

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of

Yukio KUMAZAWA et al.

Application No.: 11/637,726

Filed: December 13, 2006

Docket No.: 130610

For: IMAGE PROCESSING APPARATUS, STORAGE MEDIUM AND DATA SIGNAL

CLAIM FOR PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following prior foreign application filed in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. 2006-179257, filed June 29, 2006.

In support of this claim, a certified copy of said original foreign application:

☒ is filed herewith.

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of this document.

Respectfully submitted,

James A. Oliff

Registration No. 27,075

Jeffrey R. Bousquet

Registration No. 57,771

JAO:JRB/kam

Date: January 5, 2007

OLIFF & BERRIDGE, PLC
P.O. Box 19928
Alexandria, Virginia 22320
Telephone: (703) 836-6400

**DEPOSIT ACCOUNT USE
AUTHORIZATION**
Please grant any extension
necessary for entry;
Charge any fee due to our
Deposit Account No. 15-0461

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
る事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2006年 6月29日

出願番号
Application Number:

特願2006-179257

パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号
country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

JP 2006-179257

願人
Applicant(s):

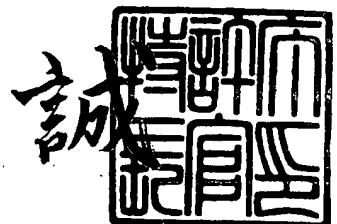
富士ゼロックス株式会社
富士フイルムホールディングス株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2006年12月11日

特許庁長官
Commissioner,
Japan Patent Office

中嶋



出証番号 出証特2006-3093774

【書類名】 特許願
【整理番号】 FE06-01459
【提出日】 平成18年 6月29日
【あて先】 特許庁長官殿
【国際特許分類】 G03G
【発明者】
 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい 富士ゼロックス株式会社内
 【氏名】 熊澤 幸夫
【発明者】
 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい 富士ゼロックス株式会社内
 【氏名】 長尾 隆
【発明者】
 【住所又は居所】 神奈川県足柄上郡中井町境 4 3 0 グリーンテクなかい 富士ゼロックス株式会社内
 【氏名】 井坂 洋一
【発明者】
 【住所又は居所】 東京都港区西麻布 2 丁目 2 6 番 3 0 号 富士写真フイルム株式会社内
 【氏名】 五十嵐 貴
【発明者】
 【住所又は居所】 東京都港区西麻布 2 丁目 2 6 番 3 0 号 富士写真フイルム株式会社内
 【氏名】 杉本 裕介
【発明者】
 【住所又は居所】 東京都港区西麻布 2 丁目 2 6 番 3 0 号 富士写真フイルム株式会社内
 【氏名】 板垣 和幸
【発明者】
 【住所又は居所】 神奈川県海老名市中央 2 丁目 9 番 5 0 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
 【氏名】 金子 順一
【特許出願人】
 【識別番号】 000005496
 【氏名又は名称】 富士ゼロックス株式会社
【特許出願人】
 【識別番号】 000005201
 【氏名又は名称】 富士写真フイルム株式会社
【代理人】
 【識別番号】 100079049
 【弁理士】
 【氏名又は名称】 中島 淳
 【電話番号】 03-3357-5171
【選任した代理人】
 【識別番号】 100084995
 【弁理士】
 【氏名又は名称】 加藤 和詳
 【電話番号】 03-3357-5171
 【連絡先】 担当

【選任した代理人】

【識別番号】 100085279

【弁理士】

【氏名又は名称】 西元 勝一

【電話番号】 03-3357-5171

【選任した代理人】

【識別番号】 100099025

【弁理士】

【氏名又は名称】 福田 浩志

【電話番号】 03-3357-5171

【手数料の表示】

【予納台帳番号】 006839

【納付金額】 16,000円

【提出物件の目録】

【物件名】 特許請求の範囲 1

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9503326

【包括委任状番号】 9503325

【包括委任状番号】 9503322

【包括委任状番号】 9503324

【包括委任状番号】 9800120

【書類名】 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機能を備えた複数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処理を行うバッファモジュールが各々連結されるように、個々のモジュールがパイプライン形態又は有向非循環グラフ形態で連結されて構成された画像処理部と、

前記画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合はメモリを確保し、前記記憶リソースの割当を必要としているモジュールに前記確保したメモリを前記記憶リソースとして割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記記憶リソースの割当を必要としているモジュールに前記確保した前記外部記憶装置の記憶領域を前記記憶リソースとして割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを前記確保した外部記憶装置の記憶領域に書き込み、前記別のモジュールに前記データを書き込んだ外部記憶装置の記憶領域を前記データが書き込まれていたメモリに代えて割り当てると共に、前記別のモジュールに割り当てていたメモリを、前記記憶リソースの割当を必要としているモジュールに前記記憶リソースとして割り当てる記憶リソース管理手段と、

を含む画像処理装置。

【請求項 2】

前記記憶リソース管理手段は、前記画像処理部を構成する個々の画像処理モジュールに対しては、個々の画像処理モジュールが必要とする記憶リソースとしてメモリを予め固定的に割り当てておくか、又は、前記個々の画像処理モジュール用のメモリを予め確保しておき、画像処理モジュールが記憶リソースを必要としている場合に、前記記憶リソースとしてメモリを割り当て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 3】

前記画像処理部の個々のバッファモジュールには、予め一定量のメモリが割り当てられており、

前記記憶リソース管理手段は、バッファモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が前記記憶リソースの割当を必要としているバッファモジュールに予め割り当てられたメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合は、前記予め割り当てられたメモリを前記記憶リソースとして前記バッファモジュールに割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記確保した前記外部記憶装置の記憶領域を前記記憶リソースとして前記バッファモジュールに割り当てることを特徴とする請求項 2 記載の画像処理装置。

【請求項 4】

前記記憶リソース管理手段は、記憶リソースとしてメモリを割り当てていた第 1 のバッファモジュールによる前記メモリの解放を監視し、前記メモリの解放を検知した場合に、記憶リソースとして既に外部記憶装置の記憶領域を割り当てた第 2 のバッファモジュールに対し、当該第 2 のバッファモジュールに記憶リソースとして既に割り当てた前記外部記憶装置の記憶領域に書き込まれているデータを前記解放されたメモリに書き込み、前記データを書き込んだメモリを前記データが書き込まれていた前記外部記憶装置の記憶領域に代えて割り当てることを特徴とする請求項 2 記載の画像処理装置。

【請求項 5】

前記記憶リソース管理手段は、前記記憶リソースとして既に外部記憶装置の記憶領域を

割り当てた第2のバッファモジュールが複数存在している場合に、前記外部記憶装置の記憶領域に代えてメモリを割り当てる第2のバッファモジュールを、前記記憶リソースとして外部記憶装置の記憶領域を割り当てた時期の古い順に選択することを特徴とする請求項4記載の画像処理装置。

【請求項6】

前記画像処理部の個々のバッファモジュールには優先度が設定されており、

前記記憶リソース管理手段は、前記記憶リソースとして既に外部記憶装置の記憶領域を割り当てた第2のバッファモジュールが複数存在している場合に、前記外部記憶装置の記憶領域に代えてメモリを割り当てる第2のバッファモジュールを前記優先度の降順に選択することを特徴とする請求項4記載の画像処理装置。

【請求項7】

前記画像処理部は、前記画像処理部を構成する個々のモジュールに対応するプログラムがプログラム実行リソースによって互いに並列に実行されることで動作し、

前記画像処理部を構成する個々の画像処理モジュールの前記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位置に応じて、前記個々の画像処理モジュールに対応する個々のプログラムの実行優先度の初期設定を行うと共に、前記画像処理部における画像処理の進行度合に応じて前記個々の画像処理モジュールに対応する個々のプログラムの実行優先度を変更し、かつ個々のバッファモジュールの優先度を、個々のバッファモジュールと直接連結された画像処理モジュールに対応するプログラムの実行優先度に応じて設定及び変更する実行優先度制御手段を更に備えたことを特徴とする請求項6記載の画像処理装置。

【請求項8】

前記記憶リソース管理手段は、前記個々のバッファモジュールに各々設けられ、自モジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合はメモリを確保し、前記確保したメモリを前記記憶リソースとして自モジュールに割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記確保した外部記憶装置の記憶領域を前記記憶リソースとして自モジュールに割り当て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9】

前記画像処理部を構成する個々のモジュールが互いに並列に動作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】

メモリ及び外部記憶装置を備えたコンピュータを、

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機能を備えた複数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処理を行うバッファモジュールが各々連結されるように、個々のモジュールがパイプライン形態又は有向非循環グラフ形態で連結されて構成された画像処理部、

及び、前記画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合はメモリを確保し、前記記憶リソースの割当を必要としているモジュールに前記確保したメモリを前記記憶リソースとして割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記記憶リソースの割当を必要としているモジュールに前記確保した前記外部記憶装置の記憶領域を前記記憶リソースとして割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを前記確保した外部記憶装置の記憶領域に書き込み、前記別のモ

ジュールに前記データを書き込んだ外部記憶装置の記憶領域を前記データが書き込まれていたメモリに代えて割り当てると共に、前記別のジュールに割り当てていたメモリを、前記記憶リソースの割当を必要としているジュールに前記記憶リソースとして割り当てる記憶リソース管理手段
として機能させるための画像処理プログラム。

【書類名】明細書

【発明の名称】画像処理装置及びプログラム

【技術分野】

【0 0 0 1】

本発明は画像処理装置及びプログラムに係り、特に、複数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モジュールが各々連結されるように、個々のモジュールがパイプライン形態又は有向非循環グラフ形態で連結され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】

【0 0 0 2】

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 D T P (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な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。

【0 0 0 3】

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 D A G (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。

【0 0 0 4】

また、上記に関連して以下の技術が知られている。すなわち、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。

【0 0 0 5】

また、特許文献 3 には、マルチプロセッサシステムにおいて、単体処理ユニットの優先度や処理結果の依存性に基づいてストール時間（次の処理が開始可能となる条件が満たされるまで処理を行わずに停止している時間）が生じないようにスケジューリングを行う技術が開示されている。

【0 0 0 6】

更に、特許文献 4 には、並列動作する複数の画像処理プロセスで必要とする最低限の画像メモリ量を予め算出し、システム起動時に算出した必要量の画像メモリをリザーブすると共に、リザーブされていないメモリのメモリ使用量が動的に変化した場合に各画像処理プロセスに通知することで、各画像処理プロセスの中断を防止しメモリ割り当ての効率化を実現する技術が開示されている。

【特許文献 1】特開平 5 - 2 6 0 3 7 3 号公報

【特許文献 2】特開 2 0 0 4 - 2 8 7 8 8 3 号公報

【特許文献 3】特開平 1 0 - 2 3 2 8 5 6 号公報

【特許文献4】特開2003-091425号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

特許文献1に記載の技術のように、複数種の画像処理モジュールを任意に組み合わせて所望の画像処理を行う画像処理装置を構成する場合、以下に述べるような問題がある。すなわち、各画像処理モジュールには、実行する画像処理の種類や内容に応じて処理し易い単位（例えば画素単位、1ライン単位、複数ライン単位、面単位等）がある。しかし、各画像処理モジュールを任意の順番で連結し協調して処理することを可能とするためには、全ての画像処理モジュールの出力単位を揃えるか、或いは各画像処理モジュールが任意の入力単位に対応可能に構成する必要があり、各画像処理モジュールの構成が複雑になる。また、各画像処理モジュールは他の画像処理モジュールと連携して動作するため、各画像処理モジュールには、入力された画像データに対して実際に画像処理を行う部分以外に、自モジュールと連結された他の画像処理モジュールとの間で画像データを受け渡す処理を制御する部分も必要となり、各画像処理モジュールの構成は一層複雑になる。

【0008】

上記の問題は、特許文献2に記載の技術のようにデータ処理手段の前後に記憶手段を設けた構成を用い、個々のデータ処理手段で各種の画像処理を行わせると共に、個々のデータ処理手段を、実行する画像処理の種類や内容に応じた処理し易い単位ずつ記憶手段から画像データを取得するように構成することで解決可能である。しかし、複数の画像処理モジュール及びバッファ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構築した画像処理部において、各モジュールを並列に動作させて所望の画像処理を行わせる場合、個々の画像処理モジュールやバッファモジュールは、処理の実行に必要なメモリの取得・解放を繰り返しながら処理を行うことになるが、或るモジュールでメモリを取得できない状況が発生すると、当該モジュールで処理を継続できないために、結果として画像処理部で行っている画像処理全体も失敗に終わってしまうという問題があった。

【0009】

また、特許文献4に記載の技術は、個々の画像処理プロセスが必要とする最低限の画像メモリ量をシステム起動時にリザーブするので、個々の画像処理プロセスが最低限の画像メモリ量で動作している間は、特定の画像処理プロセスがメモリを取得できないことで画像処理全体が失敗に終わってしまうことを回避できる。しかしながら、特許文献4に記載の技術では、或る画像処理プロセスが高解像度の画像を扱うために、通常の画像を扱う場合よりも大量のメモリを必要としている状況で、必要量のメモリが取得できなかった場合はメモリ不足エラー処理が行われるので、ユーザが高解像度の画像に対する画像処理の実行を所望していたとしても、メモリ不足が生じているとユーザが所望する画像処理が行われないことになる。

【0010】

本発明は上記事実を考慮して成されたもので、パイプライン形態又は有向非循環グラフ形態で連結された複数の画像処理モジュール及びバッファモジュールを並列に動作させる場合にも、メモリの不足に拘わらず画像処理全体を継続させることが可能な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】

【0011】

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機能を備えた複数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処理を行うバッファモジュールが各々連結されるように、個々のモジュールがパイプライン形態又

は有向非循環グラフ形態で連結されて構成された画像処理部と、前記画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合にはメモリを確保し、前記記憶リソースの割当を必要としているモジュールに前記確保したメモリを前記記憶リソースとして割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記記憶リソースの割当を必要としているモジュールに前記確保した前記外部記憶装置の記憶領域を前記記憶リソースとして割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを前記確保した外部記憶装置の記憶領域に書き込み、前記別のモジュールに前記データを書き込んだ外部記憶装置の記憶領域を前記データが書き込まれていたメモリに代えて割り当てると共に、前記別のモジュールに割り当てていたメモリを、前記記憶リソースの割当を必要としているモジュールに前記記憶リソースとして割り当てる記憶リソース管理手段と、を含んで構成されている。

【0012】

請求項1記載の発明に係る画像処理部は、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画像処理の処理結果を自モジュールの後段へ出力する機能を備えた複数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処理を行うバッファモジュールが各々連結されるように、個々のモジュールがパイプライン形態又は有向非循環グラフ形態で連結されて構成されている。

【0013】

また、請求項1記載の発明に係る記憶リソース管理手段は、画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリ（例えばRAM）の残量以下か否か判定し、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下の場合にはメモリを確保し、記憶リソースの割当を必要としているモジュールに確保したメモリを記憶リソースとして割り当てる。また、割当が必要な記憶リソースの容量が確保可能なメモリの残量よりも大きい場合、記憶リソース管理手段は、外部記憶装置（例えばHDD(Hard Disk Drive)やフラッシュメモリ等）の記憶領域を確保し、記憶リソースの割当を必要としているモジュールに確保した外部記憶装置の記憶領域を記憶リソースとして割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを確保した外部記憶装置の記憶領域に書き込み、別のモジュールにデータを書き込んだ外部記憶装置の記憶領域をデータが書き込まれていたメモリに代えて割り当てると共に、別のモジュールに割り当てていたメモリを、記憶リソースの割当を必要としているモジュールに記憶リソースとして割り当てる。

【0014】

このように、請求項1記載の発明では、画像処理部を構成する任意のモジュールが記憶リソースの割当を必要しているにも拘わらず、割当が必要な記憶リソースの容量が確保可能なメモリの残量よりも大きい場合に、記憶リソースの割当を必要としているモジュールに対し、外部記憶装置の記憶領域を確保して記憶リソースとして割り当てるか、別のモジュールに割り当てていたメモリを記憶リソースとして割り当てる（別のモジュールに対してはメモリに代えて外部記憶装置の記憶領域を割り当てる）ので、アクセス速度が低速な外部記憶装置の記憶領域を記憶リソースとして割り当てたモジュールで処理速度が低下する可能性はあるものの、記憶リソースの割当を必要としているモジュールに記憶リソースを割り当てできないことで、当該モジュールでエラーが発生して画像処理部における画像処理全体が失敗に終わってしまうことを回避することができる。従って、請求項1記載の発明によれば、パイプライン形態又は有向非循環グラフ形態で連結された複数の画像処理モジ

ジュール及びバッファモジュールを並列に動作させる場合にも、メモリの不足に拘わらず画像処理全体を継続させることが可能となる。

【0015】

なお本発明は、例えば請求項9に記載したように、画像処理部を構成する個々のモジュールが互いに並列に動作する態様に好適であるが、個々のモジュールが逐次動作する態様にも適用可能である。

【0016】

また本発明において、記憶リソースは、バッファモジュールでは主に画像データの一時保管に用いられるのに対し、画像処理モジュールでは主に画像処理時のワークメモリとして用いられ頻繁にアクセスされるので、アクセス速度が低速な外部記憶装置の記憶領域が記憶リソースとして割り当てられた場合に処理速度に与える影響（処理速度の低下度合）は、バッファモジュールよりも画像処理モジュールの方が大きい。これを考慮すると、請求項1記載の発明において、記憶リソース管理手段は、例えば請求項2に記載したように、画像処理部を構成する個々の画像処理モジュールに対しては、個々の画像処理モジュールが必要とする記憶リソースとしてメモリを予め固定的に割り当てておくか、又は、個々の画像処理モジュール用のメモリを予め確保しておき、画像処理モジュールが記憶リソースを必要としている場合に、記憶リソースとしてメモリを割り当てるように構成することが好ましい。これにより、画像処理モジュールに対しては記憶リソースとして常にメモリが割り当てられ、外部記憶装置の記憶領域はバッファモジュールに対してのみ記憶リソースとして割り当てられることになり、外部記憶装置の記憶領域を記憶リソースとして割り当てるに伴う処理速度の低下を抑制することができる。

【0017】

また、請求項2記載の発明において、画像処理部の個々のバッファモジュールには予め一定量のメモリが割り当てられていてもよく、この場合、記憶リソース管理手段は、例えば請求項3に記載したように、バッファモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が記憶リソースの割当を必要としているバッファモジュールに予め割り当てられたメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が予め割り当てられたメモリの残量以下の場合は、予め割り当てられたメモリを記憶リソースとして前記バッファモジュールに割り当て、割当が必要な記憶リソースの容量が予め割り当てられたメモリの残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、確保した外部記憶装置の記憶領域を記憶リソースとしてバッファモジュールに割り当てるように構成してもよい。請求項3記載の発明では、予め割り当てられた一定量のメモリを越えるメモリがバッファモジュールに割り当てられることがないので、バッファモジュールを含む画像処理部の各モジュールに記憶リソースとして大量のメモリが割り当てられることで、メモリを必要としている他のスレッド、プロセス、オブジェクトの実行に支障が生ずることを防止することができる。

