付し、この位置値に基づいて判断することができる。

【0122】

また、パイプライン形態又は有向非循環グラフ形態の連結形態における画像処理モジュール38の位置が前段側になるに従って、対応するスレッドの実行優先度を高くすることは、例えば画像処理モジュールに対応するスレッドに設定可能な実行優先度が1~9の9段階であり、個々の画像処理モジュール38に対して位置値を前段側から初期値=1で昇順に付したとすると、個々の画像処理モジュール38に対応するスレッドに対し、

実行優先度 = 10 - (位置値) 但し、実行優先度 < 1 の場合は実行優先度 = 1 に設定してもよいし、位置値が最小値のときには実行優先度が「9」となり、位置値が最大値のときには実行優先度が「1」となるような特定の単調減少関数(例えば位置値の増大に対して実行優先度が線形に減少する関数)を用いて実行優先度を設定するようにしてもよい。これにより、画像処理部で一連の画像処理が開始される時点で、パイプライン形態又は有向非循環グラフ形態の連結形態における対応する画像処理モジュール38の位置が前段側のスレッド程、高い実行優先度でCPU12によって実行されることになり、CPU12を有効に利用して高い処理効率で画像処理を行うことができる。

【0123】

なお、バッファモジュール40に対応するスレッドの実行優先度については、一定の優先度を固定的に設定するようにしてもよいし、個々のバッファモジュール40の前段及び後段に連結された画像処理モジュール38に対応するスレッドの実行優先度の平均値や、最大値、最小値を設定し、次に述べるように、画像処理の進行度合に応じて個々の画像処理モジュール38に対応するスレッドの実行優先度を変更する際に、個々のバッファモジュール40に対応するスレッドの実行優先度も変更するようにしてもよい。

【0124】

また、図11に示す並列制御処理では、画像処理モジュール38から処理完了通知が入力される毎に実行する並列制御処理3(図11(C)参照)において、ステップ522で画像処理部50全体としての画像処理の進行度合を判定する。この判定は、例えば個々の画

10

20

30

40

50

像処理モジュール38からワークフロー管理部46Aへ処理完了通知が送信される際に、個々の画像処理モジュール38における画像処理の進行度合を判断可能な進行度合情報も併せて送信されるように個々の画像処理モジュール38を構成すると共に、ワークフロー管理部46Aは、個々の画像処理モジュール38から処理完了通知を受信する毎に、同時に受信した進行度合情報を保持し（同一の画像処理モジュール38から以前に受信した進行度合情報を既に保持している場合は、既に保持している進行度合情報を新たに受信した進行度合情報で上書きし）た後に、個々の画像処理モジュール38に対応する進行度合情報から画像処理部50全体としての画像処理の進行度合を集計することによって行うことができる。

【0125】

上記の進行度合情報は、導出にあたって画像処理モジュール38（に対応するスレッドを実行するCPU12）に加わる負荷がなるべく小さい情報であることが好ましく、例えば処理対象の画像データ全体に対する個々の画像処理モジュール38の処理済み画像データの割合（詳しくはデータ量の割合やライン数の割合等）を表す情報を用いることができる。また、個々の画像処理モジュール38からは進行度合情報として処理済み画像データのデータ量やライン数を表す情報を送信させ、個々の画像処理モジュール38における画像処理の進行度合（上記の割合等）はワークフロー管理部46Aで演算するようにしてもよい。

【0126】

次のステップ524では、ステップ522で判定した画像処理部50全体としての画像処理の進行度合が、個々の画像処理モジュール38に対応するスレッドの実行優先度を変更すべき値になったか否かが判定する。なお、スレッドの実行優先度の変更は頻繁に行う必要はなく、実行優先度の変更は頻繁に行うことでCPU12に余分な負荷が加わることを回避するために、ステップ524の判定における判定条件としては、例えばスレッドの実行優先度の変更（又は初期設定）を前回行ってから画像処理の進行度合が10%増加する毎に上記判定が肯定される等のように、余分な負荷にならない程度に疎な間隔でスレッドの実行優先度の変更される判定条件を用いればよい。

【0127】

上記判定が否定された場合は何ら処理を行うことなく並列制御処理3を終了するが、上記判定が肯定された場合は、ステップ526において、初期設定時に各スレッドに設定した実行優先度の中央値（又は平均値）を基準として、初期設定時に高い実行優先度を設定したスレッドについては画像処理の進行に伴って実行優先度が徐々に低下し、初期設定時に低い実行優先度を設定したスレッドについては画像処理の進行に伴って実行優先度が徐々に増大するように、個々の画像処理モジュール38に対応するスレッドの実行優先度を変更設定した後に並列制御処理3を終了する。

【0128】

なお、このステップ528における実行優先度の変更は、画像処理モジュール38の位置が最前段又は最後段に近づくに従って対応するスレッドの実行優先度の変更量を多くし、例として図12(B),(C)に示すように、画像処理部50全体としての画像処理の終盤には前段側の画像処理モジュール38に対応するスレッドの実行優先度と後段側の画像処理モジュール38に対応するスレッドの実行優先度の大小関係が逆転するように行ってもよいし、例として図12(D),(E)に示すように、画像処理部50全体としての画像処理の終盤には各画像処理モジュール38に対応するスレッドの実行優先度が一定となるように行ってもよい。画像処理部50における一連の画像処理の進行に伴って、個々の画像処理モジュール38に対応するスレッドの実行優先度を上記のように変更することで、CPU12を有効に利用して高い処理効率で画像処理を行うことができる。なお、個々の画像処理モジュール38に対応するスレッドの実行優先度を上記のように初期設定及び変更する並列処理管理部46は、請求項4記載の第2の並列処理制御手段に対応している。

【0129】

また、個々の画像処理モジュール38に対応する個々のスレッドの実行優先度は、上記

10

20

30

40

50

のように初期設定及び変更することに限られるものではなく、例えば個々の画像処理モジュール38に対応するスレッドの実行優先度を、個々の画像処理モジュールについて、後段のバッファモジュール40を介して後段に連結された画像処理モジュール38（位置値 = 自モジュールの位置値 + 1の画像処理モジュール38）から前記後段のバッファモジュール40に読出要求が入力されたものの、前記後段のバッファモジュール40に記憶されている有効データが単位読出データ量未満であったために、前記後段のバッファモジュール40からデータ要求が入力されると共に、前記後段の画像処理モジュールに「待ち」（バッファモジュール40の有効データが単位読出データ量以上となる迄待機する状況）を発生させた回数（待ち発生回数）に応じて変更するようにしてもよい。

【0130】

具体的には、ワークフロー管理部46Aは、個々の画像処理モジュール38について待ち発生回数（初期値 = 0）を予め保持しており、任意のバッファモジュール40からデータ要求が入力される毎に、データ要求入力元のバッファモジュール40の前段の画像処理モジュール38について待ち発生回数を1だけインクリメントする。そしてワークフロー管理部46Aは、一定の時間周期で、個々の画像処理モジュール38について各々保持している待ち発生回数の平均値を演算し、演算した待ち発生回数の平均値と、個々の画像処理モジュール38の待ち発生回数の偏差に応じて、各画像処理モジュール38に対応するスレッドの実行優先度を変更する。実行優先度の変更は、待ち発生回数が平均値よりも多い画像処理モジュール38に対しては、対応するスレッドの実行優先度が上記偏差が大きくなるに従って増大し、待ち発生回数が平均値よりも少ない画像処理モジュール38に対しては、対応するスレッドの実行優先度が上記偏差が大きくなるに従って低下するように行うことができ、具体的には、例えば以下の式に従って行うことができる。

【0131】

実行優先度の変更率(%) = (待ち発生回数 - 待ち発生回数の平均値) / 平均値 × 100

変更後の実行優先度 = 実行優先度 + (実行優先度 × 変更率) / 100

なお、上記演算に際し、待ち発生回数の平均値に代えて待ち発生回数の中央値を用いてもよい。

【0132】

待ち発生回数が平均値よりも多い画像処理モジュール38は、後段のバッファモジュール40を介して後段に連結された画像処理モジュール38に比較的多数回の「待ち」を発生させており、当該画像処理モジュール38における画像処理が画像処理部全体としての画像処理のボトルネックになっていると判断できるが、上記処理ではこのような画像処理モジュール38に対応するスレッドの実行優先度が増大される。また、待ち発生回数が平均値よりも少ない画像処理モジュール38は、後段のバッファモジュール40を介して後段に連結された画像処理モジュール38における「待ち」の発生回数が比較的少ないので、当該画像処理モジュール38よりも待ち発生回数が比較的多い他の画像処理モジュール38における画像処理を優先させた方が画像処理部全体としての画像処理を効率化できるが、上記処理ではこのような画像処理モジュール38に対応するスレッドの実行優先度が低下される。

【0133】

これにより、個々の画像処理モジュール38に対応するスレッドの実行優先度を、後段の画像処理モジュール38における待ち発生回数（と待ち発生回数の平均値との偏差）に応じて最適化することができ、CPU12を有効に利用して高い処理効率で画像処理を行うことができる。また、メモリ取得待ち行列に登録される取得要求情報が、上記のようにして設定された実行優先度に従って配列されることで、要求されたメモリ領域を確保できずに複数のスレッドの実行を停止させた後に、確保可能なメモリ領域が増大したことでスレッドの実行停止状態を解除してメモリ領域を割り当てる場合にも、画像処理部50における以降の処理を高い処理効率で行わせることができる。

【0134】

なお、上記態様において、後段のバッファモジュール40からデータ要求が入力された

10

20

30

40

50

回数、すなわち後段のバッファモジュール40を介して後段に連結された画像処理モジュール38で「待ち」が発生した回数に、個々の画像処理モジュール38が後段のバッファモジュール40へ画像データを書き込んだものの後段のバッファモジュール40の有効データが、後段の画像処理モジュール38の単位読出データ量に達しなかった回数を加算した回数を待ち発生回数として用いるようにしてもよい。この場合、待ち発生回数が、後段の画像処理モジュール38における「待ち」の度合いをより正確に反映した値となるので好ましい。

【0135】

また、上記態様において、後段のバッファモジュール40を介して連結された後段の画像処理モジュール38における「待ち」の発生回数に応じて、個々の画像処理モジュール38に対応するスレッドの実行優先度を変更することに加え、自モジュールにおける「待ち」の発生回数に応じてスレッドの実行優先度を変更する（詳しくは、待ち発生回数が比較的多い画像処理モジュール38に対応するスレッドは実行優先度を低下させ、待ち発生回数が比較的小さい画像処理モジュール38に対応するスレッドは実行優先度を増大させる）ようにしてもよい。