【0018】

また、請求項2記載の発明において、記憶リソース管理手段は、例えば請求項4に記載したように、記憶リソースとしてメモリを割り当てていた第1のバッファモジュールによるメモリの解放を監視し、メモリの解放を検知した場合に、記憶リソースとして既に外部記憶装置の記憶領域を割り当てた第2のバッファモジュールに対し、当該第2のバッファモジュールに記憶リソースとして既に割り当てた外部記憶装置の記憶領域に書き込まれているデータを解放されたメモリに書き込み、データを書き込んだメモリをデータが書き込まれていた外部記憶装置の記憶領域に代えて割り当てるように構成することが好ましい。これにより、メモリの一時的な不足により記憶リソースとして外部記憶装置の記憶領域が割り当てられたバッファモジュールが存在していた場合に、メモリの不足が解消したにも拘わらず前記バッファモジュールが記憶リソースとして外部記憶装置の記憶領域を継続使用することで、画像処理部における画像処理全体の処理速度が低下している状態が無駄に続くことを防止することができる。

【0019】

なお、請求項4記載の発明において、記憶リソースとして既に外部記憶装置の記憶領域を割り当てた第2のバッファモジュールが複数存在している場合、記憶リソース管理手段は、例えば請求項5に記載したように、外部記憶装置の記憶領域に代えてメモリを割り当てる第2のバッファモジュールを、記憶リソースとして外部記憶装置の記憶領域を割り当てた時期の古い順に選択することができる。この場合、記憶リソースとして外部記憶装置の記憶領域が割り当てられた個々のバッファモジュールが、記憶リソースとして外部記憶装置の記憶領域が割り当てられてから当該外部記憶装置の記憶領域に代えてメモリが割り当てられる迄の待ち時間のばらつきを小さくすることができる。

【0020】

また、例えば請求項6に記載したように、画像処理部の個々のバッファモジュールに優先度を設定しておき、記憶リソース管理手段は、記憶リソースとして既に外部記憶装置の記憶領域を割り当てた第2のバッファモジュールが複数存在している場合に、外部記憶装置の記憶領域に代えてメモリを割り当てる第2のバッファモジュールを優先度の降順に選択するようにしてもよい。この場合、記憶リソースとして外部記憶装置の記憶領域が割り当てられた複数のバッファモジュールのうち、設定された優先度の高いバッファモジュールから優先的に記憶リソースの再割り当て（外部記憶装置の記憶領域に代わるメモリの割り当て）が行われることになる。

【0021】

また、請求項6記載の発明において、画像処理部が、画像処理部を構成する個々のモジュールに対応するプログラムがプログラム実行リソース（例えばCPU、MMX (MultiMedia eXtention)用の演算器やSSE (Streaming SIMD Extension)用の演算器、CPUと別に設けられたDSP (Digital Signal Processor)等の高速演算器等）によって互いに並列に実行されることで動作する場合、例えば請求項7に記載したように、画像処理部を構成する個々の画像処理モジュールの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位置に応じて、個々の画像処理モジュールに対応する個々のプログラムの実行優先度の初期設定を行うと共に、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画像処理モジュールに対応する個々のプログラムの実行優先度を変更し、かつ個々のバッファモジュールの優先度を、個々のバッファモジュールと直接連結された画像処理モジュールに対応するプログラムの実行優先度に応じて設定及び変更する実行優先度制御手段を設けてもよい。

【0022】

なお、個々のバッファモジュールの優先度を、個々のバッファモジュールと直接連結された画像処理モジュールに対応するプログラムの実行優先度に応じて設定及び変更することは、具体的には、例えば個々のバッファモジュールの優先度を、個々のバッファモジュールと直接連結されている画像処理モジュールに対応するプログラムの実行優先度に一致させるか、複数の画像処理モジュールと直接接続されている場合は、複数の画像処理モジュールに対応するプログラムの実行優先度の平均、最大、最小の何れかに相当する優先度を設定することで行うことができる。個々の画像処理モジュールに対応する個々のプログラムの実行優先度を上記のように初期設定及び変更することで、画像処理部における画像処理の処理効率を向上させることができる。また、個々のバッファモジュールの優先度を、個々のバッファモジュールと直接連結された画像処理モジュールに対応するプログラムの実行優先度に応じて設定及び変更することで、対応するプログラムの実行優先度の高い画像処理モジュールと直接接続されたバッファモジュールに、記憶リソースとしてメモリが割り当てられる可能性が高くなり、画像処理部における画像処理の処理効率を更に向上させることが可能となる。

【0023】

なお、請求項6に記載した個々のバッファモジュールの優先度は、記憶リソースの再割り当てを行うバッファモジュールの選択にのみ適用することに限られるものではなく、例えば記憶リソースの割当を必要としているバッファモジュールが存在しており、かつ割当が必要な記憶リソースの容量が確保可能なメモリの残量よりも大きい場合に、記憶リソー

スとして既にメモリを割り当てたバッファモジュールの中に、記憶リソースの割当を必要としているバッファモジュールよりも優先度の低いバッファモジュールが存在しているか否か判定し、該当するバッファモジュールが存在しているときに、外部記憶装置の記憶領域を確保し、優先度の低いバッファモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを確保した外部記憶装置の記憶領域に書き込み、優先度の低いバッファモジュールに、データを書き込んだ外部記憶装置の記憶領域を前記データが書き込まれていたメモリに代えて割り当てると共に、優先度の低いバッファモジュールに割り当てていたメモリを、記憶リソースの割当を必要としているバッファモジュール（優先度がより高いバッファモジュール）に記憶リソースとして割り当てるようにしてもよい。

【0024】

また、請求項1記載の発明において、記憶リソース管理手段は個々のバッファモジュールに各々設けられていてもよく、この場合、記憶リソース管理手段は、例えば請求項8に記載したように、個々のバッファモジュールに各々設けられ、自モジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下の場合はメモリを確保し、確保したメモリを記憶リソースとして自モジュールに割り当て、割当が必要な記憶リソースの容量が確保可能なメモリの残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、確保した外部記憶装置の記憶領域を記憶リソースとして自モジュールに割り当てるように構成することができる。

【0025】

請求項10記載の発明に係る画像処理プログラムは、メモリ及び外部記憶装置を備えたコンピュータを、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機能を備えた複数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処理を行うバッファモジュールが各々連結されるように、個々のモジュールがパイプライン形態又は有向非循環グラフ形態で連結されて構成された画像処理部、及び、前記画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、割当が必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が前記残量以下の場合はメモリを確保し、前記記憶リソースの割当を必要としているモジュールに前記確保したメモリを前記記憶リソースとして割り当て、割当が必要な記憶リソースの容量が前記残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、前記記憶リソースの割当を必要としているモジュールに前記確保した前記外部記憶装置の記憶領域を前記記憶リソースとして割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに記憶リソースとして既に割り当てたメモリに書き込まれているデータを前記確保した外部記憶装置の記憶領域に書き込み、前記別のモジュールに前記データを書き込んだ外部記憶装置の記憶領域を前記データが書き込まれていたメモリに代えて割り当てると共に、前記別のモジュールに割り当てていたメモリを、前記記憶リソースの割当を必要としているモジュールに前記記憶リソースとして割り当てる記憶リソース管理手段として機能させる。

【0026】

請求項10記載の発明に係る画像処理プログラムは、メモリ及び外部記憶装置を備えたコンピュータを、上記の画像処理部及び記憶リソース管理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、パイプライン形態又は有向非循環グラフ形態で連結された複数の画像処理モジュール及びバッファモジュールを並列に動作させる場合にも、メモリの不足に拘わらず画像処理全体を継続させることが可能となる。

【発明の効果】

【0027】

以上説明したように本発明は、画像処理部を構成するモジュールが記憶リソースの割当を必要としている場合に、必要な記憶リソースの容量が確保可能なメモリの残量以下か否か判定し、割当が必要な記憶リソースの容量が残量以下の場合はメモリを確保し、確保したメモリを割り当て、割当が必要な記憶リソースの容量が残量よりも大きい場合は、外部記憶装置の記憶領域を確保し、確保した外部記憶装置の記憶領域を割り当てるか、又は、外部記憶装置の記憶領域を確保し、別のモジュールに既に割り当てたメモリに書き込まれているデータを確保した外部記憶装置の記憶領域に書き込み、データを書き込んだ外部記憶装置の記憶領域をデータが書き込まれていたメモリに代えて前記別のモジュールに割り当てると共に、前記別のモジュールに割り当てていたメモリを、記憶リソースの割当を必要としているモジュールに割り当てるようにしたので、パイプライン形態又は有向非循環グラフ形態で連結された複数の画像処理モジュール及びバッファモジュールを並列に動作させる場合にも、メモリの不足に拘わらず画像処理全体を継続させることが可能となる、という優れた効果を有する。

【発明を実施するための最良の形態】

【0028】

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。

【0029】

〔第1実施形態〕

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。

【0030】

コンピュータ10はCPU12、DRAM又はSRAM等から成るメモリ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。

【0031】

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。

【0032】

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 1 0 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 3 4、CPU 1 2 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 3 2 のプログラム（図 1 ではアプリケーションプログラム群 3 2 と表記）が各々記憶されている。

【0033】

画像処理プログラム群 3 4 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 3 4 によって実現される画像処理装置は、アプリケーション 3 2 からの構築指示に従い、アプリケーション 3 2 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 3 2 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 3 4 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 3 2 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 3 4 に行わせるアプリケーション 3 2 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なくなるので、開発負荷を軽減することができる。

【0034】

また、画像処理プログラム群 3 4 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 3 2 からの構築指示に従い、アプリケーション 3 2 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 3 2 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。

【0035】

以下、画像処理プログラム群 3 4 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 3 4 はモジュールライブラリ 3 6 と、処理構築部 4 2 のプログラムと、処理管理部 4 6 のプログラムに大別される。詳細は後述するが、本実施形態に係る処理構築部 4 2 は、アプリケーションからの指示により、例として図 2 に示すように、予め定められた画像処理を行う 1 つ以上の画像処理モジュール 3 8 と、個々の画像処理モジュール 3 8 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 4 0 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 5 0 を構築する。画像処理部 5 0 を構成する個々の画像処理モジュールの実体は CPU 1 2 によって実行され CPU 1 2 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、CPU 1 2 によって実行され CPU 1 2 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 3 6 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 3 8 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 5 0 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。

【0036】

個々の画像処理モジュール 3 8 は、例として図 3 (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 3 8 A と、画像処理モジュール 3 8 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

3 8 A の制御を行う制御部 3 8 B から構成されている。個々の画像処理モジュール 3 8 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 3 8 A 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 3 8 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 3 8 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 3 8 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 3 8 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。

【0037】

また、モジュールライブラリ 3 6 には、画像処理エンジン 3 8 A 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 3 8 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 3 8 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 3 8、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 3 8 等の複数の画像処理モジュール 3 8 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 3 8 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 3 8 やその逆へ変換する画像処理モジュール 3 8、 $L^*a^*b^*$ 色空間等の他の色空間変換を行う画像処理モジュール 3 8 が各々用意されている。

【0038】

また、画像処理モジュール 3 8 の制御部 3 8 B は、画像処理エンジン 3 8 A 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 4 0）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 3 8 A 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 4 0）へ出力する（画像処理エンジン 3 8 A 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 3 8 A 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 3 8 A 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 3 6 には、画像処理エンジン 3 8 A 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、単位書込データ量が異なる画像処理モジュール 3 8 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 3 8 における単位処理データ量についても、前述した画像 1 面分に限られるものではなく、同じ画像回転処理を行いかつ単位処理データ量が互いに異なる（例えば画像の 1 ライン分や複数ライン分等の）複数の画像処理モジュール 3 8 がモジュールライブラリ 3 6 に含まれていても良い。

【0039】

また、モジュールライブラリ 3 6 に登録されている個々の画像処理モジュール 3 8 のプログラムは、画像処理エンジン 3 8 A に相当するプログラムと制御部 3 8 B に相当するプログラムから構成されているが、制御部 3 8 B 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 3 8 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 3 8 は、画像処理エンジン 3 8 A で実行される画像処理の種類や内容に拘わらず、制御部 3 8 B に相当するプログラムが共通化されている（制御部 3 8 B 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 3 8 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。

【0040】

なお、画像処理モジュール 3 8 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 3 8 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 3 8 について、制御部 3 8 B 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 3 4 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 3 4 のうちモジュールライブラリ 3 6 に登録する画像処理モジュール 3 8 の数や種類等については、画像処理プログラム群 3 4 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。

【0 0 4 1】

また、画像処理部 5 0 を構成する個々のバッファモジュール 4 0 は、例として図 3 (B) にも示すように、バッファ 4 0 A と、バッファモジュール 4 0 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 4 0 A の管理を行うバッファ制御部 4 0 B から構成されている。なお、バッファ 4 0 A はコンピュータ 1 0 に設けられたメモリ 1 4 からオペレーティングシステム 3 0 及びリソース管理部 4 6 B を通じて確保されたメモリ領域で構成されるが、バッファモジュール 4 0 の生成時にメモリ 1 4 が不足していた場合は、リソース管理部 4 6 B によって代わりに記憶部 2 0 の記憶領域が確保され、確保された記憶部 2 0 の記憶領域で代替される（リソース管理部 4 6 B による処理の詳細は後述する）。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 もその実体は CPU 1 2 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 3 6 にはバッファ制御部 4 0 B 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 4 0 B 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）。

【0 0 4 2】

また、アプリケーション 3 2 からの指示に従って画像処理部 5 0 を構築する処理構築部 4 2 は、図 1 に示すように複数種のモジュール生成部 4 4 から構成されている。複数種のモジュール生成部 4 4 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 3 2 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 3 8 及びバッファモジュール 4 0 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 4 4 の一例として、モジュールライブラリ 3 6 に登録されている個々の画像処理モジュール 3 8 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 4 4 を示しているが、個々のモジュール生成部 4 4 に対応する画像処理は、複数種の画像処理モジュール 3 8 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 3 2 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 4 4 を順次起動する。これにより、アプリケーション 3 2 によって順次起動されたモジュール生成部 4 4 により、必要とされる画像処理を行う画像処理部 5 0 が構築されることになる。

【0 0 4 3】

また図 1 に示すように、処理管理部 4 6 は、画像処理部 5 0 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 4 6 A、画像処理部 5 0 の各モジュールによるメモリ 1 4 や各種のファイル等のコンピュータ 1 0 のリソースの使用を管理するリソース管理部 4 6 B、及び、画像処理部 5 0 で発生したエラーを管理するエラー管理部 4 6 C を含んで構成されている。なお、本実施形態において、処理構築部 4 2 によって構築される画像処理部 5 0 は、画像処理部 5 0 を構成する個々の画像処理モジュール 3 8 が、画像 1 面分よりも小さいデータ量を単位として後段へ画像データを引き渡ししながら並列に画像処理を行うように動作する。

【0 0 4 4】

なお、エラー管理部 4 6 C は、画像処理部 5 0 が画像処理を実行している途中でエラーが発生した場合に、発生したエラーの種別・発生箇所等のエラー情報を取得し、画像処理プログラム群 3 4 がインストールされたコンピュータ 1 0 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処理を行う。

【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。

【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、まず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 の種別を認識し、認識した種別がバッファ領域（メモリ 14 又は記憶部 20 の一部領域）であった場合には、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し、バッファ制御部 40B のプログラムを実行するスレッドを生成する（バッファ制御部 40B を生成する）ことで、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）を生成する。また、スレッドに代えてプロセス又はオブジェクトとしてバッファモジュール 40 を生成するようにしてもよい。

【0047】

続いてアプリケーション 32 は、上記と同様に、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 の種別を認識し、認識した種別がバッファ領域（メモリ 14 又は記憶部 20 の一部領域）であった場合は、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 を上記と同様に生成する。ここで生成されたバッファモジュール 40 は画像出力部 24 として機能する。また、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。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 が、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処理を行うことによって実現することができる。

【0048】

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（モジュール生成部 44 のプログラムを実行するプロセス、スレッド又はオブジェクトを生成）した後に、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。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、アプリケーション 3 2 は、指示したモジュール生成部 4 4 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 4 4 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を個々の画像処理の実行順序の昇順に繰り返す。

【0049】

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 2 2 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 4 0）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 2 4 が出力モジュールとなるので、画像出力部 2 4 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 3 2 による指定は行われず、必要な場合はモジュール生成部 4 4 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 3 2 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。

【0050】

一方、モジュール生成部 4 4 は、アプリケーション 3 2 によって起動されるとモジュール生成処理を行う。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 3 8 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 3 8 の前段にバッファモジュール 4 0 が存在している場合、当該バッファモジュール 4 0 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 3 8 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。

【0051】

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 3 8 の生成が必要か否か判定する。例えばモジュール生成部 4 4 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 3 2 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 3 8 として RGB→CMY の色空間変換を行う画像処理モジュール 3 8 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 3 8 は生成不要と判断する。

【0052】

生成対象の画像処理モジュール 3 8 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 3 8 の後段にバッファモジュール 4 0 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 2 4）である場合（例えば図 2 (A)～(C)に示す画像処理部 5 0 における最後段の画像処理モジュール 3 8 を参照）や、例として図 2 (B)に示す画像処理部 5 0 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 3 8 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 3 8 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てバッファ制御部 4 0 B を起動する（バッファ制御部 4 0 B のプログラムを実行するスレッドを各々生成する）ことで、画像処理モジュール 3 8 の後段に連結するバッファモジュール 4 0 を生成する。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バッファモジュール 4 0 を生成するようにしてもよい。

【0053】

続いて、前段のモジュール（例えばバッファモジュール 4 0）の情報、後段のバッファモジュール 4 0 の情報（後段にバッファモジュール 4 0 を生成した画像処理モジュール 3 8 のみ）、画像処理モジュール 3 8 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 3 6 に登録されており、画像処理モジュール 3 8 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 3 8 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 3 8 を選択・生成（画像処理エンジン 3 8 A 及び制御部 3 8 B のプログラムを実行するスレッドを生成）する。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画像処理モジュール 3 8 を生成するようにしてもよい。

【0054】

例えばモジュール生成部 4 4 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 C M Y 色空間が指定され、更に入力画像データが R G B 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 3 6 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 3 8の中から、R G B → C M Y の色空間変換を行う画像処理モジュール 3 8 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 3 8 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 3 8 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 3 8 が選択・生成される。