10

【0136】

また、個々の画像処理モジュール38に対応するスレッドの実行優先度を、個々のバッファモジュール40の後段の画像処理モジュール38が、個々のバッファモジュール40から画像データを取得する際の単位データ量に対する、個々のバッファモジュール40に記憶されている画像データのデータ量の比率に応じて変更するようにしてもよい。これは、例えばワークフロー管理部46Aが、一定の時間周期で以下の処理を行うことで実現できる。

20

【0137】

すなわち、まず個々のバッファモジュール40の現在の蓄積データ量を取得し、個々のバッファモジュール40について、後段の画像処理モジュール38の単位読出データ量に対する各バッファモジュール40の現在の蓄積データ量の比率を演算する。例えば蓄積データ量が画像のライン数で表される値であり、或るバッファモジュール40における蓄積データ量が「10ライン」、当該バッファモジュール40の後段の画像処理モジュール38の単位読出データ量が「1ライン」であるとすると蓄積データ量の比率は $10/1=10$ となり、後段の画像処理モジュール38の単位読出データ量が「8ライン」であるとすると蓄積データ量の比率は $10/8=1.25$ となる。そして、各バッファモジュール40毎に演算した蓄積データ量の比率の平均値を演算し、演算した蓄積データ量の比率の平均値と、個々のバッファモジュール40の蓄積データ量の比率の偏差に応じて、個々のバッファモジュール40の前段の各画像処理モジュール38に対応するスレッドの実行優先度を変更する。

30

【0138】

この実行優先度の変更は、蓄積データ量の比率が平均値よりも少ないバッファモジュール40の前段の画像処理モジュール38に対しては、対応するスレッドの実行優先度が上記偏差が大きくなるに従って増大し、蓄積データ量の比率が平均値よりも少ないバッファモジュールの前段の画像処理モジュール38に対しては、対応するスレッドの実行優先度が上記偏差が大きくなるに従って増大するように行うことができ、具体的には、例えば以下の式に従って行うことができる。

40

【0139】

実行優先度の変更率(%) = (平均値 - 蓄積データ量の比率) / 平均値 × 100

変更後の実行優先度 = 元の実行優先度 + (実行優先度 × 変更率) / 100

なお、上記演算に際し、蓄積データ量の比率の平均値に代えて蓄積データ量の比率の中央値を用いてもよい。

【0140】

蓄積データ量の比率が平均値よりも小さいバッファモジュール40は、後段の画像処理モジュール38における単位読出データ量に比して有効データのデータ量が乏しく、後段の画像処理モジュール38で比較的多数回の「待ち」が発生している可能性が高く、また

50

当該バッファモジュール40の前段の画像処理モジュール38における画像処理が画像処理部全体としての画像処理のボトルネックになっている可能性が高いが、上記処理ではこのような画像処理モジュール38に対応するスレッドの実行優先度が増大される。また、蓄積データ量の比率が平均値よりも大きいバッファモジュール40には、後段の画像処理モジュール38における単位読出データ量に比して十分なデータ量の有効データが記憶されているので、当該バッファモジュールの前段の画像処理モジュール38よりも、蓄積データ量の比率が比較的小さい他のバッファモジュール40の前段の画像処理モジュール38における画像処理を優先させた方が画像処理部全体としての画像処理を効率化できるが上記処理ではこのような画像処理モジュール38に対応するスレッドの実行優先度が低下される。

10

【0141】

これにより、個々の画像処理モジュール38に対応するスレッドの実行優先度が、後段のバッファモジュール40における蓄積データ量の比率（と蓄積データ量の比率の平均値との偏差）に応じて最適化されることになり、CPU12を有効に利用して高い処理効率で画像処理を行うことができる。また、メモリ取得待ち行列に登録される取得要求情報が、上記のようにして設定された実行優先度に従って配列されることで、要求されたメモリ領域を確保できずに複数のスレッドの実行を停止させた後に、確保可能なメモリ領域が増大したことでスレッドの実行停止状態を解除してメモリ領域を割り当てる場合にも、画像処理部50における以降の処理を高い処理効率で行わせることができる。

20

【0142】

また、画像処理部50を逐次並列処理方式で動作させる際のワークフロー管理部46Aの処理についても、図10に示したブロック単位逐次制御処理1~4に限られるものではなく、例として図14に示す面単位逐次制御処理1, 3, 4を行うようにしてもよい。図14に示す面単位逐次制御処理では、画像処理の実行が指示されると、図14(A)に示す面単位逐次制御処理1において、まずステップ500で1回の処理要求で指定する単位処理の実行回数を個々の画像処理モジュール38毎に決定し、次のステップ505において、画像処理部50のうち最前段の画像処理モジュール38に処理要求を入力する。また、画像処理モジュール38から処理完了通知が入力される毎に、図14(B)に示す面単位逐次制御処理3において、ステップ520で処理完了通知元の画像処理モジュール38に処理要求を再度入力する。これにより、画像処理部50のうちの最前段の画像処理モジュール38にのみ処理要求が繰り返し入力されることになる。

30

【0143】

また、画像処理モジュール38から全体処理終了通知が入力される毎に、図14(C)に示す面単位逐次制御処理4において、まずステップ540で全体処理終了通知入力元の画像処理モジュール38が最後段の画像処理モジュール38か否かが判定する。判定が否定された場合はステップ544へ移行し、パイプライン形態又は有向非循環グラフ形態の連結形態における画像処理モジュール38の位置が、全体処理終了通知元の画像処理モジュール38の次段である画像処理モジュール38に処理要求を入力する。これにより、画像処理を実行している画像処理モジュール38が、処理対象の画像データに対する画像処理を終了し、当該画像処理モジュール38から全体処理終了通知が入力された後に、画像処理を実行する画像処理モジュール38が次段の画像処理モジュール38へ切り替わることになり、画像1面分のデータに対する画像処理が最前段の画像処理モジュール38から逐次行われる逐次処理方式によって、処理対象の画像データに対する画像処理が行われる。そして、最後段の画像処理モジュール38から全体処理終了通知が入力され、ステップ540の判定が肯定されると、ステップ542で画像処理の完了がアプリケーション32へ通知される。なお、上述した面単位逐次制御処理1, 3, 4を行う逐次処理管理部は請求項5に記載の第2の逐次処理制御手段に対応している。

40

【0144】

また、上記では画像処理部50を並列処理方式で動作させる場合に、画像処理部50の個々のモジュールのプログラムを互いに独立したスレッドとして実行させる態様を説明し

50

たが、これに限定されるものではなく、画像処理部50を構成する個々のモジュールのプログラムを、各々複数のモジュールに対応する複数のスレッドとして実行させるようにしてもよい。図15は、4個の画像処理モジュール38と、画像処理モジュール38の間に設けられたバッファモジュール40をパイプライン形態で連結した構成において、最前段及び2段目の画像処理モジュール38とその間に設けられたバッファモジュール40をスレッドAとして実行させると共に、3段目及び最後段の画像処理モジュール38とその間に設けられたバッファモジュール40をスレッドBとしてスレッドAと並列に実行させる例を示しているが、このような場合、逐次動作する最前段の画像処理モジュール38と2段目の画像処理モジュール38の間に設けられたバッファモジュール40、及び、同じく逐次動作する3段目の画像処理モジュール38と最終段の画像処理モジュール38の間に設けられたバッファモジュール40については、排他制御無しのパッファモジュール40とし、並列に動作する2段目の画像処理モジュール38と3段目の画像処理モジュール38の間に設けられたバッファモジュール40については、排他制御有りのバッファモジュール40とすればよい。

10

【0145】

また、上記では並列処理管理部及び逐次処理管理部として、各々単一の処理管理部のプログラムが処理管理部ライブラリ47に登録されている態様を説明したが、これに限定されるものではなく、画像処理部50の動作環境等に応じて、複数の並列処理管理部の中から単一の並列処理管理部を選択したり、複数の逐次処理管理部の中から単一の逐次処理管理部を選択するように構成することも可能である。例えばプログラム実行リソース(例えばCPU12等)の数が1なら逐次処理管理部を選択することで画像処理部50で逐次処理方式によって画像処理を行わせ、プログラム実行リソースの数が2以上、かつ画像処理部50を構成する画像処理モジュール38の数未満ならば、画像処理部50で図15に示した並列処理方式(スレッドの数を抑制した並列処理)によって画像処理を行わせる並列処理管理部を選択し、プログラム実行リソースの数が2以上、かつ画像処理部50を構成する画像処理モジュール38の数以上ならば、画像処理部50の個々のモジュールのプログラムを互いに独立したスレッドとして実行させる並列処理方式で画像処理部50における画像処理を行わせる並列処理管理部を選択するようにしてもよい。

20

また、画像処理部50の個々のモジュールのプログラムを互いに独立したスレッドとして実行させる並列処理方式は、プログラム実行リソースの数が画像処理モジュールの数に近ければ有効に機能する可能性が高いので、「プログラム実行リソースの数/画像処理モジュールの数」が閾値(例えば0.8等)以上であれば、画像処理部50の個々のモジュールのプログラムを互いに独立したスレッドとして実行させる並列処理方式で画像処理部50における画像処理を行わせる並列処理管理部を選択するようにしてもよい。

30

【0146】

また、処理管理部46のプログラムは、記憶部20の処理管理部ライブラリ47に固定的に記憶されている必要はなく、コンピュータ10の外部から、例えばUSBメモリ等の外部記憶装置や通信回線等を介して、新たな処理管理部(並列処理管理部や逐次処理管理部)のプログラムを追加したり、既登録の処理管理部のプログラムを上書き更新可能としてもよい。CPU12の新たなアーキテクチャの採用等に応じて、最適な並列化の手法が変わる事も考えられるし、また最適な処理管理部のプログラムを当初より提供することが困難な場合や、処理管理部のアルゴリズムとしてより高効率のアルゴリズムが今後新たに開発される可能性もある。このような場合を考慮し、記憶部20の処理管理部ライブラリ47は、処理管理部のプログラムの新規追加や上書き更新が可能に構成することが望ましい。上記事項は請求項9記載の発明に対応している。

40

【0147】

また、例えば当初は逐次処理用の処理管理部(逐次処理管理部)のみを提供し、画像処理部50で並列処理を行わせることによる画像処理の高速化を所望しているユーザに対しては、別途料金を支払ってもらうことで並列処理管理部のプログラムの新規追加等の処理管理部のプログラムの更新を許可したり、ユーザとの間で締結したメンテナンス契約によ

50

り一定期間は処理管理部のプログラムの更新を許可するようにしてもよい。

【0148】

また、上記では本発明に係る画像処理プログラムに対応する画像処理プログラム群34が記憶部20に予め記憶(インストール)されている態様を説明したが、本発明に係る画像処理プログラムは、CD-ROMやDVD-ROM等の記録媒体に記録されている形態で提供することも可能である。