【0055】

なお、画像処理モジュール 3 8 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 3 8 A 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 3 8 をモジュールライブラリ 3 6 に複数登録しておき、画像処理部 5 0 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 3 8 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 3 8 を選択する等）ようにしてもよいし、アプリケーション 3 2 或いはユーザに選択させるようにしてもよい。

【0056】

画像処理モジュール 3 8 の生成が完了すると、後段のバッファモジュール 4 0 の I D と生成した画像処理モジュール 3 8 の I D の組をワークフロー管理部 4 6 A に通知する。この I D 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 4 0 や画像処理モジュール 3 8 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 4 4 が、複数種の画像処理モジュール 3 8 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 3 8 と画像回転処理を行う画像処理モジュール 3 8 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 3 8 を含むモジュール群が生成される。アプリケーション 3 2 によって順次起動された個々のモジュール生成部 4 4 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 2 (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 5 0 が構築される。

【0057】

なお、本実施形態では、特定の画像処理の実行頻度が高い等の場合に、アプリケーション 3 2 が、特定の画像処理を行う画像処理部 5 0 を生成するための複数種のモジュール生成部 4 4 に対し、特定の画像処理を行う画像処理部 5 0 を生成させた後も処理終了を指示しないことでスレッド（プロセス又はオブジェクトでもよい）として残しておき、特定の画像処理を行う必要が生ずる毎に、スレッドとして残しておいた各モジュール生成部 4 4 に対してモジュール群の生成を順次指示することで、特定の画像処理を行う画像処理部 5 0 を再生成させることも可能とされている。これにより、特定の画像処理を行う必要が生

ずる毎に対応する各モジュール生成部 4 4 を各々起動する処理が不要となり、特定の画像処理を行う画像処理部 5 0 を再生成するのに要する時間を短縮することができる。

【0058】

ところで、画像処理モジュール 3 8 の制御部 3 8 B は、モジュール生成部 4 4 によって起動されると画像処理モジュール 3 8 の初期化を行う。この初期化では、まずモジュール生成部 4 4 から与えられた自モジュールの前段及び後段のモジュールの情報を記憶する。続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には、前段のバッファモジュール 4 0 からの 1 回の画像データの読み出しによって取得する画像データのデータ量（単位読出データ量）を認識する。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 2 (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、前段に存在している全てのバッファモジュール 4 0 へ通知することで、前段に存在している全てのバッファモジュール 4 0 に単位読出データ量を設定する（図 3 (A) の(1)も参照）。

【0059】

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 3 (A) の(2)も参照）する。そして、当該画像処理モジュール 3 8 の初期化の完了をモジュール生成部 4 4 に通知して処理を終了する。

【0060】

一方、画像処理部 5 0 を構成する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、モジュール生成部 4 4 又はアプリケーション 3 2 によって起動されるとバッファモジュール 4 0 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）。

【0061】

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 3 8 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 3 8 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 4 0 A 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 3 8 が連結されている場合は、個々の画像処理モジュール 3 8 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。

【0062】

また、自モジュールがアプリケーション32によって生成され、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14又は記憶部20上の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、自モジュールが自モジュールがアプリケーション32によって生成され、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14又は記憶部20上の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は初期化の完了をワークフロー管理部46Aへ通知する。

【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の構築が完了すると、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）を起動することで、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。

【0064】

処理管理部46のワークフロー管理部46Aは、プログラムが起動されることで図9に示す並列制御処理を行う。ワークフロー管理部46Aは並列制御処理において、画像処理部50を構成する個々の画像処理モジュール38に処理要求を入力することで、画像処理部50による画像処理を個々の画像処理モジュール38単位で並列に行わせるが、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によって行われる初期化処理完了以降の処理、個々の画像処理モジュール38の制御部38Bによって行われる画像処理モジュール制御処理について順に説明する。

【0065】

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。前段の画像処理モジュール38からの書込要求を含む何らかの情報がバッファモジュール40に入力された場合（及び、後述するタイマがタイムアウトした場合）は、図4に示す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。なお、以下で説明するデータ書込処理は、関数やメソッドの呼び出しで処理が始まるようにしてもよい。

【0066】

データ書込処理では、まずステップ100において、今回のデータ書込処理の起動要因が、タイマがタイムアウトしたことによる起動か否か判定する。判定が否定された場合はステップ104へ移行するが、上記判定が肯定された場合はステップ102へ移行し、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す。ステップ106では自モジュールのバッファ40Aがアクセス中か否か判定する。バ

ッファ 4 0 A に対してはデータの読み出しも行われるので、当該判定が肯定された場合はステップ 1 0 8 へ移行して今回の処理対象の書込要求情報をワークメモリ等に保管し、次のステップ 1 1 0 で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。

【0 0 6 7】

一方、ステップ 1 0 6 の判定が否定された場合にはステップ 1 1 2 へ移行し、確保すべき記憶リソースのサイズとして単位書込データ量をリソース管理部 4 6 B に通知して、書込用に用いる記憶リソース領域（書込用バッファ領域：図 5 (B) も参照）をリソース管理部 4 6 B によって確保させる。なお、このときメモリ 1 4 の残量があれば、記憶リソース領域としてメモリ 1 4 が確保されるが、メモリ 1 4 の残量が不足している場合は記憶リソース領域として記憶部 2 0 の記憶領域が確保される。次のステップ 1 1 4 では、自モジュールのバッファ 4 0 A 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判定する。

【0 0 6 8】

モジュール生成部 4 4 によって生成されたバッファモジュール 4 0 は、当初はバッファ 4 0 A として用いる記憶リソース領域（単位バッファ領域）が確保されておらず、記憶リソース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 4 0 に最初に書込要求が入力されたときにはバッファ 4 0 A として用いる記憶リソース領域（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 4 0 A 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。

【0 0 6 9】

ステップ 1 1 4 の判定が否定された場合はステップ 1 1 6 へ移行し、確保すべき記憶リソースのサイズ（単位バッファ領域のサイズ）をリソース管理部 4 6 B に通知して、自モジュールのバッファ 4 0 A として用いる記憶リソース領域（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 4 6 B によって確保させた後にステップ 1 1 8 へ移行する。なお、このときもメモリ 1 4 の残量があれば記憶リソース領域としてメモリ 1 4 が確保されるが、メモリ 1 4 の残量が不足している場合は記憶リソース領域として記憶部 2 0 の記憶領域が確保される。また、ステップ 1 1 4 の判定が肯定された場合はステップ 1 1 6 をスキップしてステップ 1 1 8 へ移行する。そしてステップ 1 1 8 では、先のステップ 1 1 2 で確保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 3 8 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 3 8 は、先頭アドレスが通知された書込領域（単位バッファ領域又は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）。

【0 0 7 0】

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 4 0 A（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 4 0 A 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際の記憶リソース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 3 8 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 3 8 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 3 8 とのインタフェースが簡単になる。

【0 0 7 1】

なお、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、すなわちバッファ 4 0 A として用いる記憶リソース領域が既に確保されている場合には、先に説明したステップ 1 1 2 ～ステップ 1 1 6 をスキップしてステップ 1 1 8 へ移行し、既に確保されている記憶リソース領域のアドレスを画像処理モジュール 3 8 に書込領域のアドレスとして通知し、書込領域への画像データの書き込みを行わせる。

【0 0 7 2】

前段の画像処理モジュール 3 8 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ステップ 1 2 0 へ移行し、書込用バッファ領域に書き込まれた画像データを保管用バッファ領域に書き込む。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。次のステップ 1 2 2 では、自モジュールの後段の個々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）。そして、次のステップ 1 2 4 では、先に書込用バッファ領域として確保した記憶リソース領域をリソース管理部 4 6 B によって解放させ、データ書込処理を終了する。なお、書込用バッファ領域はバッファモジュール 4 0 の初期化時に確保し、バッファモジュール 4 0 の消去時に解放するように構成してもよい。

【0 0 7 3】

続いて、後段の画像処理モジュール 3 8 から読出要求がバッファモジュール 4 0 に入力された場合（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 4 0 の読出制御部 4 0 D によって実行されるデータ読出処理について、図 6 を参照して説明する。なお、以下で説明するデータ読出処理についても、関数やメソッドの呼び出しで処理が始まるようにしてもよい。

【0 0 7 4】

データ読出処理では、まずステップ 1 7 0 において、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読出要求を受信したことによる起動か否か判定する。判定が否定された場合はステップ 1 7 4 へ移行するが、上記判定が肯定された場合はステップ 1 7 2 へ移行し、後段の画像処理モジュールから今回受信した読出要求情報を読出用の待ち行列の末尾に登録する。ステップ 1 7 4 では、自モジュールのバッファ 4 0 A がアクセス中か否か判定する。バッファ 4 0 A に対してはデータの書き込みも行われるので、当該判定が肯定された場合はステップ 2 0 8 へ移行し、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する。判定が否定された場合はデータ読出処理を終了するが、判定が肯定された場合はステップ 2 1 0 でタイマをスタートさせてデータ読出処理を一旦終了する。タイマをスタートさせた場合には、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 7 5】

一方、ステップ 1 7 4 の判定が否定された場合にはステップ 1 7 6 へ移行し、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す。次のステップ 1 7 8 では、読出用の待ち行列から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タに基づいて、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。次のステップ 1 8 0 では、ステップ 1 7 8 で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判定する。

【0076】

ステップ180の判定が否定された場合はステップ182へ移行し、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末尾が処理対象の画像データの末尾か否か判定する。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がバッファ40A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40Aに記憶されている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、ステップ180又はステップ182の判定が肯定されてステップ184へ移行する。ステップ184では、確保すべき記憶リソース領域のサイズとして読出要求元の画像処理モジュール38に対応する単位読出データ量をリソース管理部46Bに通知すると共に、読出に用いる記憶リソース領域（読出用バッファ領域：図8(B)も参照）の確保をリソース管理部46Bに要求する。なお、このときメモリ14の残量があれば、記憶リソース領域としてメモリ14が確保されるが、メモリ14の残量が不足している場合は記憶リソース領域として記憶部20の記憶領域が確保される。

【0077】

読出用バッファ領域を確保すると、次のステップ186では、読出対象の有効データをバッファ40Aから単位読出データ量分だけ読み出し、読み出した有効データを読出用バッファ領域に書き込む。次のステップ188では、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38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38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38に通知する。また、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である場合は、バッファ40Aとして用いている記憶リソース領域（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38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。

【0078】

例として図8(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際の記憶リソース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8(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。

【0079】

次のステップ190では、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が完了したか否か判定し、判定が肯定される迄ステップ190を繰り返す。読出要求元の画像処理モジュール38から読出完了が通知されると、ステップ190の判定が肯定されてステップ192へ移行し、読出用バッファ領域として確保した記憶リソース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、当該記憶リソース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。またステップ194では、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8(C)も参照）。

【0080】

ステップ196では、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、ステップ194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ステップ208へ移行し、前述のステップ208、210を経てデータ読出処理を終了するが、判定が肯定された場合はステップ198へ移行し、有効データを記憶していない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ステップ208へ移行し、ステップ208、210を経てデータ読出処理を終了する。

【0081】

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10(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、ステップ180、182の判定が各々否定されてステップ200へ移行し、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へ出力する（図3(B)の(5)も参照）。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。またステップ202では、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、ステップ208、210を経てデータ読出処理を終了する。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（ステップ180又はステップ182の判定が肯定される迄）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。

【0082】

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3(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（図4）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に画像データが書き込まれる（図3(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3(B)の(9)も参照）。

【0083】

上述したデータ書込処理及びデータ読出処理では、一方が自モジュールのバッファ40Aにアクセスしている間、他方がバッファ40Aへのアクセスを停止する排他制御が行われる。これにより、コンピュータ10のCPU12が画像処理部50を構成する個々のモジュールに対応するプロセス又はスレッドを並列に実行しても、単一のバッファモジュール40に複数の要求が同時又は略同時に入力されることによる不都合の発生を回避できるので、コンピュータ10のCPU12が個々のモジュールに対応するプロセス又はスレッドを並列に実行することができる。もちろん、バッファモジュールを通常のプログラムまたはオブジェクトとして実現しても良い。

【0084】

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 8）を説明する。

【0085】

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記憶リソース領域（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 が使用する記憶リソースは、画像処理エンジン 38A が画像処理を行うために必要な記憶リソースが主であるが、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記憶リソース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記憶リソース領域が必要となることもある。そして、認識したサイズの記憶リソース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記憶リソース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）が記憶リソース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。

【0086】

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 で取得した記憶リソース領域のうちデータの一時保管用の記憶リソース領域に書き込むデータ取得処理を行う。

【0087】

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、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、一時保管用の記憶リソース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）。

【0088】

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、一時保管用の記憶リソース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 3 8 によってデータを書き込ませる一時保管用の記憶リソース領域のアドレスを通知して書き込みを要請することで、前段の画像処理モジュール 3 8 から出力されるデータを一時保管用の記憶リソース領域に書き込ませるデータ取得処理を行う。

【0089】

次のステップ 2 2 8 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 2 3 2 へ移行するが、判定が肯定された場合はステップ 2 3 0 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 2 3 0 の判定が否定された場合はステップ 2 2 0 に戻り、ステップ 2 3 0 の判定が肯定される迄ステップ 2 2 0 ～ステップ 2 3 0 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 2 2 8 の判定が否定されるかステップ 2 3 0 の判定が肯定されてステップ 2 3 2 へ移行する。

【0090】

次のステップ 2 3 2 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 2 3 2 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 4 0 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 4 0 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 4 0 であれば当該バッファモジュール 4 0 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 2 3 6 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 2 1 9 で取得した記憶リソース領域のうち画像処理エンジンによる画像処理用の記憶リソース領域（の先頭アドレス及びサイズ）を画像処理エンジン 3 8 A に入力し、入力したデータに対し画像処理用の記憶リソース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 3 8 A への単位読出データ量のデータの inputs が完了し、画像処理エンジン 3 8 A 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 2 3 8 で出力が完了したことを後段のモジュールに通知する。

【0091】

上記のステップ 2 2 0 ～ステップ 2 3 8 により画像処理モジュール 3 8 における単位処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。

【0092】

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。

【0093】

一方、ワークフロー管理部 4 6 A は、アプリケーション 3 2 によって画像処理の実行が指示されると、図 9 (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。

【0094】

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）。

【0095】

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。

【0096】

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 の(6)参照）。ワークフロー管理部 4 6 A は、バッファモジュール 4 0₂ 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 3 8₂ に処理要求を入力し（図 9 の(7)参照）、画像処理モジュール 3 8₃ の制御部 3 8 B は前段のバッファモジュール 4 0₁ 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 4 0₁ 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 0₁ のバッファ制御部 4 0 B もワークフロー管理部 4 6 A 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 4 6 A は、バッファモジュール 4 0₁ 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 3 8₁ に処理要求を入力する（図 9 の(10)参照）。

【0097】

ここで、画像処理モジュール 3 8₁ の前段のモジュールは画像データ供給部 2 2 であるので、画像処理モジュール 3 8₁ の制御部 3 8 B は、画像データ供給部 2 2 にデータ要求を入力することで画像データ供給部 2 2 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 3 8 A 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 4 0₁ のバッファ 4 0 A に書き込む（図 9 の(12)参照）。

【0098】

また、バッファモジュール 4 0₁ のバッファ制御部 4 0 B は、後段の画像処理モジュール

ル 3 8₂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 3 8₂に対して読出を要請し、これに伴い画像処理モジュール 3 8₂の制御部 3 8 B は、バッファモジュール 4 0₁のバッファ 4 0 A 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 3 8 A 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 4 0₂のバッファ 4 0 A に書き込む（図 9 の(14)参照）。バッファモジュール 4 0₂のバッファ制御部 4 0 B は、後段の画像処理モジュール 3 8₃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 3 8₃へ読出を要請し、画像処理モジュール 3 8₃の制御部 3 8 B は、バッファモジュール 4 0₂のバッファ 4 0 A 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 3 8 A 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 4 0₃のバッファ 4 0 A に書き込む（図 9 の(16)参照）。

【0 0 9 9】

更に、バッファモジュール 4 0₃のバッファ制御部 4 0 B は、後段の画像処理モジュール 3 8₄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 3 8₄に対して読出を要請し、これに伴い画像処理モジュール 3 8₄の制御部 3 8 B は、バッファモジュール 4 0₃のバッファ 4 0 A 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 3 8 A 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 2 4 へ出力する（図 9 の(18)参照）。

【0 1 0 0】

また、個々の画像処理モジュール 3 8 の制御部 3 8 B は、後段のバッファモジュール 4 0 のバッファ 4 0 A 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 4 6 A へ処理完了通知を入力する。ワークフロー管理部 4 6 A は、画像処理モジュール 3 8 から処理完了通知が入力される毎に、図 9 (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 5 2 0 で処理完了通知元の画像処理モジュール 3 8 に処理要求を再度入力して処理を終了する。

【0 1 0 1】

このように、ワークフロー管理部 4 6 A による並列制御処理では、任意の画像処理モジュール 3 8 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 3 8 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）を単位として順に引き渡されると共に、個々の画像処理モジュール 3 8 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。また、画像データ供給部 2 2 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 3 8 からワークフロー管理部 4 6 A 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 3 8 から順次行われる。

【0 1 0 2】

ワークフロー管理部 4 6 A は、画像処理モジュール 3 8 から全体処理終了通知が入力される毎に、図 9 (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 5 4 0 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 3 8 が最後段の画像処理モジュール 3 8 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 2 4 へ全て出力されることで、最後段の画像処理モジュール 3 8 から全体処理終了通知が入力された場合には、ステップ 5 4 0 の判定が肯定されてステップ 5 4 2 へ移行し、アプリケーション 3 2 に対して画像処理の完了を通知し、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 3 2 は、ユーザに対して画像処理の完了を通知する。

【0 1 0 3】

また、画像処理部 5 0 が画像処理を行っている間、画像処理部 5 0 を構成する個々の画

像処理モジュール 38 及び個々のバッファモジュール 40（に対応するスレッド）は、先にも説明したようにリソース管理部 46B を介して記憶リソースの取得及び解放を繰り返す。ここで、各モジュールに割り当てる記憶リソースはアクセス速度が高速なメモリ 14 であることが望ましいが、或るモジュールに対応するスレッドがリソース管理部 46B に対して記憶リソースの取得を要求したものの、メモリ 14 の残量が不足していることでリソース管理部 46B からエラーが返ってきた場合、当該モジュールが以降の処理を正常に行えない状態に陥るので、画像処理部 50 における処理全体が失敗に終わってしまうことになる。このため、本実施形態に係るリソース管理部 46B は、メモリ 14 の残量不足が直ちに画像処理部 50 における処理全体の失敗を引き起こすことを回避するため、以下で説明する処理を行う。