【図面の簡単な説明】

【0149】

【図1】本実施形態に係るコンピュータ(画像処理装置)の概略構成を示すブロック図である。

10

【図2】画像処理部の構築から画像処理の実行に至る一連の処理を説明するためのシーケンス図である。

【図3】画像処理部の構成例を示すブロック図である。

【図4】(A)は画像処理モジュール、(B)はバッファモジュールの概略構成及び実行される処理を各々示すブロック図である。

【図5】書込対象の画像データが複数の保管用単位バッファ領域に跨る場合を説明する概略図である。

【図6】読出対象の画像データが複数の保管用単位バッファ領域に跨っていた場合を説明する概略図である。

【図7】画像処理モジュールの制御部によって実行される画像処理モジュール制御処理の内容を示すフローチャートである。

20

【図8】並列処理管理部のワークフロー管理部によって実行される並列制御処理の内容を示すフローチャートである。

【図9】画像処理部における画像処理の流れを説明する概略図である。

【図10】逐次処理管理部のワークフロー管理部によって実行されるブロック単位逐次制御処理の内容を示すフローチャートである。

【図11】並列処理管理部のワークフロー管理部によって実行される並列制御処理の内容の他の例を示すフローチャートである。

【図12】画像処理部における一連の画像処理の進行に伴う、個々の画像処理モジュールに対応するスレッドの実行優先度の推移の一例を示す概略図である。

30

【図13】パイプライン形態又は有向非循環グラフ形態の連結形態における画像処理モジュールの位置の定義を説明するためのブロック図である。

【図14】逐次処理管理部のワークフロー管理部によって実行される面単位逐次制御処理の内容を示すフローチャートである。

【図15】画像処理部の構成の他の例を示すブロック図である。

【符号の説明】

【0150】

10 コンピュータ

12 CPU

20 記憶部

22 画像データ供給部

24 画像出力部

34 画像処理プログラム群

38 画像処理モジュール

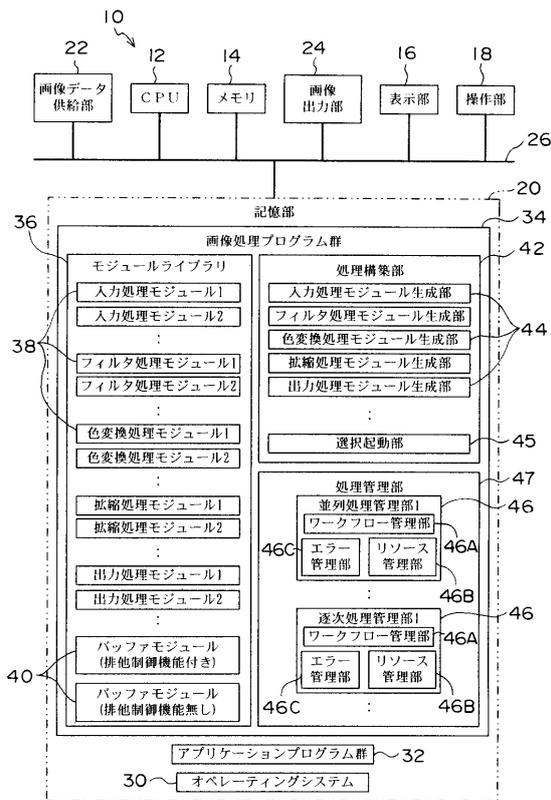
40 バッファモジュール

45 選択起動部

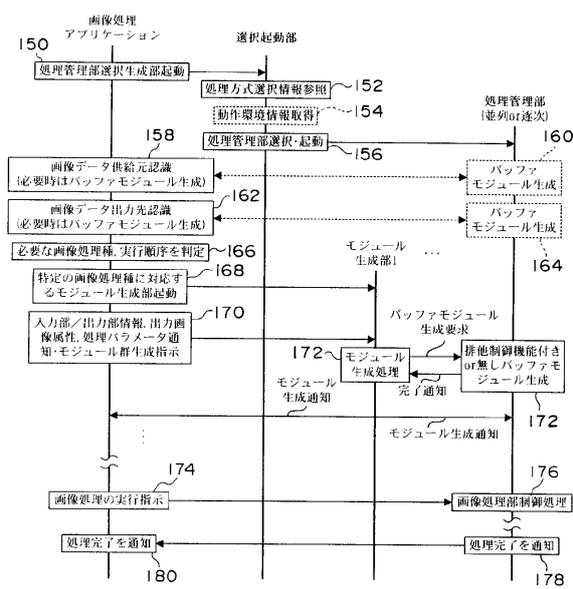
46 処理管理部

40

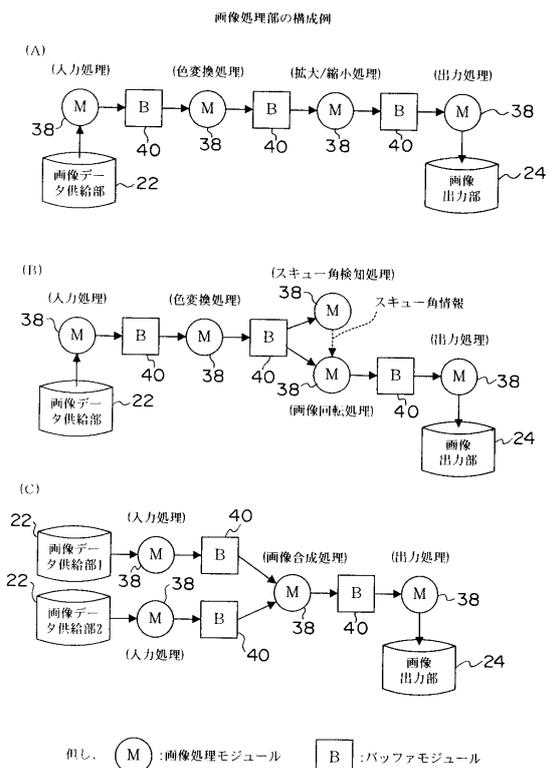
【図1】



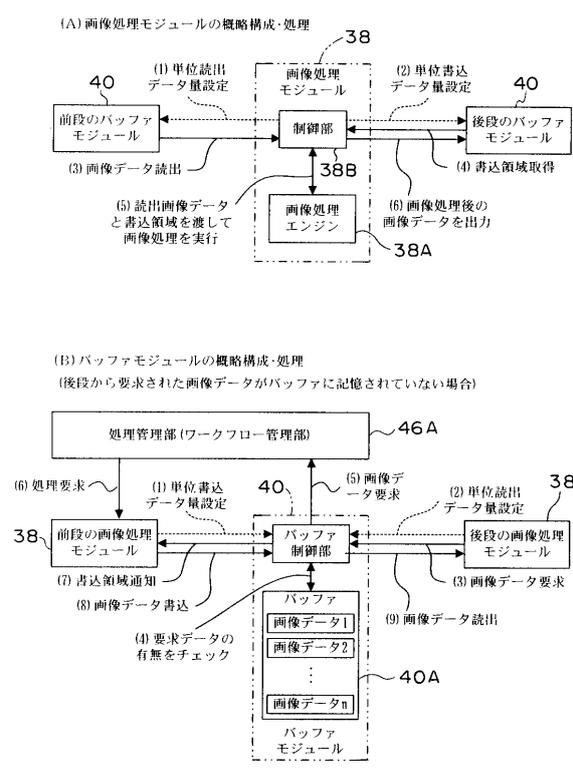