【0104】

すなわち、リソース管理部 46B は、任意のモジュール（画像処理モジュール 38 又はバッファモジュール 40）に対応する任意のスレッドから記憶リソース取得要求が入力される毎に、図 11 に示す記憶リソース取得要求時処理を行う。この記憶リソース取得要求時処理では、まずステップ 600 では、記憶リソース取得要求元が画像処理モジュール 38 に対応するスレッドか否か判定する。本実施形態に係るリソース管理部 46B は、画像処理モジュール 38 に割り当てるためのメモリ領域をオペレーティングシステム 30 を通じて予め確保しており、上記判定が肯定された場合はステップ 602 へ移行し、画像処理モジュール 38 用に予め確保したメモリ領域から、記憶リソース取得要求元より要求されているサイズのメモリ領域を確保し、次のステップ 604 において、確保したメモリ領域の先頭アドレスを通知することで、確保したメモリ領域を記憶リソース取得要求元の画像処理モジュール 38 へ引き渡す。このように、本実施形態では画像処理モジュール 38 に対しては記憶リソースとしてアクセス速度が高速なメモリ 14 を常に割り当てるので、アクセス速度がより低い記憶リソースを割り当てることで、画像処理モジュール 38 の処理速度の低下、ひいては画像処理部 50 における画像処理全体の処理速度の低下が生ずることを防止することができる。

【0105】

また、記憶リソース取得要求元がバッファモジュール 40 に対応するスレッドであった場合には、ステップ 600 の判定が否定されてステップ 606 へ移行し、メモリ 14 のうち使用されていない領域（画像処理モジュール用として予め確保した領域を除く）のサイズ（オペレーティングシステム 30 を通じて確保可能なメモリ 14 の残量）をオペレーティングシステム 30 に問い合わせることでメモリ 14 の残量を検知し、検知したメモリ 14 の残量が記憶リソース取得要求元より要求されている記憶リソースのサイズ以上か否か判定する。判定が肯定された場合はステップ 608 へ移行し、オペレーティングシステム 30 を通じて要求サイズのメモリ領域を確保する。またステップ 610 では、バッファモジュール 40 に割り当てたメモリ領域を管理するためにワークメモリ等に設けたメモリ管理テーブルに、記憶リソース取得要求元のバッファモジュール 40 の ID、確保したメモリ領域の先頭アドレス及びサイズ等の情報を登録する。そしてステップ 612 では、確保したメモリ領域の先頭アドレスを記憶リソース取得要求元のバッファモジュール 40 に対応するスレッドへ通知することで、確保したメモリ領域を記憶リソース取得要求元のバッファモジュール 40 へ引き渡し、記憶リソース取得要求時処理を終了する。

【0106】

一方、記憶リソース取得要求元がバッファモジュール 40 に対応するスレッドであり、メモリ 14 の残量が要求サイズよりも小さい場合には、ステップ 606 の判定が否定されてステップ 614 へ移行し、記憶部 20 の記憶領域を要求サイズ分確保する。次のステップ 616 では、バッファモジュール 40 に割り当てた記憶部 20 の記憶領域を管理するためにワークメモリ等に設けた記憶部管理テーブルに、記憶リソース取得要求元のバッファモジュール 40 の ID、確保した記憶部 20 の記憶領域の先頭アドレス及びサイズ、テーブルへの登録日時等の情報を登録する。そしてステップ 618 では、確保した記憶部 20 の記憶領域の先頭アドレスを記憶リソース取得要求元のバッファモジュール 40 に対応す

るスレッドへ通知することで、確保した記憶部 2 0 の記憶領域を記憶リソース取得要求元のバッファモジュール 4 0 に対応するスレッドへ引き渡し、記憶リソース取得要求時処理を終了する。

【0107】

この場合、記憶部 2 0 はメモリ 1 4 に比べるとアクセス速度が低速であるので、バッファモジュール 4 0 へ引き渡された記憶部 2 0 の記憶領域に対するアクセス（画像データの書き込みや読み出し）は低速化することになるが、画像処理モジュール 3 8 が記憶リソースを使って画像処理を行う場合と比較すると記憶リソースへのアクセス頻度は低いので、アクセス速度が低速の記憶リソースが割り当てられたことに伴う処理速度の低下の度合は画像処理モジュール 3 8 よりも小さくて済む。そして、メモリ 1 4 の残量が要求サイズよりも小さいときにも記憶リソース（記憶部 2 0 の記憶領域）が確保されて要求元のバッファモジュール 4 0 に引き渡されるので、メモリ 1 4 の残量不足に拘わらず画像処理部 5 0 における処理全体を継続させることができる。

【0108】

また、リソース管理部 4 6 B は、任意のモジュール（画像処理モジュール 3 8 又はバッファモジュール 4 0）に対応する任意のスレッドから記憶リソース解放要求が入力される毎に、図 1 2 に示す記憶リソース解放要求時処理を行う。この記憶リソース解放要求時処理では、まずステップ 6 3 0 において、記憶リソース解放要求元が画像処理モジュール 3 8 に対応するスレッドか否か判定する。判定が肯定された場合、解放対象の記憶リソースは画像処理モジュール 3 8 に割り当てるために予め確保したメモリ領域であるので、ステップ 6 3 2 へ移行し、解放が要求されたメモリ領域をオペレーティングシステム 3 0 を通じて解放し処理を終了する。

【0109】

また、記憶リソース解放要求元がバッファモジュール 4 0 に対応するスレッドである場合には、ステップ 6 3 0 の判定が否定されてステップ 6 3 4 へ移行し、解放対象の記憶リソースがメモリ領域か否か判定する。解放対象の記憶リソースが記憶部 2 0 の記憶領域である場合には、判定が否定されてステップ 6 3 6 へ移行し、解放が要求された記憶部 2 0 の記憶領域を解放する。またステップ 6 3 8 では、記憶部管理テーブルに登録されている情報のうち、今回解放した記憶領域に対応する情報を記憶部管理テーブルから削除し、処理を終了する。

【0110】

一方、記憶リソース解放要求元がバッファモジュール 4 0 に対応するスレッドであり、かつ解放対象の記憶リソースがメモリ領域である場合には、ステップ 6 3 4 の判定が肯定されてステップ 6 4 0 へ移行し、解放が要求されたメモリ領域をオペレーティングシステム 3 0 を通じて解放する。またステップ 6 4 2 では、メモリ管理テーブルに登録されている情報のうち、今回解放したメモリ領域に対応する情報をメモリ管理テーブルから削除する。次のステップ 6 4 4 では、記憶部管理テーブルに情報が登録されているか否か、すなわち記憶リソースとして記憶部 2 0 の記憶領域が割り当てられているバッファモジュール 4 0 が存在しているか否か判定する。判定が否定された場合は処理を終了するが、判定が肯定された場合はステップ 6 4 6 へ移行し、記憶部管理テーブルに登録されている情報のうち、テーブルへの登録日時が最も古い情報を参照することで、記憶リソースとして記憶部 2 0 の記憶領域が割り当てられているバッファモジュール 4 0 のうち、記憶部 2 0 の記憶領域が割り当てられた時期が最も古いバッファモジュール 4 0 を認識する。

【0111】

次のステップ 6 4 8 では、メモリ 1 4 の残量が、ステップ 6 4 6 で認識したバッファモジュール 4 0 による要求サイズ（当該バッファモジュール 4 0 に割り当てている記憶部 2 0 の記憶領域のサイズ）以上か否か判定する。判定が否定された場合は認識したバッファモジュール 4 0 にメモリ領域を割り当てできない状況ではないので処理を終了するが、ステップ 6 4 8 の判定が肯定された場合には認識したバッファモジュール 4 0 にメモリ領域を割当可能であるので、次のステップ 6 5 0 以降で、認識したバッファモジュール 4 0 に対

し、割り当て済みの記憶部 2 0 の記憶領域に代えてメモリ領域を割り当てる処理を行う。

【0 1 1 2】

すなわちステップ 6 5 0 では認識したバッファモジュール 4 0 による要求サイズのメモリ領域をオペレーティングシステム 3 0 を通じて確保する。次のステップ 6 5 2 では、認識したバッファモジュール 4 0 に割り当て済みの記憶部 2 0 の記憶領域に書き込まれているデータを、ステップ 6 5 0 で確保したメモリ領域に複写する。また、ステップ 6 5 4 ではメモリ管理テーブルに、認識したバッファモジュール 4 0 の ID、確保したメモリ領域の先頭アドレス及びサイズ等の情報を登録する。次のステップ 6 5 6 では、認識したバッファモジュール 4 0 に対し、データを複写したメモリ領域の先頭アドレスを通知することで、割り当てていた記憶部 2 0 の記憶領域に代えてデータを複写したメモリ領域の割り当て（引き渡し）を行う。また、ステップ 6 5 8 では、新たにメモリ領域を割り当てたバッファモジュール 4 0 に割り当てていた記憶部 2 0 の記憶領域を解放し、次のステップ 6 6 0 では、記憶部管理テーブルに登録されている情報のうち、今回解放した記憶領域に対応する情報を記憶部管理テーブルから削除する。

【0 1 1 3】

そしてステップ 6 6 0 の処理を行うとステップ 6 4 4 に戻る。従って、記憶リソース解放要求時処理では、バッファモジュール 4 0 に割り当てていたメモリ領域を解放することで、メモリ 1 4 の残量が増大したタイミングで、記憶リソースとして記憶部 2 0 の記憶領域を割り当てていたバッファモジュール 4 0 に対し、記憶リソースとしてメモリ領域を割り当てし直すことが可能か否かが、記憶部 2 0 の記憶領域が割り当てられた時期が古い順に判定され、メモリ領域が割当可能であれば記憶部 2 0 の記憶領域に代えてメモリ領域が割り当てされる。これにより、バッファモジュール 4 0 に記憶リソースとして記憶部 2 0 の記憶領域を割り当てることで、当該バッファモジュール 4 0 の処理速度が低下していたとしても、メモリ 1 4 の残量が増大した際に記憶リソースとしてメモリ領域が割り当てし直されることで、上記バッファモジュール 4 0 の処理速度が低下している状態が長時間継続することを防止することができ、メモリ 1 4 の一時的な残量不足が画像処理部 5 0 における処理全体の処理速度に継続的に悪影響を及ぼすことを防止することができる。

【0 1 1 4】

また、記憶リソースとしてメモリ領域を割り当てし直すことが可能か否かの判定を、記憶部 2 0 の記憶領域が割り当てられた時期が古い順に行うことで、記憶リソースとして記憶部 2 0 の記憶領域が一旦割り当てられてから、記憶リソースとしてメモリ領域が割り当てし直される迄の所要時間を、個々のバッファモジュール 4 0 について平均化することができる。

【0 1 1 5】

〔第 2 実施形態〕

次に本発明の第 2 実施形態を説明する。なお、本第 2 実施形態は第 1 実施形態と同一構成であるので、各部分に同一の符号を付して構成の説明を省略し、以下、本第 2 実施形態の作用を説明する。

【0 1 1 6】

第 1 実施形態で説明したように、画像処理部 5 0 は、画像処理モジュール 3 8 及びバッファモジュール 4 0 がパイプライン形態又は有向非循環グラフ形態で連結されて構築されており、個々の画像処理モジュール 3 8 は、前段に連結されたバッファモジュール 4 0 に単位読出データ量以上の画像データが蓄積されないと自モジュールにおける画像処理を開始できない（但し、画像データ供給部 2 2 に連結された最前段の画像処理モジュール 3 8 を除く）ので、個々の画像処理モジュール 3 8 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 3 8 における画像処理の進行状況に依存し、特に画像処理部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。

【0 1 1 7】

また、画像処理モジュール 3 8 及びバッファモジュール 4 0 がパイプライン形態又は有向非循環グラフ形態で連結された構成では、前段側の画像処理モジュール 3 8 よりも後段側の画像処理モジュール 3 8 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 3 8 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 3 8 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。

【0 1 1 8】

上記に基づき、本第 2 実施形態に係るワークフロー管理部 4 6 A は、図 1 3 に示す並列制御処理を行う。すなわち、アプリケーション 3 2 によって画像処理の実行が指示された際に実行する並列制御処理 1（図 9（A）参照）では、ステップ 5 0 1 において、個々の画像処理モジュール 3 8 のプログラムを実行する個々のスレッドの実行優先度が、例として図 1 4（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 3 8 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。

【0 1 1 9】

なお、上記の「画像処理モジュール 3 8 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 1 5（A）に示すように先頭（最前段）の画像処理モジュール 3 8 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 3 8 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 1 5（B）に示すように先頭（最前段）の画像処理モジュール 3 8 から昇順に（或いは最後尾（最後段）の画像処理モジュール 3 8 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 3 8（図 1 5（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。

【0 1 2 0】

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 3 8 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 3 8 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 3 8 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 3 8 の位置が前段側のスレッド程、高い実行優先度で CPU 1 2 によって実行されることになり、CPU 1 2 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。

【0 1 2 1】

また、次のステップ 5 0 2 では、画像処理部 5 0 を構成する個々のバッファモジュール 4 0 の優先度を、個々のバッファモジュール 4 0 と直接連結された画像処理モジュール 3 8 に対応するスレッドの実行優先度に応じて設定する。個々のバッファモジュール 4 0 の優先度は、具体的には、例えば個々のバッファモジュール 4 0 の前段側に存在している画像処理モジュール 3 8 に対応するスレッドの実行優先度と一致させてもよいし、個々のバッファモジュール 4 0 の後段側に存在している画像処理モジュール 3 8 に対応するスレッドの実行優先度と一致させてもよいし、個々のバッファモジュール 4 0 の前段側及び後段

側に存在している複数の画像処理モジュール 3 8 に対応するスレッドの実行優先度の平均値、最大値、最小値の何れかをを用いてもよい。

【0 1 2 2】

なお、上述したバッファモジュール 4 0 の優先度は、後述する記憶リソース取得要求時処理及び記憶リソース解放要求時処理で用いる情報であり、バッファモジュール 4 0 に対応するスレッドの実行優先度とは無関係な情報であるが、これに代えてバッファモジュール 4 0 に対応するスレッドの実行優先度を上記のように設定すると共に、後述する記憶リソース取得要求時処理及び記憶リソース解放要求時処理においても、バッファモジュール 4 0 に対応するスレッドの実行優先度を用いるようにしてもよい。

【0 1 2 3】

また、本第 2 実施形態に係るワークフロー管理部 4 6 A は、画像処理モジュール 3 8 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 9（C）参照）において、ステップ 5 2 2 で画像処理部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、例えば個々の画像処理モジュール 3 8 からワークフロー管理部 4 6 A 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 3 8 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 3 8 を構成すると共に、ワークフロー管理部 4 6 A は、個々の画像処理モジュール 3 8 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 3 8 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 3 8 に対応する進行度合情報から画像処理部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。

【0 1 2 4】

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 3 8（に対応するスレッドを実行する CPU 1 2）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 3 8 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 3 8 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 3 8 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 4 6 A で演算するようにしてもよい。

【0 1 2 5】

次のステップ 5 2 4 では、ステップ 5 2 2 で判定した画像処理部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 3 8 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 CPU 1 2 に余分な負荷が加わることを回避するために、ステップ 5 2 4 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 1 0 % 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。

【0 1 2 6】

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 5 2 6 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 3 8 に対応するスレッドの実行優先度を変更設定する。

【0 1 2 7】

なお、このステップ 5 2 6 における実行優先度の変更は、画像処理モジュール 3 8 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 1 4 (B), (C) に示すように、画像処理部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 3 8 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 3 8 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 1 4 (D), (E) に示すように、画像処理部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 3 8 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 3 8 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、CPU 1 2 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。

【0 1 2 8】

また、次のステップ 5 2 8 では、変更設定した個々の画像処理モジュール 3 8 に対応するスレッドの実行優先度に応じて、並列制御処理 1 のステップ 5 0 2 と同様に、個々のバッファモジュール 4 0 の優先度を変更設定し、並列制御処理 3 を終了する。なお、個々の画像処理モジュール 3 8 に対応するスレッドの実行優先度及び個々のバッファモジュール 4 0 の優先度を上記のように初期設定及び変更することは、請求項 7 に記載の実行優先度制御手段に対応している。個々の画像処理モジュール 3 8 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ことで、画像処理部 5 0 における画像処理全体の処理効率を向上させることができる。

【0 1 2 9】

次に、本第 2 実施形態に係る記憶リソース取得要求時処理について、図 1 6 を参照し、第 1 実施形態で説明した記憶リソース取得要求時処理（図 1 1）と異なる部分についてのみ説明する。第 2 実施形態に係る記憶リソース取得要求時処理では、記憶リソース取得要求元がバッファモジュール 4 0 に対応するスレッド（ステップ 6 0 0 の判定が否定）であり、メモリ 1 4 の残量が要求サイズよりも小さい（ステップ 6 0 0 の判定が肯定の）場合に、ステップ 6 2 0 へ移行し、記憶リソース取得要求元のバッファモジュール 4 0 の優先度をワークフロー管理部 4 6 A 等から取得する。

【0 1 3 0】

また、ステップ 6 2 1 では、メモリ管理テーブルに情報が登録されている個々のバッファモジュール 4 0（記憶リソースとしてメモリ領域が割り当てられているバッファモジュール 4 0）の優先度をワークフロー管理部 4 6 A 等から取得し、取得した個々のバッファモジュール 4 0 の優先度をステップ 6 2 0 で取得した記憶リソース取得要求元のバッファモジュール 4 0 の優先度と比較する。そしてステップ 6 2 2 では、ステップ 6 2 1 における優先度の比較結果に基づいて、優先度が記憶リソース取得要求元のバッファモジュール 4 0 よりも低く、かつ割り当てされているメモリ領域のサイズが要求サイズ以上のバッファモジュール 4 0 が有るか否かを判定する。判定が否定された場合は記憶リソース取得要求元のバッファモジュール 4 0 にメモリ領域を割り当てできないので、要求サイズ分の記憶部 2 0 の記憶領域の確保（ステップ 6 1 4）、記憶部管理テーブルへの情報の登録（ステップ 6 1 6）、記憶リソース取得要求元のバッファモジュール 4 0 への確保した記憶部 2 0 の記憶領域の引き渡し（ステップ 6 1 8）を順に行う。

【0 1 3 1】

一方、ステップ 6 2 2 の条件に該当するバッファモジュール 4 0 が存在している場合には、ステップ 6 2 2 からステップ 6 2 3 へ移行し、記憶部 2 0 の記憶領域を、上記条件に該当するバッファモジュール 4 0 に割り当てられているメモリ領域と同サイズ分確保し、次のステップ 6 2 4 では、上記条件に該当するバッファモジュール 4 0 に割り当てられているメモリ領域から確保した記憶部 2 0 の記憶領域へデータを複写する。またステップ 6 2 5 では、上記条件に該当するバッファモジュール 4 0 の ID、確保した記憶部 2 0 の記憶領域の先頭アドレス及びサイズ等の情報を記憶部管理テーブルに登録する。そしてステップ 6 2 6 では、上記条件に該当するバッファモジュール 4 0（記憶リソースとしてメモリ領域が割り当てられていたバッファモジュール 4 0）に対し、今まで割り当てられていたメモリ領域に代えて、データを複写した記憶部 2 0 の記憶領域の引き渡し（割り当てし

直し)を行う。

【0132】

また、ステップ627では上記のバッファモジュール40に割り当てられていたメモリ領域をクリアし、次のステップ628では、メモリ管理テーブルに登録されている上記メモリ領域の情報を、バッファモジュール40のIDとして記憶リソース取得要求元のバッファモジュールのIDに書替えると共に、割当サイズの変動があれば（要求サイズが上記のバッファモジュールに割り当てられていたメモリ領域のサイズよりも小さい場合）サイズも書替えることで更新する。そしてステップ629では、記憶リソース取得要求元のバッファモジュール40に対し、他のバッファモジュール40に割り当てられていたメモリ領域の先頭アドレスを通知することで、メモリ領域の割り当て（引き渡し）を行う。なお、割当サイズの変動がある場合には、不要となった一部のメモリ領域を解放する処理も行う。

【0133】

このように、本第2実施形態に係る記憶リソース取得要求時処理では、バッファモジュール40から記憶リソースの取得が要求されたものの、メモリ14の残量が不足している場合に、既にメモリ領域を割り当てたバッファモジュール40の中に、記憶リソース取得要求元のバッファモジュール40よりも優先度の低いバッファモジュール40が存在しており、かつ当該優先度の低いバッファモジュール40に割り当てられているメモリ領域のサイズが要求サイズ以上であれば、当該優先度の低いバッファモジュール40に対してはメモリ領域に代えて記憶部20の記憶領域を割り当てし直すと共に、優先度の低いバッファモジュール40に割り当てられていたメモリ領域を記憶リソース取得要求元のバッファモジュール40に割り当てるので、優先度の高いバッファモジュール40に優先的にメモリ領域が割り当てされることになり、画像処理部50における画像処理全体の処理効率を更に向上させることができる。

【0134】

続いて、本第2実施形態に係る記憶リソース解放要求時処理（図17）について説明する。第1実施形態で説明した記憶リソース解放要求時処理（図12）では、ステップ646において、記憶リソースとして記憶部20の記憶領域が割り当てられているバッファモジュール40のうち、記憶部20の記憶領域が割り当てられた時期が最も古いバッファモジュール40を認識することで、記憶リソースとしてメモリ領域を割り当てし直すことが可能か否かの判定を、記憶部20の記憶領域が割り当てられた時期が古い順に行っていたが、本第2実施形態に係る記憶リソース解放要求時処理では、上記のステップ646に代えて、ステップ647において、記憶リソースとして記憶部20の記憶領域が割り当てられているバッファモジュール40のうち、優先度が最も高いバッファモジュール40を認識している。これにより、記憶リソースとして記憶部20の記憶領域を一旦割り当てたバッファモジュール40に対してメモリ領域を割り当てし直すことが可能か否かの判定が、優先度の降順に行われることになり、画像処理部50における画像処理全体の処理効率が更に向上する確率を増大させることができる。

【0135】

なお、第2実施形態において、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、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。

【0136】

具体的には、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は、後段のバッ

ファモジュール 4 0 を介して後段に連結された画像処理モジュール 3 8 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 3 8 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、このような画像処理モジュール 3 8 に対応するスレッドの実行優先度を増大させる。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 3 8 は、後段のバッファモジュール 4 0 を介して後段に連結された画像処理モジュール 3 8 における「待ち」の発生回数が比較的小さいので、当該画像処理モジュール 3 8 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 3 8 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、このような画像処理モジュール 3 8 に対応するスレッドの実行優先度を低下させる。

【0137】

これにより、個々の画像処理モジュール 3 8 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 3 8 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、CPU 1 2 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 5 0 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。

【0138】

なお、上記態様において、後段のバッファモジュール 4 0 からデータ要求が入力された回数、すなわち後段のバッファモジュール 4 0 を介して後段に連結された画像処理モジュール 3 8 で「待ち」が発生した回数に、個々の画像処理モジュール 3 8 が後段のバッファモジュール 4 0 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 4 0 の有効データが、後段の画像処理モジュール 3 8 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 3 8 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。

【0139】

また、上記態様において、後段のバッファモジュール 4 0 を介して連結された後段の画像処理モジュール 3 8 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 3 8 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 3 8 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 3 8 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。

【0140】

また、個々の画像処理モジュール 3 8 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 4 0 の後段の画像処理モジュール 3 8 が、個々のバッファモジュール 4 0 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 4 0 に蓄積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。

【0141】

具体的には、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 4 0 は、後段の画像処理モジュール 3 8 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 3 8 で比較的多数回の「待ち」が発生している可能性が高く、また当該バッファモジュール 4 0 の前段の画像処理モジュール 3 8 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、このような画像処理モジュール 3 8 に対応するスレッドの実行優先度を増大させる。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 4 0 には、後段の画像処理モジュール 3 8 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 3 8 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 4 0 の前段の画像処理モジュール 3 8 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、このような画像処理モジュール 3 8 に対応するスレッドの実行優先度を低下させる。

【0 1 4 2】

これにより、個々の画像処理モジュール 3 8 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 4 0 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、CPU 1 2 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 5 0 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。

【0 1 4 3】

また、第 1 実施形態及び第 2 実施形態では、本発明に係る記憶リソース管理手段として機能する処理管理部 4 6 のリソース管理部 4 6 B が、画像処理部 5 0 の各モジュールからの記憶リソース取得／解放要求を一括して処理する態様を説明したが、本発明はこれに限定されるものではなく、本発明に係る記憶リソース管理手段として機能するリソース管理部を個々のバッファモジュール 4 0 内及び個々の画像処理モジュール 3 8 内に各々設け、個々のモジュール内のリソース管理部が、同一モジュール内の制御部（バッファ制御部 4 0 B 又は制御部 3 8 B）から出力される記憶リソース取得／解放要求を処理するようにしてもよい。この場合、個々のモジュールに所定サイズのメモリ領域を予め割り当てておき、個々のモジュールに設けたリソース管理部は、自モジュールに予め割り当てられたメモリ領域の残量が十分に有る間は、同一モジュール内の制御部からの記憶リソース取得要求に対してメモリ領域を割り当て、自モジュールに予め割り当てられたメモリ領域の残量が不足した場合には、同一モジュール内の制御部からの記憶リソース取得要求に対して記憶部 2 0 の記憶領域を割り当てるように構成することができる。また、この態様において、画像処理モジュール 3 8 でメモリ領域の残量不足が生ずることで、画像処理部 5 0 における画像処理の処理速度が大幅に低下することを回避するために、画像処理モジュール 3 8 に予め割り当てられるメモリ領域を、画像処理の実行中にメモリ領域の残量不足が生じないように十分大きなサイズとする一方で、メモリ 1 4 の節減のために、バッファモジュール 4 0 に予め割り当てられるメモリ領域については、画像処理の実行中にメモリ領域の残量不足が生ずる可能性もある小さなサイズとすることが望ましい。上記態様は請求項 3 及び請求項 8 記載の発明に対応している。

【0 1 4 4】

また、上記ではリソース管理部 4 6 B によるメモリ管理の方式として、画像処理部 5 0 の個々のモジュールからの要求の都度、メモリ 1 4 から要求元のモジュールに割り当てるメモリ領域をオペレーティングシステム 3 0 を通じて確保する管理方式を前提に説明したが、これに限定されるものではなく、メモリ 1 4 から一定サイズのメモリ領域をオペレーティングシステム 3 0 を通じて事前に確保し、個々のモジュールからの要求があると、要求されたメモリ領域のサイズが閾値未満であれば事前に確保したメモリ領域のうちの一部領域を要求元のモジュールに割り当て、要求されたメモリ領域のサイズが閾値以上であれば要求元のモジュールに割り当てるメモリ領域をオペレーティングシステム 3 0 を通じて確保する管理方式に本発明を適用することも可能である。

【0 1 4 5】

更に、上記では画像処理部 5 0 を並列処理方式で動作させるに際し、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）を各々別のスレッドとして実行させる態様を説明したが、これに限定されるものではなく、画像処理モジュール 3 8 が前段のバッファモジュール 4 0 から画像データを読み出す際及び後段のバッファモジュール 4 0 に画像データ等を書き込む際には、画像処理モジュール 3 8 における画像処理等が一時的に停止している状態であることを考慮すると、個々のバッファモジュール 4 0 のバッファ制御部に

相当するプログラムを、データ書込処理（図 4）を行う書込制御部のプログラムと、データ読出処理（図 6）を行う読出制御部のプログラムに分離し、画像処理モジュール 3 8 のプログラムと、前記画像処理モジュール 3 8 の前段のバッファモジュール 4 0 の読出制御部のプログラムと、前記画像処理モジュール 3 8 の後段のバッファモジュール 4 0 の書込制御部のプログラムを単一のスレッドとして実行させ、画像処理部 5 0 を構成する個々の画像処理モジュール 3 8 について上記のスレッドを生成し、並列に実行させることで画像処理部 5 0 を並列処理方式で動作させるようにしてもよい。

【0 1 4 6】

また、上記では本発明に係る外部記憶装置として、HDDやフラッシュメモリ等から成る記憶部 2 0 を例に説明したが、これに限定されるものではなく、通信回線を介してコンピュータ 1 0 に接続された各種の外部記憶デバイス等を適用することも可能である。また、上記では本発明に係るメモリとしてDRAMやSRAM等から成るメモリ 1 4 を例に説明したが、本発明に係るメモリは外部記憶装置と比較して相対的にアクセス速度が高速の記憶デバイスであればよく、例えば通信回線を介してコンピュータ 1 0 に接続された各種の外部記憶デバイスを本発明に係る外部記憶装置として用いる場合には、本発明に係るメモリとしてHDDやフラッシュメモリ等を適用することも可能である。

【図面の簡単な説明】

【0 1 4 7】

【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。

【図 2】画像処理部の構成例を示すブロック図である。

【図 3】(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 4】バッファモジュールで実行されるデータ書込処理の内容を示すフローチャートである。

【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】バッファモジュールで実行されるデータ読出処理の内容を示すフローチャートである。

【図 7】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 8】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。

【図 9】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 1 0】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 1 1】リソース管理部によって実行される記憶リソース取得要求時処理の内容を示すフローチャートである。

【図 1 2】リソース管理部によって実行される記憶リソース解放要求時処理の内容を示すフローチャートである。

【図 1 3】第 2 実施形態に係る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 1 4】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。

【図 1 5】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 1 6】第 2 実施形態に係るリソース管理部によって実行される記憶リソース取得要求時処理の内容を示すフローチャートである。

【図 1 7】第 2 実施形態に係るリソース管理部によって実行される記憶リソース解放要求時処理の内容を示すフローチャートである。

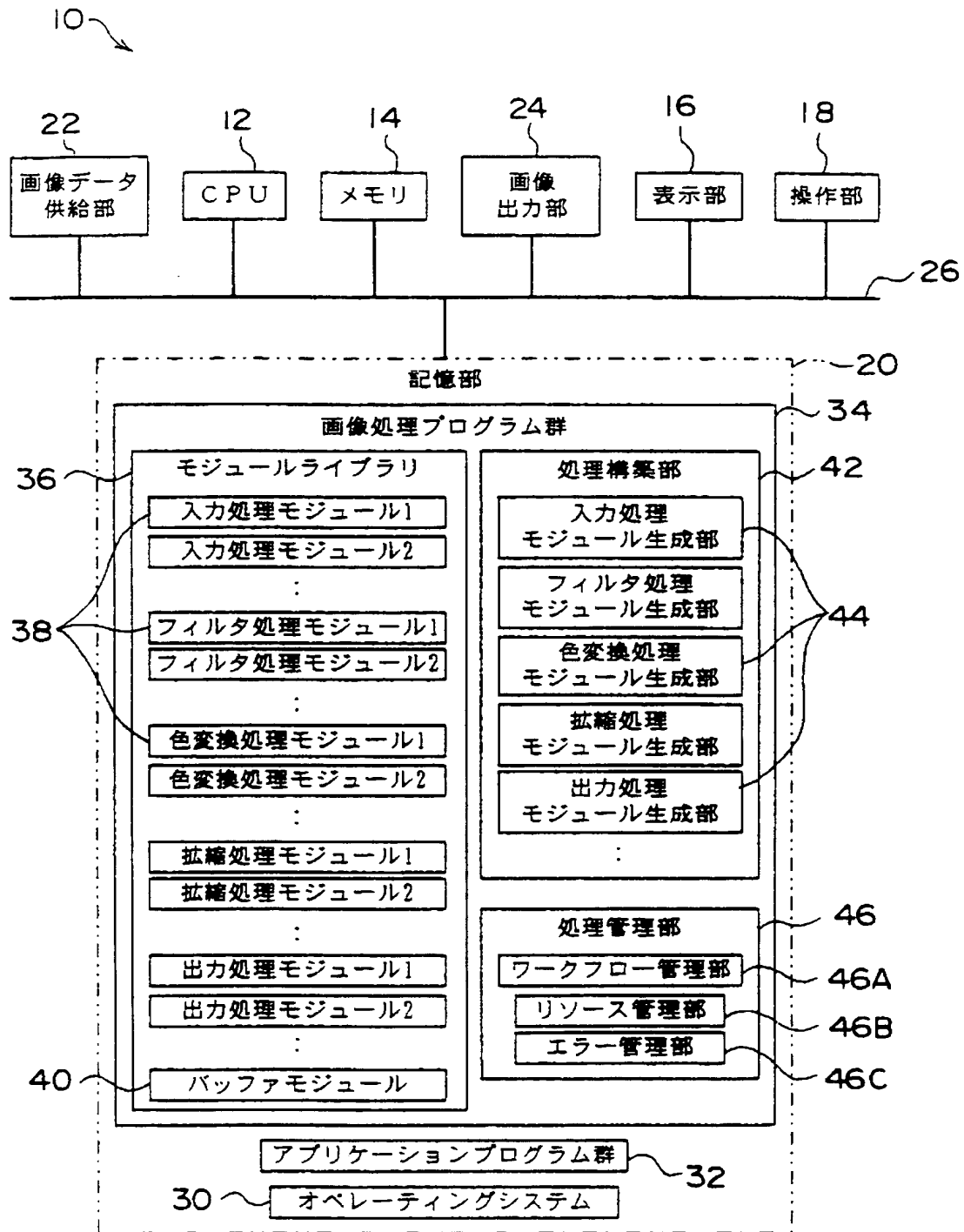
【符号の説明】

【 0 1 4 8 】

- 1 0 コンピュータ
- 1 2 CPU
- 1 4 メモリ
- 2 0 記憶部
- 3 8 画像処理モジュール
- 3 8 A 画像処理エンジン
- 3 8 B 制御部
- 4 0 バッファモジュール
- 4 0 B バッファ制御部
- 4 6 処理管理部
- 4 6 A ワークフロー管理部
- 4 6 B リソース管理部
- 5 0 画像処理部