【図2】



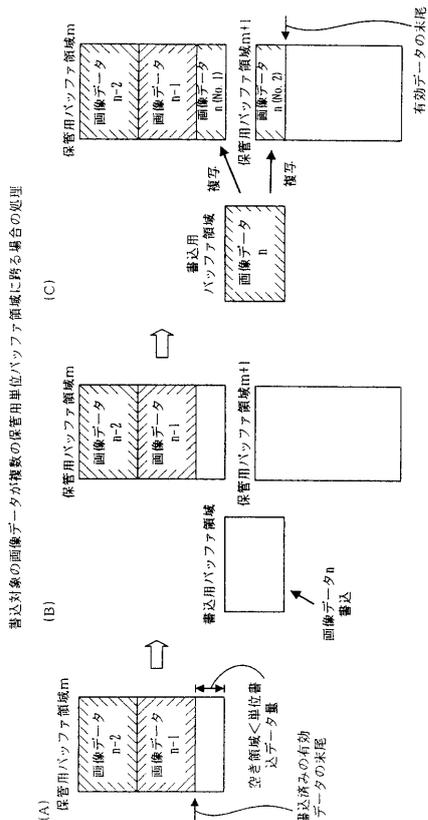
【図3】



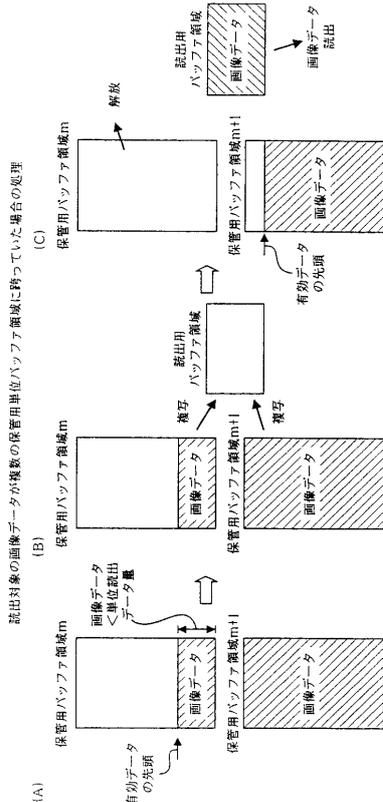
【図4】



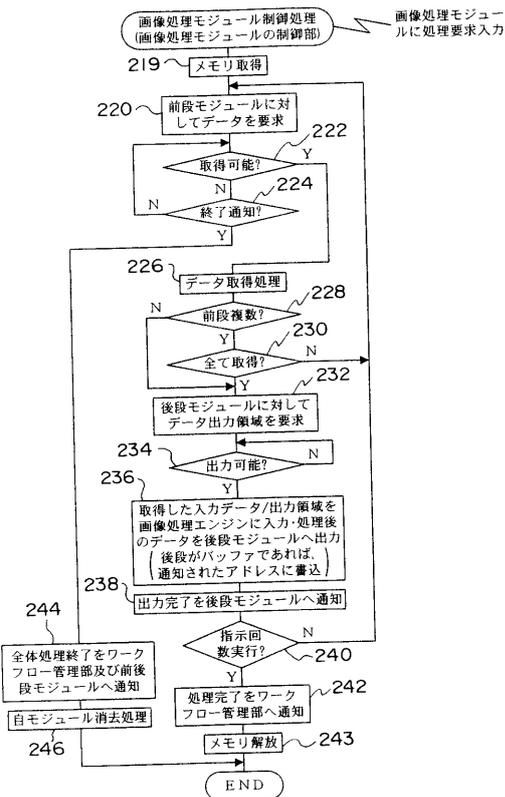
【図5】



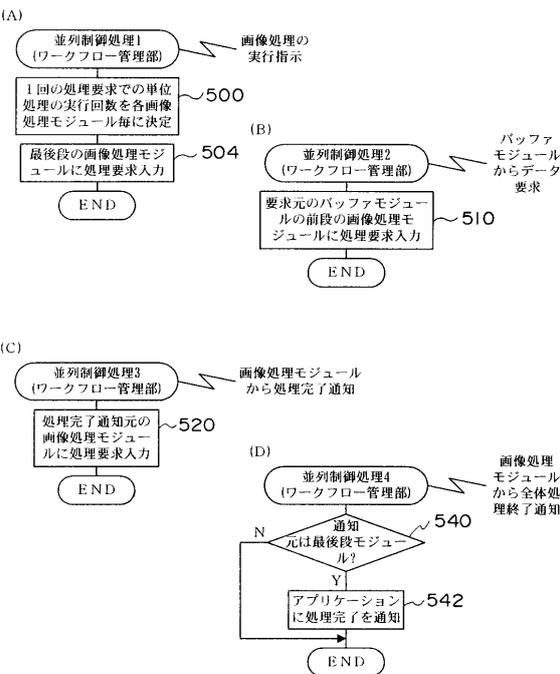
【図6】



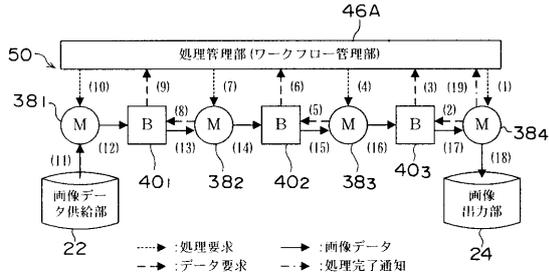
【図7】



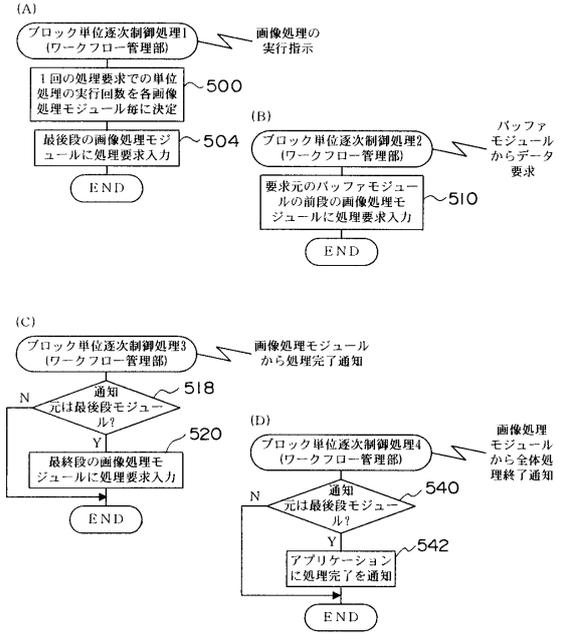
【図8】



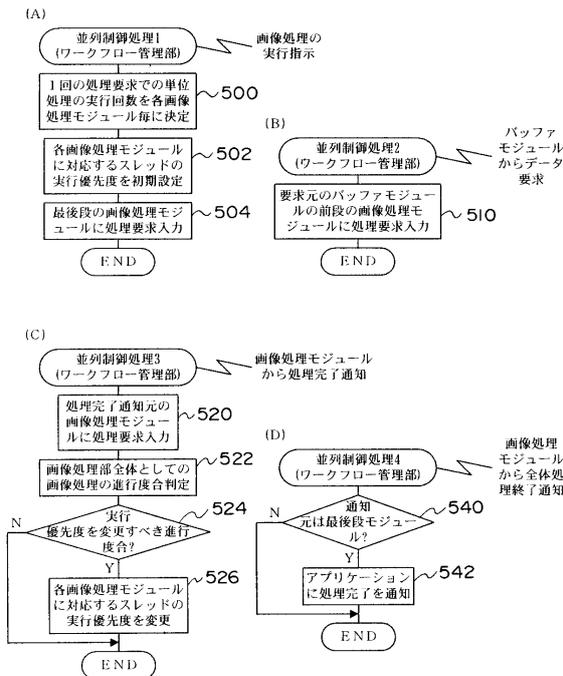
【図9】



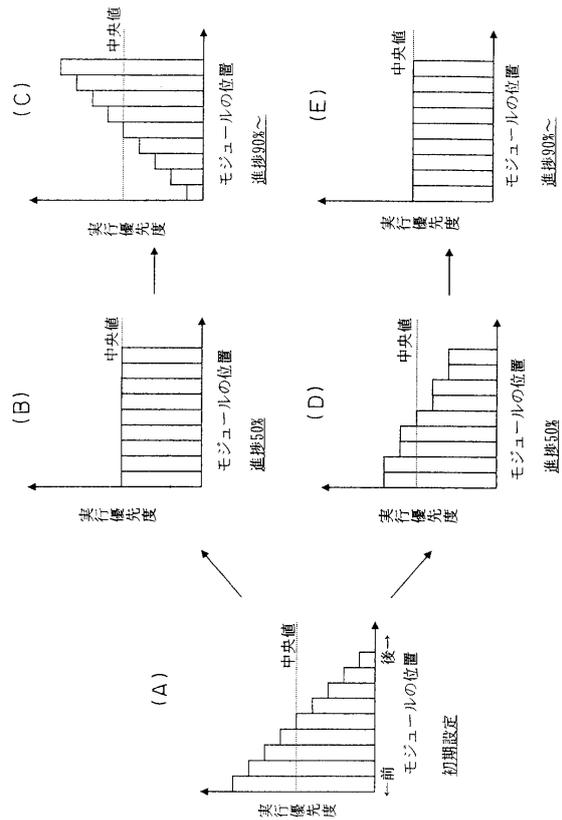
【図10】



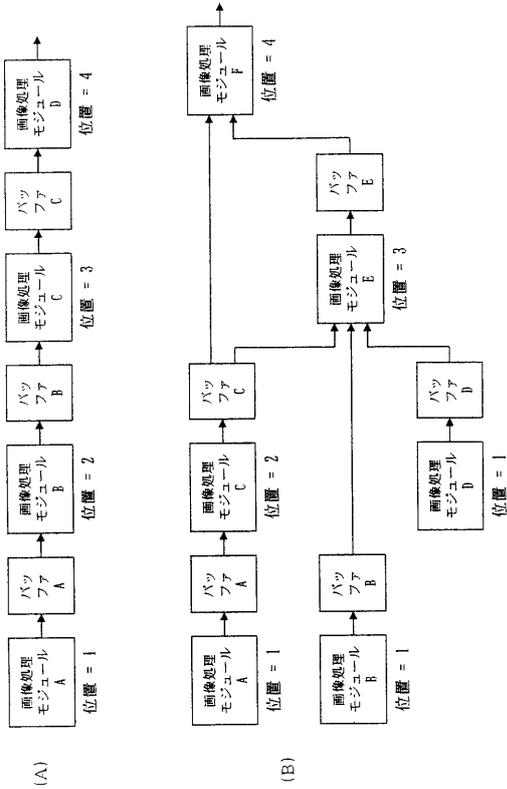
【図11】



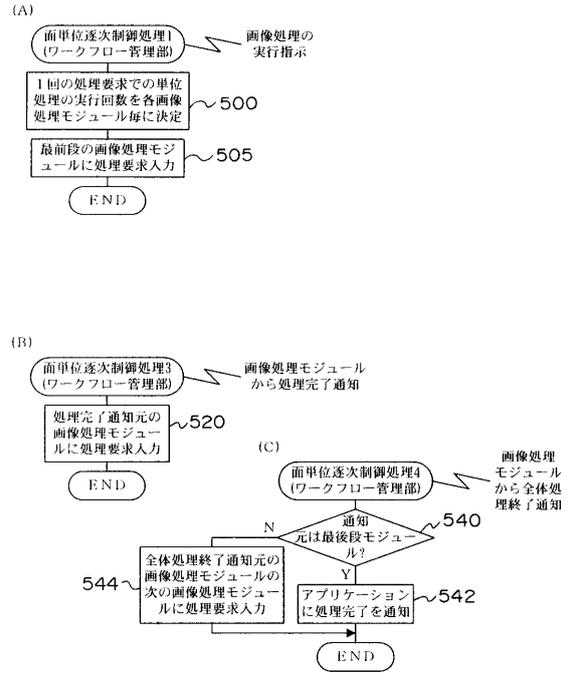
【図12】



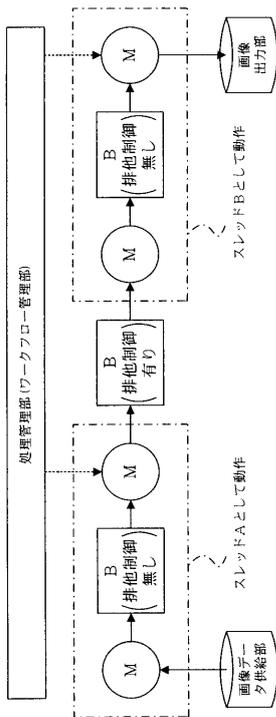
【図13】