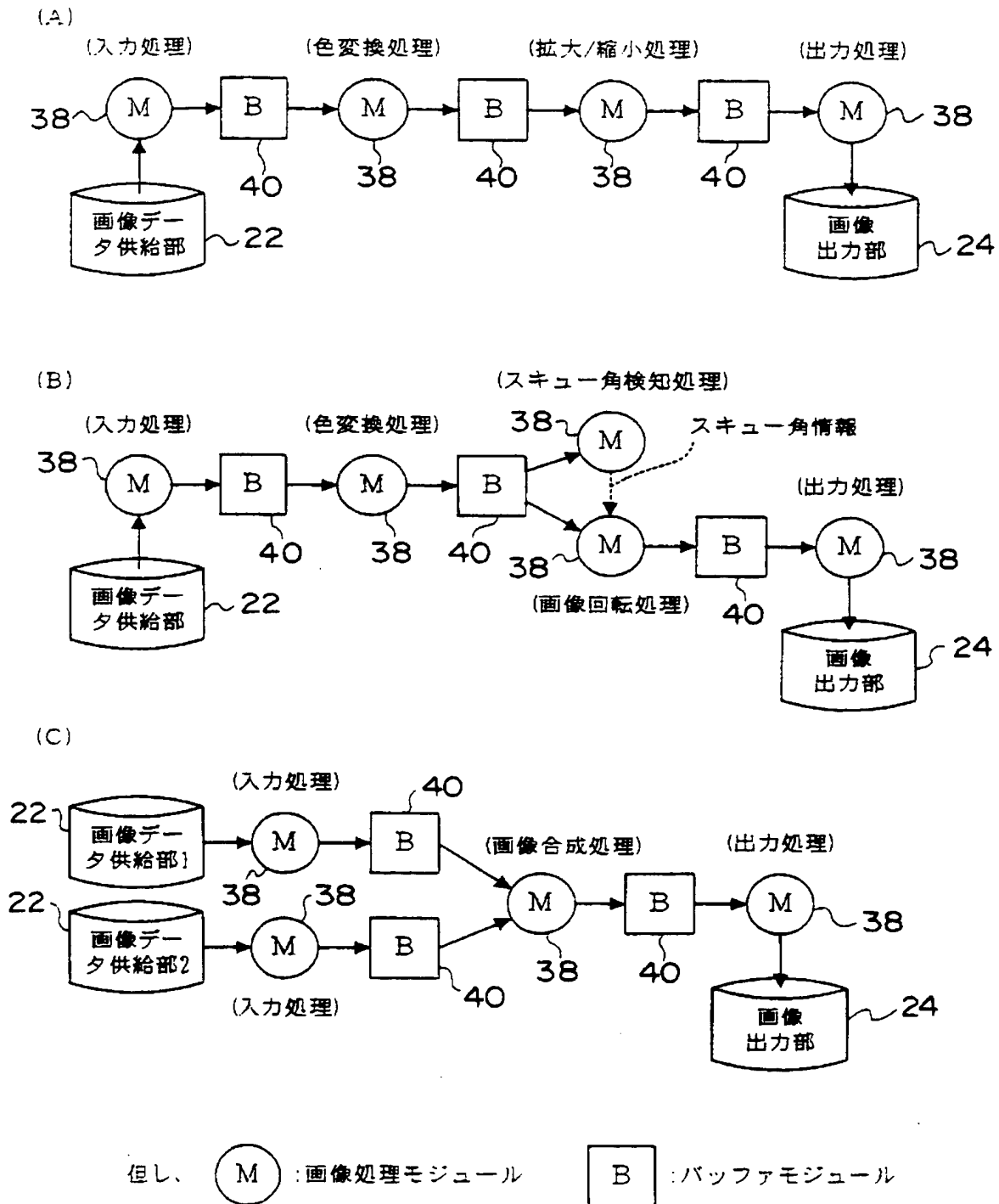
【書類名】 図面

【図 1】



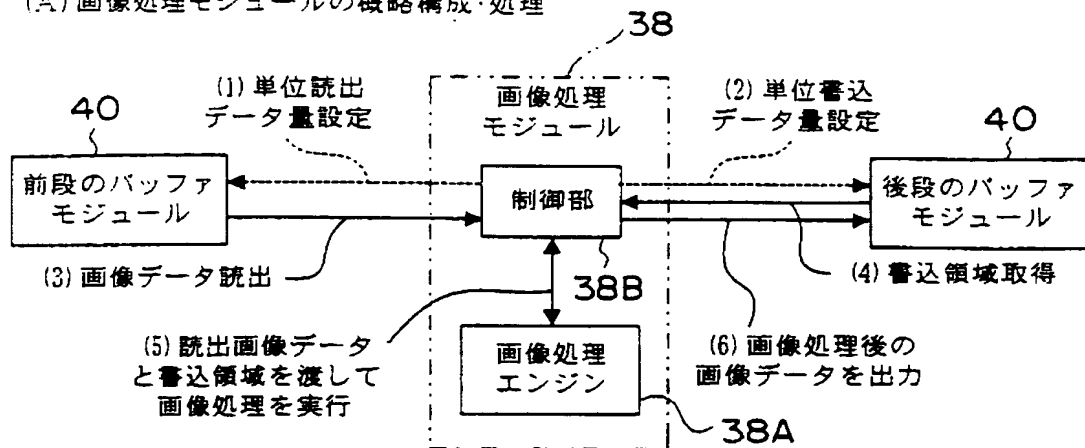
【図 2】

画像処理部の構成例



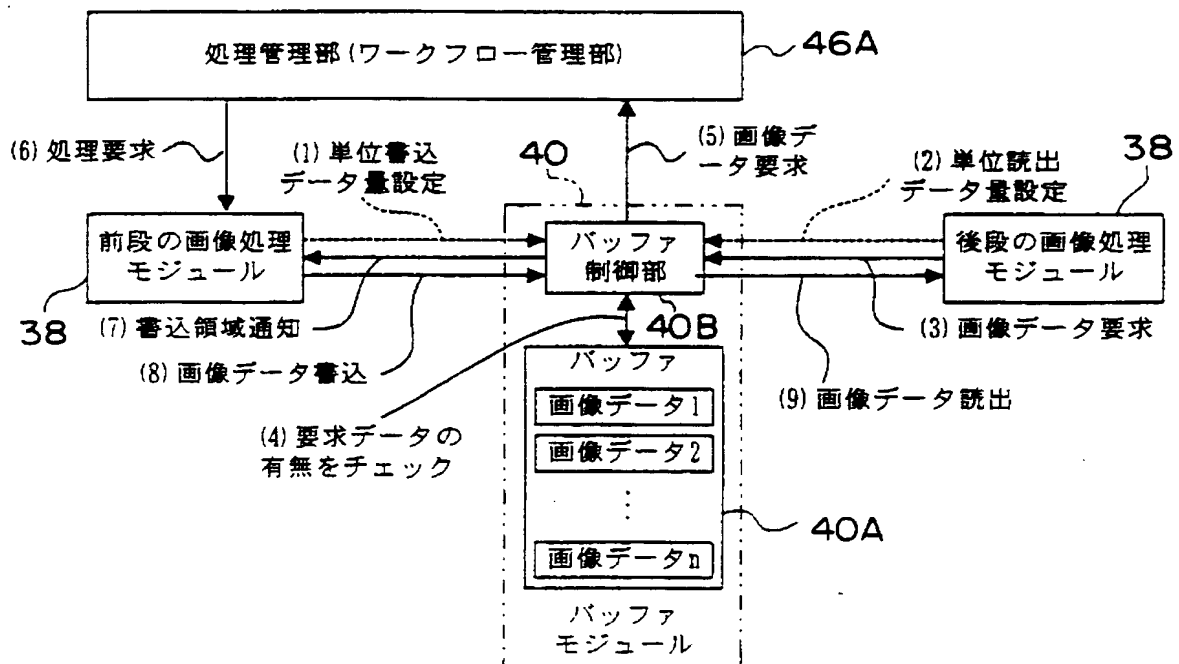
【図 3】

(A) 画像処理モジュールの概略構成・処理

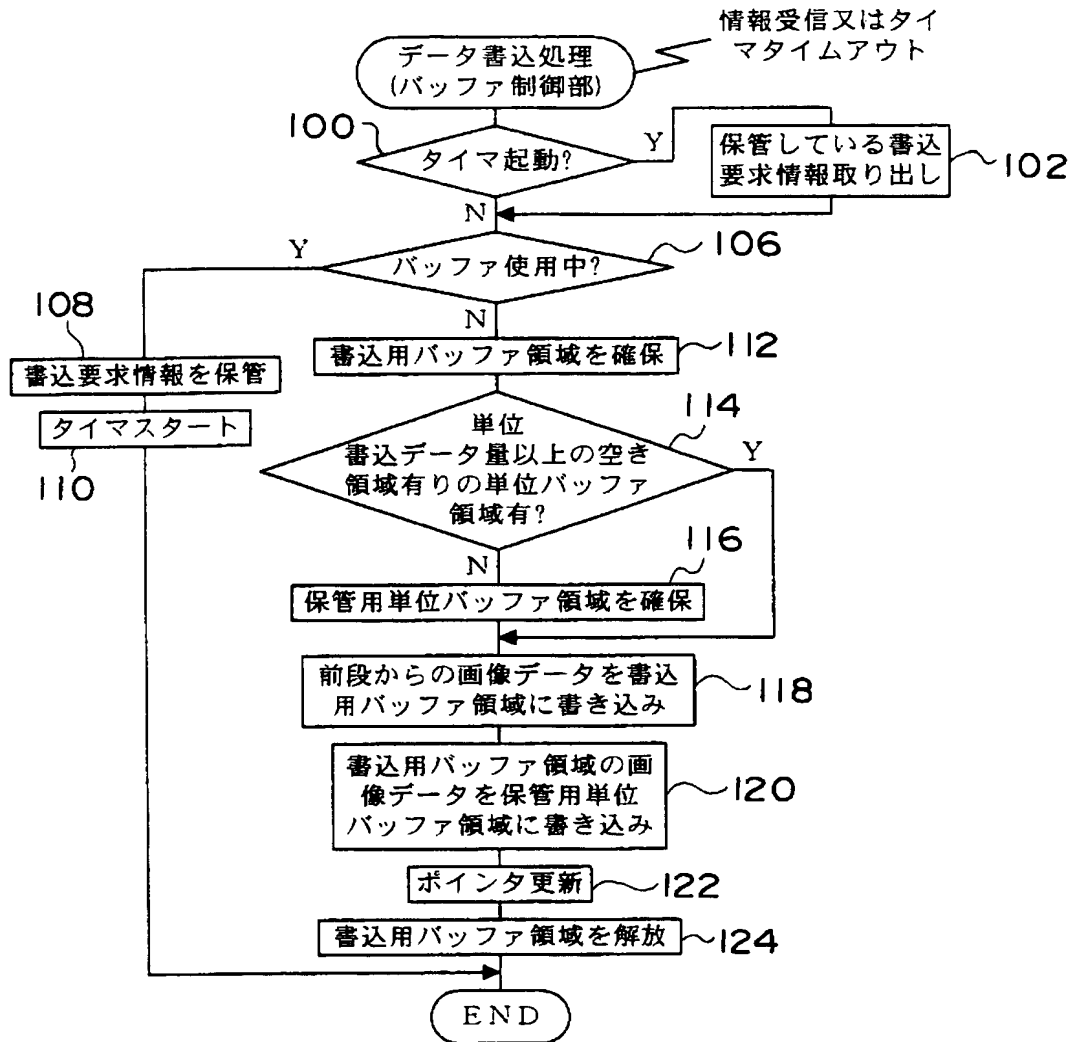


(B) バッファモジュールの概略構成・処理

(後段から要求された画像データがバッファに記憶されていない場合)

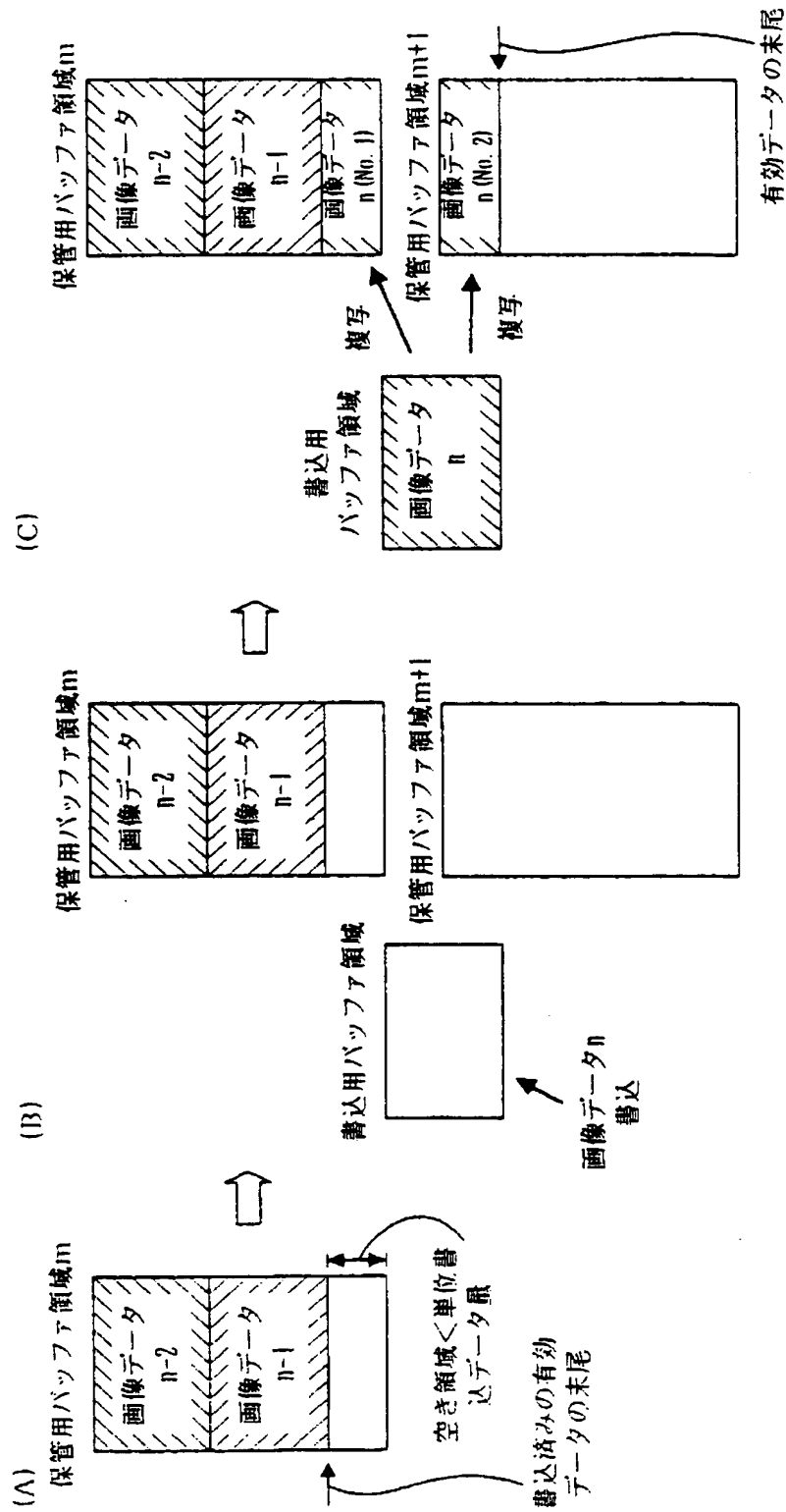


【図 4】

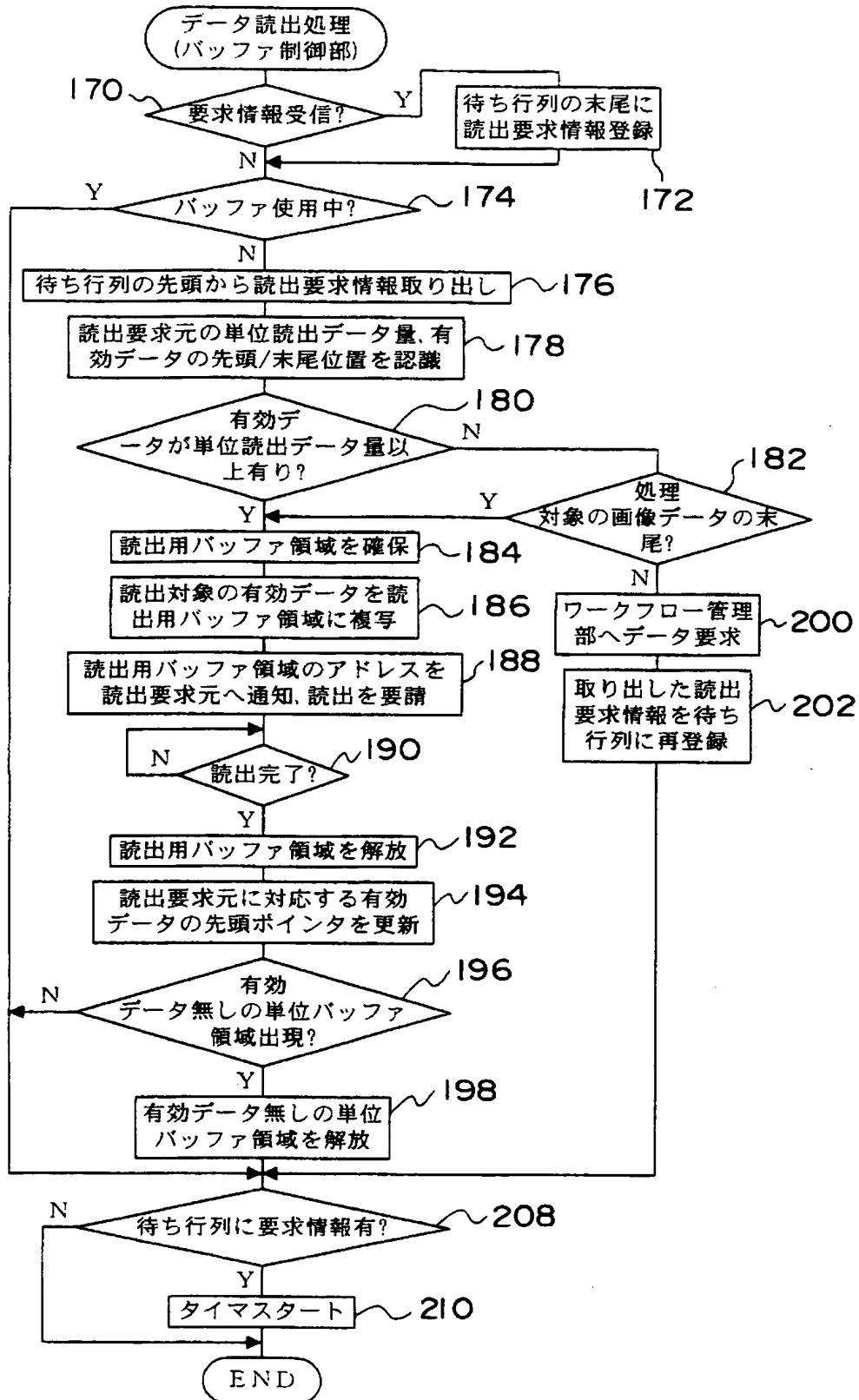


【図 5】

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の処理

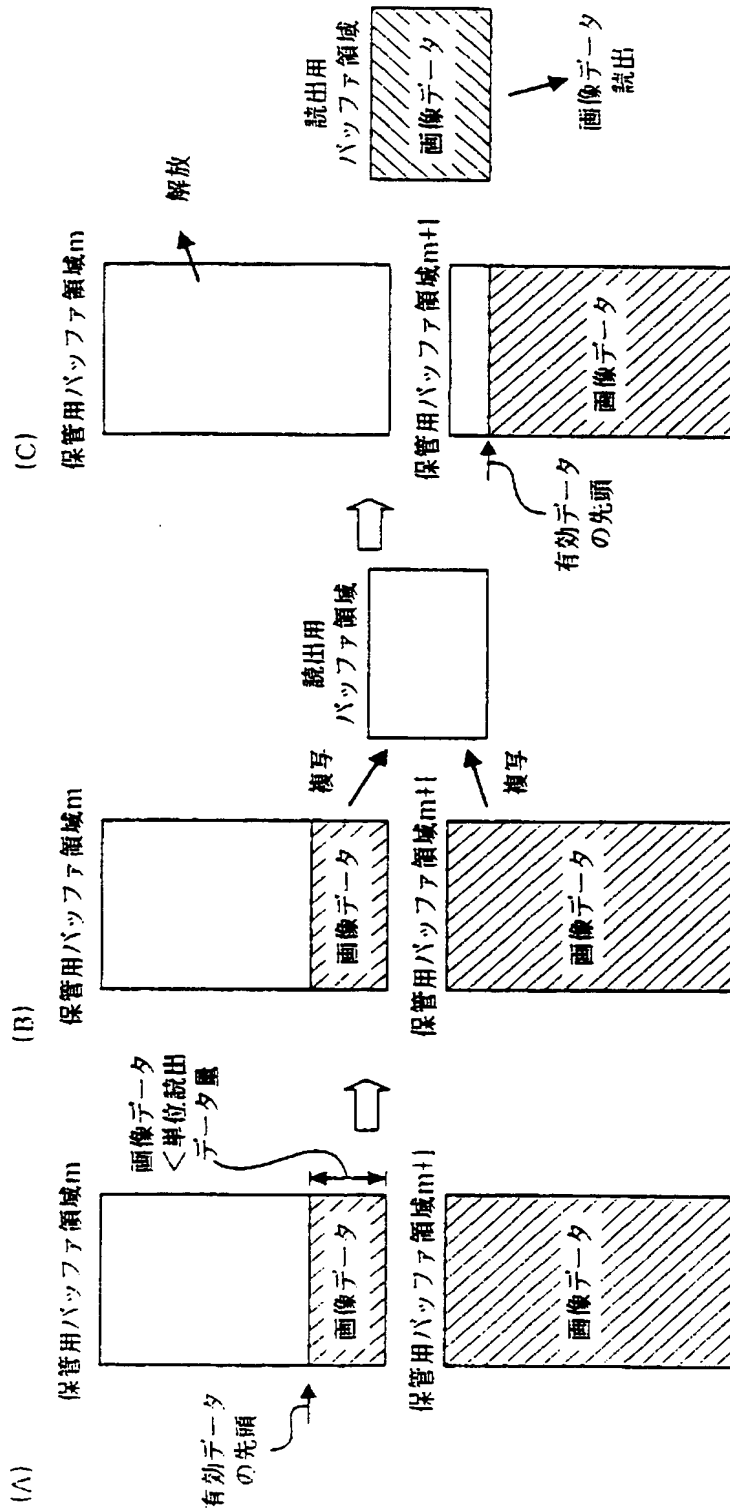


【図 6】

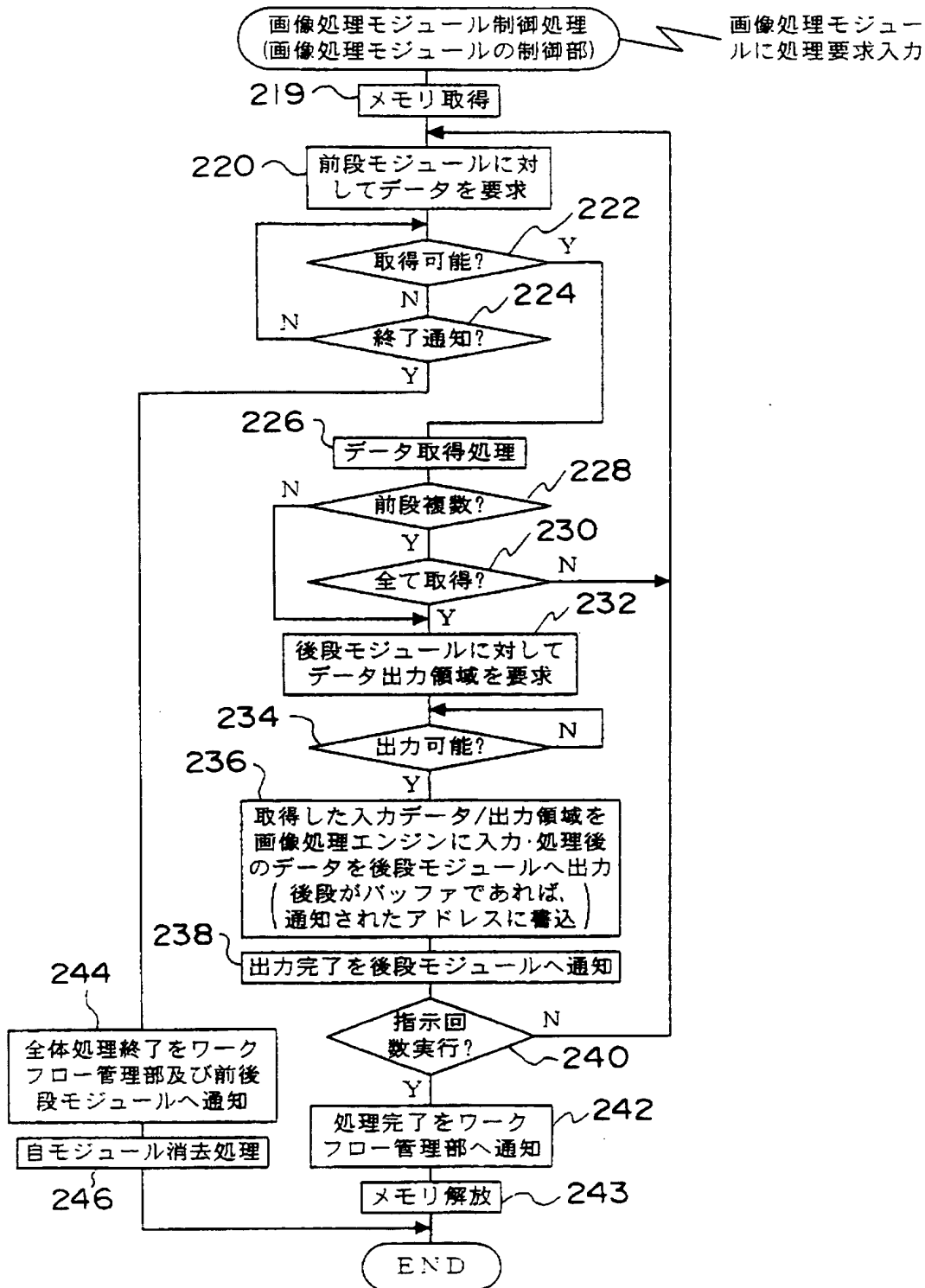


【図 7】

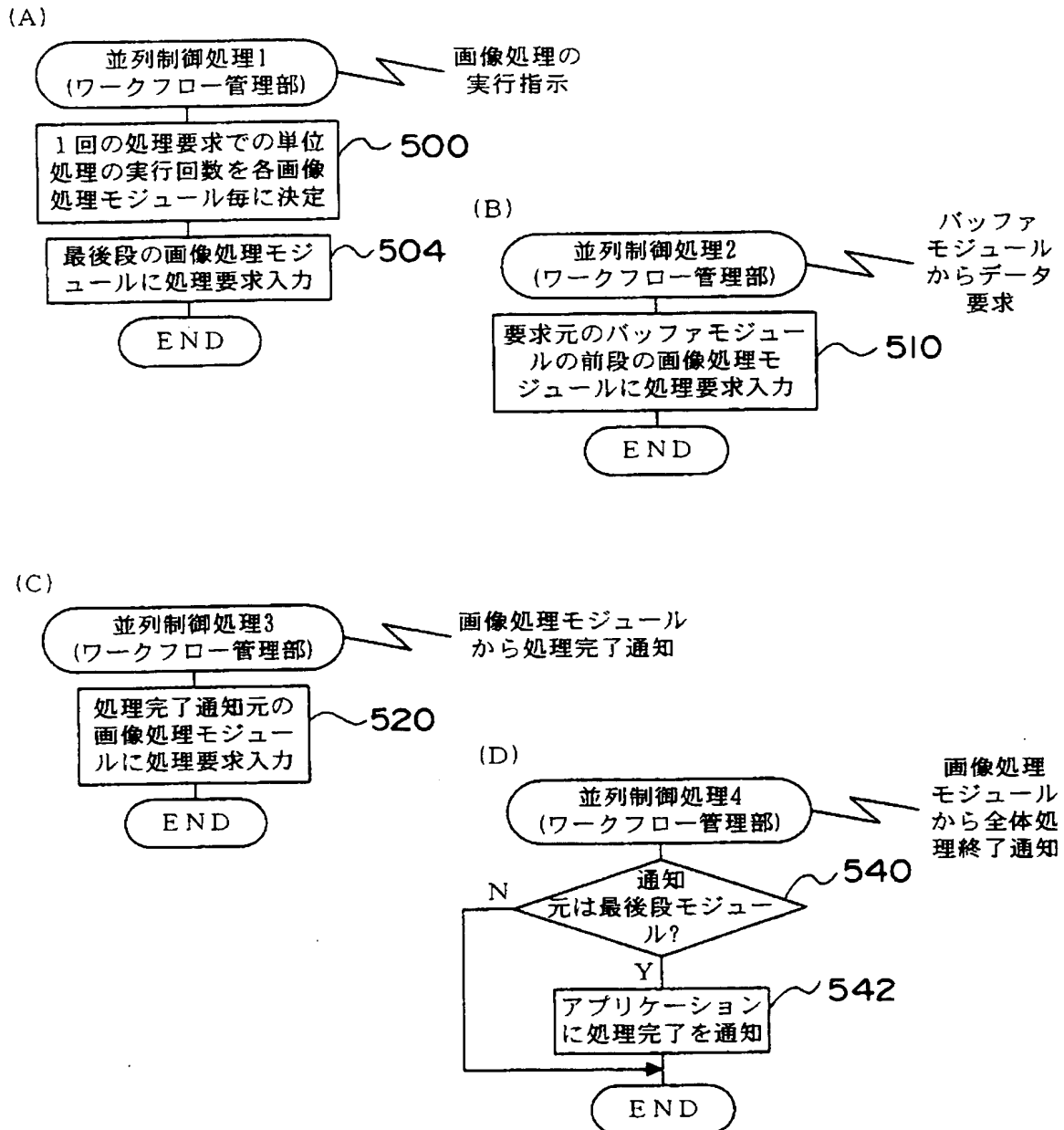
読出対象の画像データが複数の保管用単位バッファ領域に跨っていた場合の処理



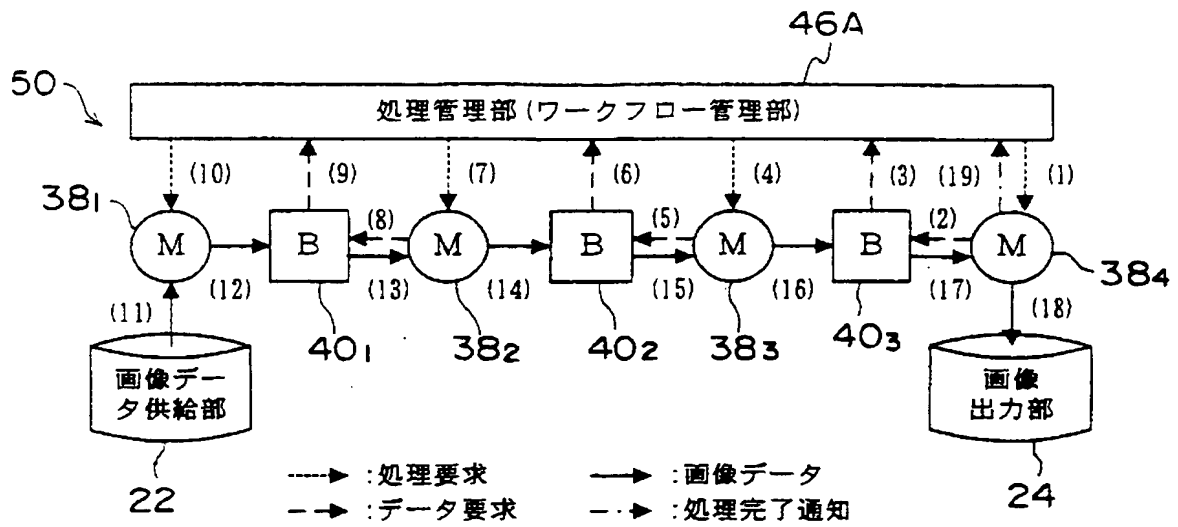
【図 8】



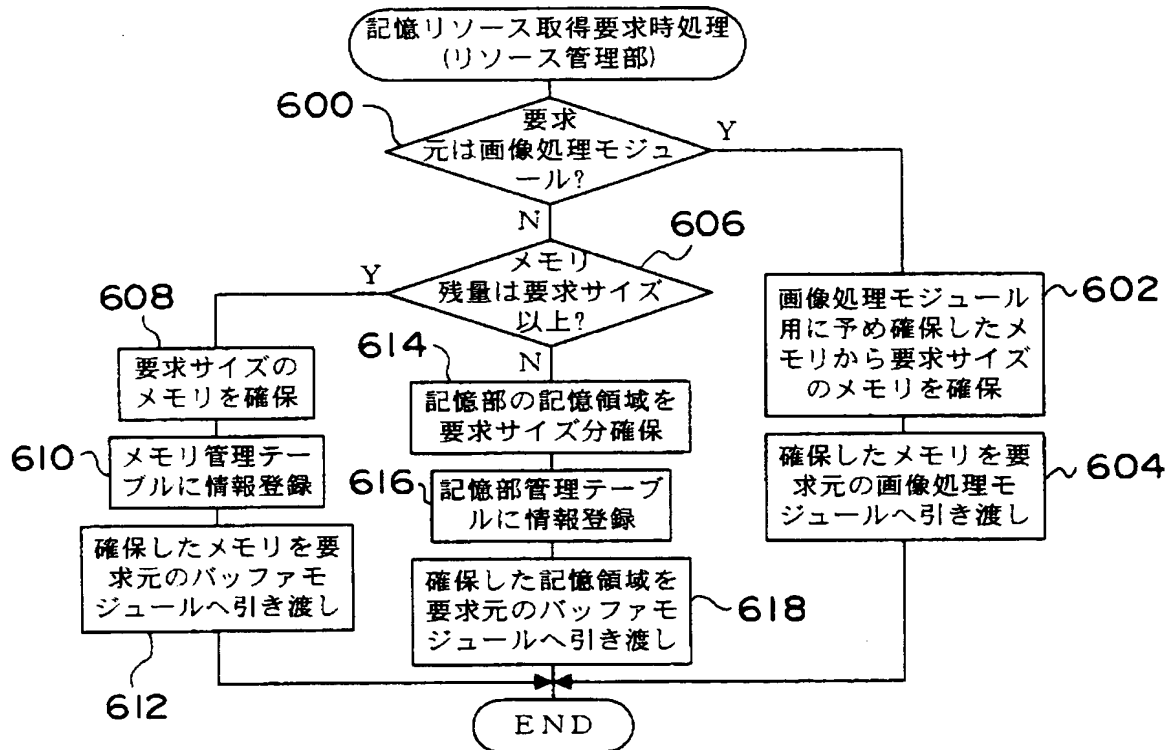
【図 9】



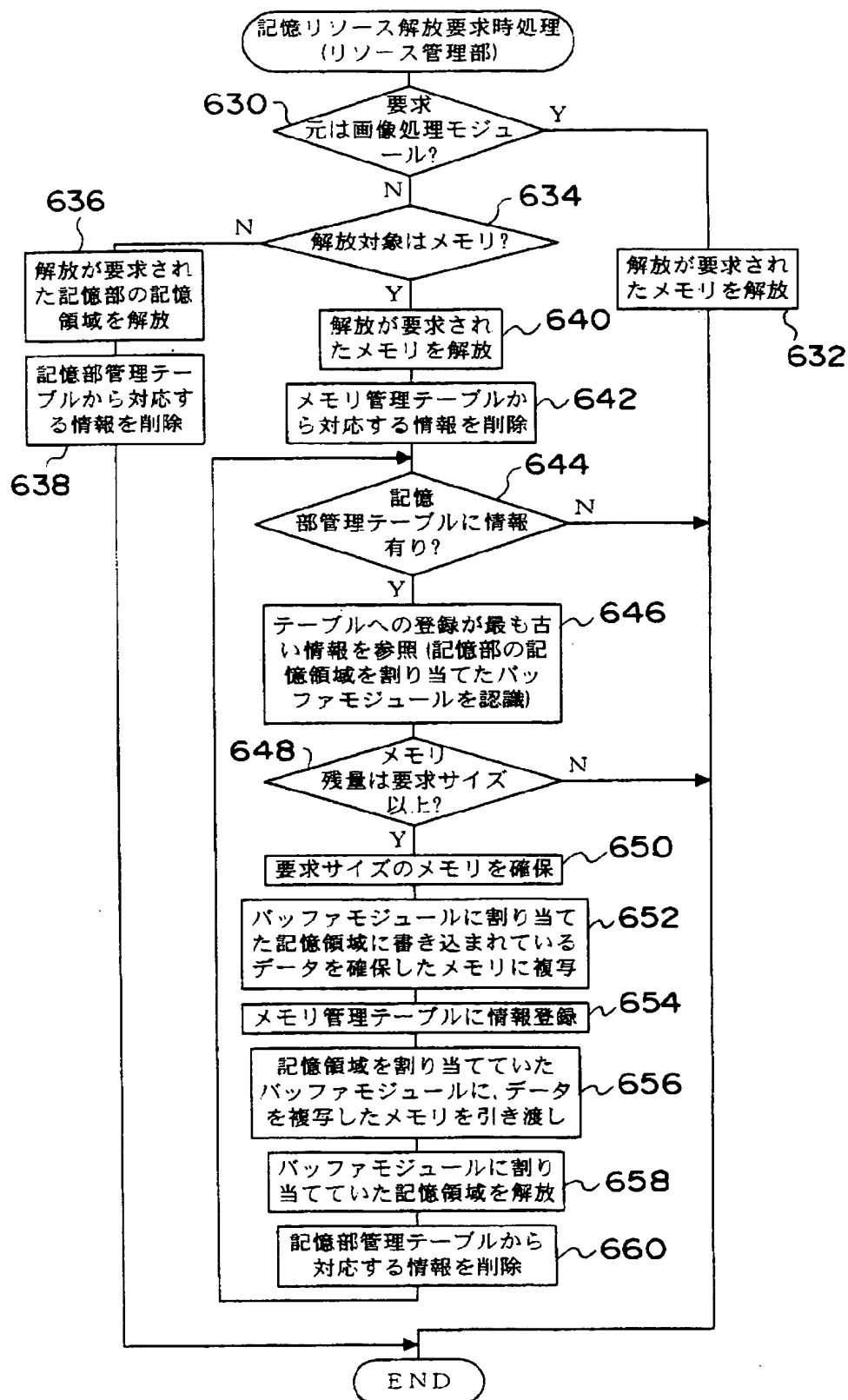
【図 10】



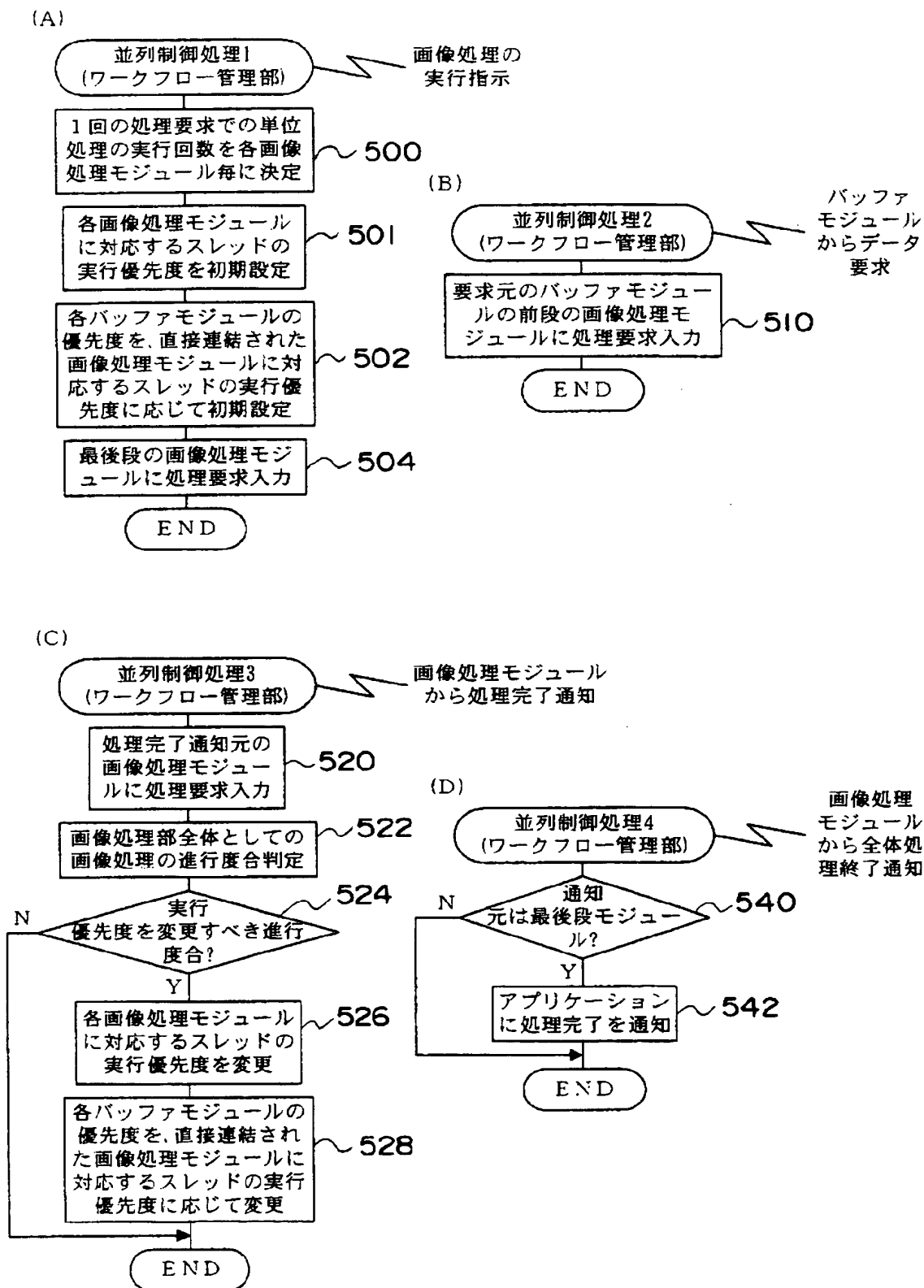
【図 11】



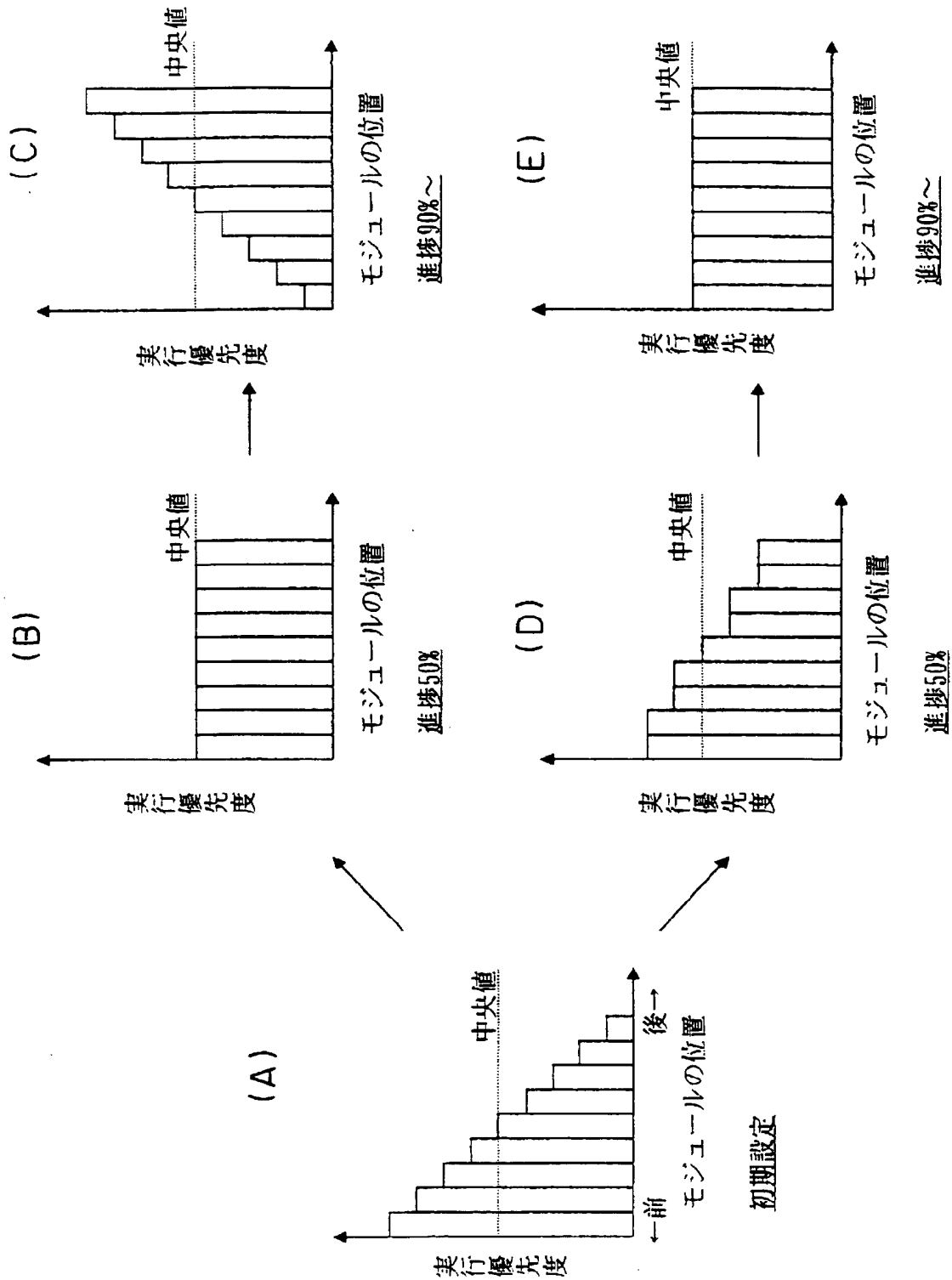
【図 12】



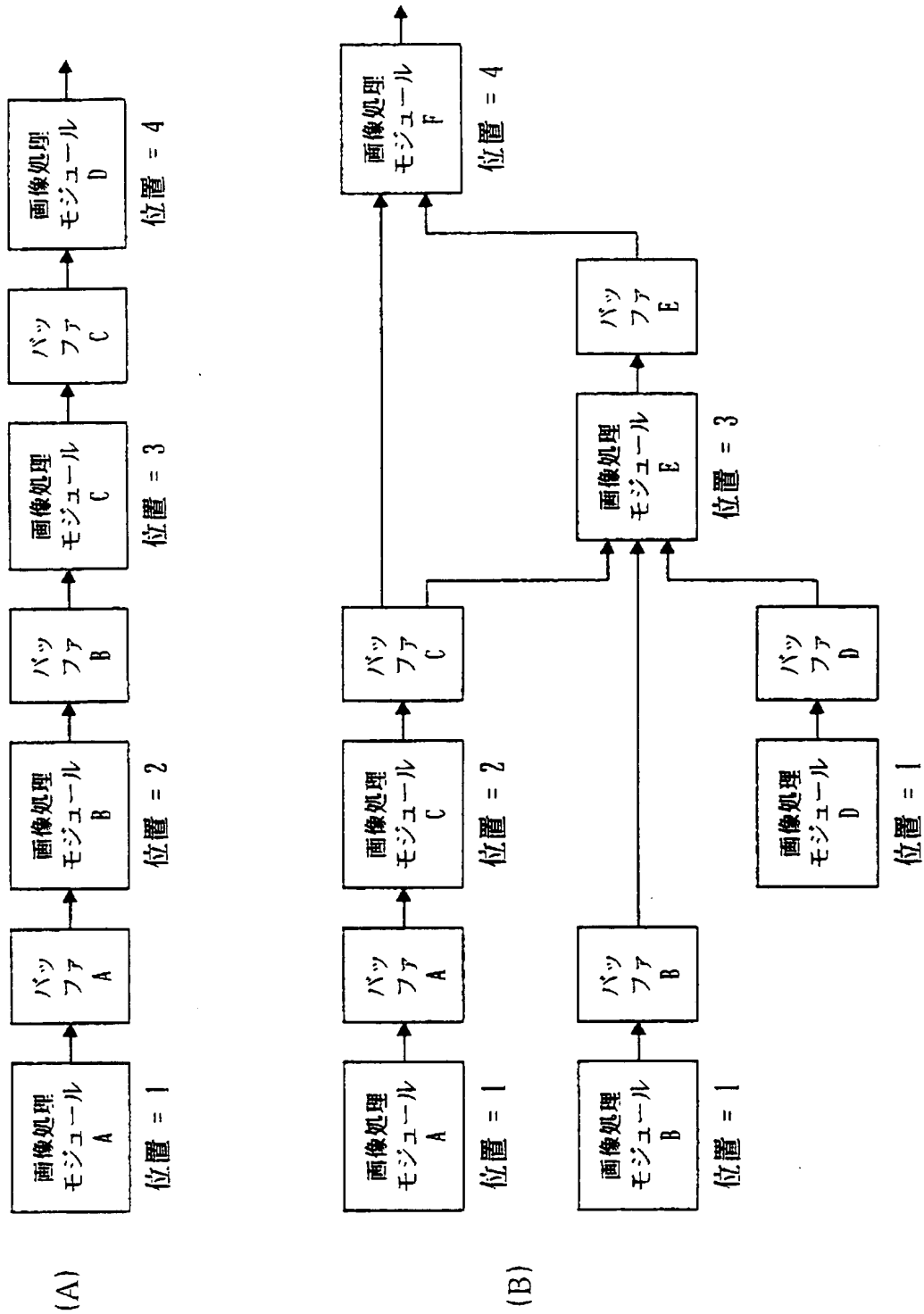
【図 13】



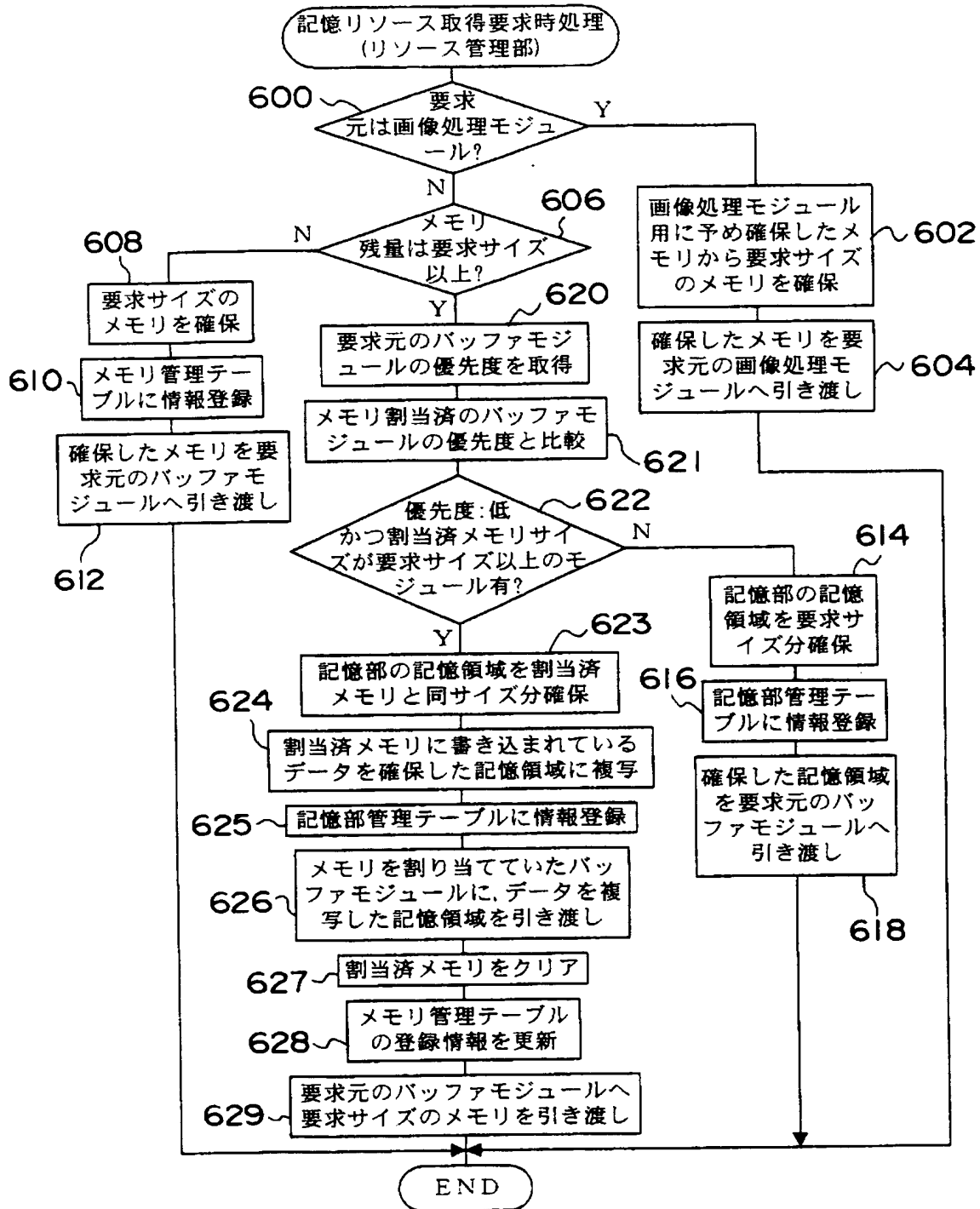
【図14】



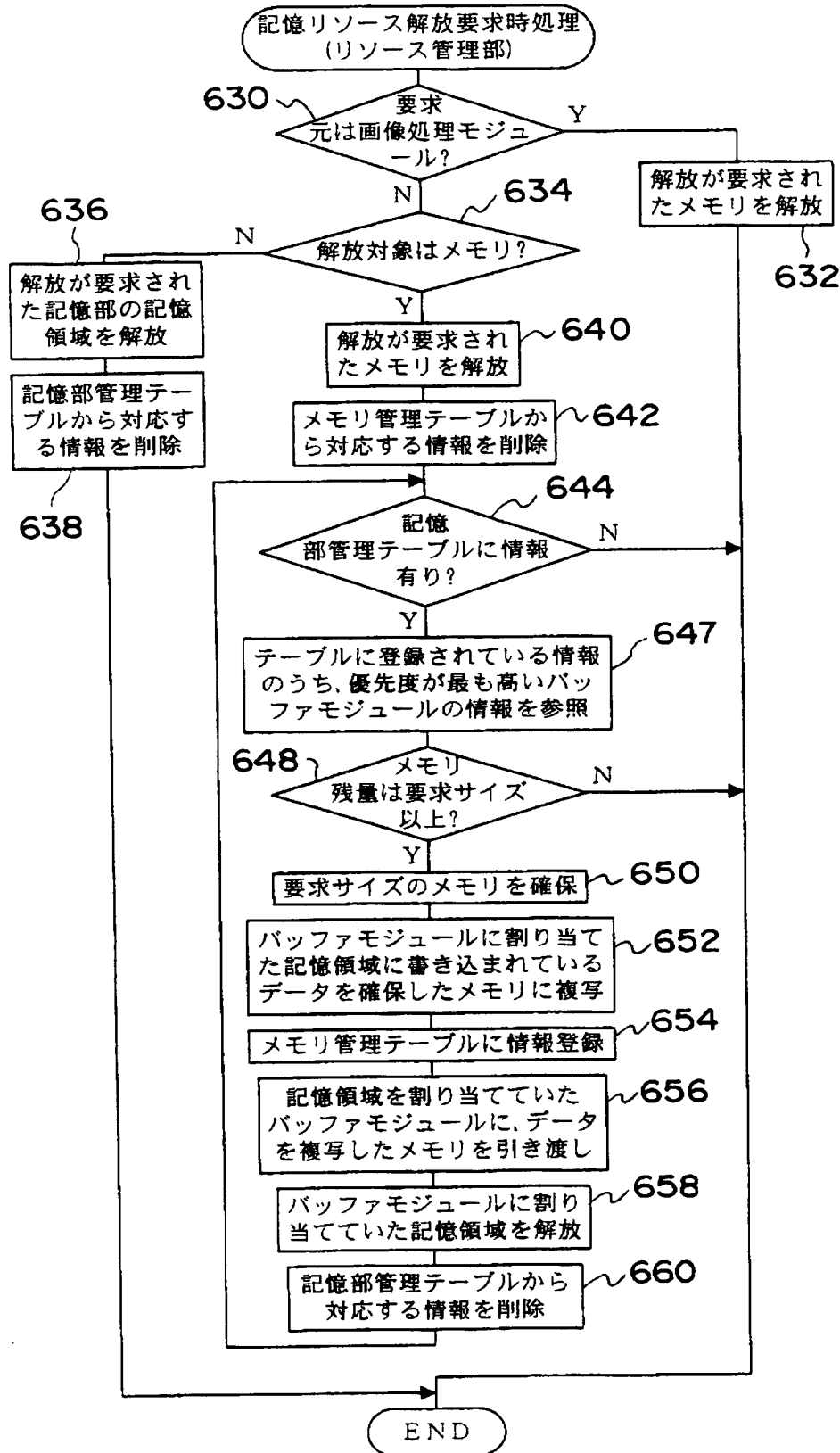