【図14】



【図15】



フロントページの続き

- (72)発明者 長尾 隆
神奈川県足柄上郡中井町境430グリーンテクなかい 富士ゼロックス株式会社内
- (72)発明者 熊澤 幸夫
神奈川県足柄上郡中井町境430グリーンテクなかい 富士ゼロックス株式会社内
- (72)発明者 井坂 洋一
神奈川県足柄上郡中井町境430グリーンテクなかい 富士ゼロックス株式会社内
- (72)発明者 五十嵐 貴
東京都港区西麻布2丁目26番30号 富士写真フイルム株式会社内
- (72)発明者 杉本 裕介
東京都港区西麻布2丁目26番30号 富士写真フイルム株式会社内
- (72)発明者 板垣 和幸
東京都港区西麻布2丁目26番30号 富士写真フイルム株式会社内
- (72)発明者 金子 順一
神奈川県海老名市中央2丁目9番50号海老名プライムタワー 富士ゼロックス情報システム株式会社内

審査官 秦野 孝一郎

- (56)参考文献 特開平10-337932(JP,A)
特開2000-255117(JP,A)
特開2000-035890(JP,A)
特開平11-157144(JP,A)
特開2000-148577(JP,A)

(58)調査した分野(Int.Cl., DB名)
G06T 1/20