【図 15】



【図 16】



【図 17】



【書類名】 要約書**【要約】**

【課題】 パイプライン形態等で連結された複数の画像処理モジュール及びバッファモジュールを並列に動作させる場合にも、メモリの不足に拘わらず画像処理全体を継続させることを可能とする。

【解決手段】 画像処理部を構成するバッファモジュールから記憶リソースの取得が要求されたものの、メモリの残量が記憶リソースの要求サイズよりも小さい場合(600, 606が否定)には、HDD又はフラッシュメモリ等から成る記憶部の記憶領域を要求サイズ分確保し(614)、テーブルに情報を登録した上で要求元へ引き渡す(616, 618)。また、他のバッファモジュールに割り当てていたメモリ領域が解放された場合に、テーブルに登録した情報に基づき、メモリ領域を割り当てし直すことが可能か確認し、可能であれば既に割り当てた記憶領域からデータを複写したメモリ領域を前記記憶領域に代えて割り当てる。

【選択図】 図 1 2

特願 2 0 0 6 - 1 7 9 2 5 7

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 4 9 6]

1. 変更年月日	1 9 9 6 年 5 月 2 9 日
[変更理由]	住所変更
住 所	東京都港区赤坂二丁目 1 7 番 2 2 号
氏 名	富士ゼロックス株式会社

特願 2 0 0 6 - 1 7 9 2 5 7

出 願 人 履 歴 情 報

識別番号 [0 0 0 0 0 5 2 0 1]

1. 変更年月日 1 9 9 0 年 8 月 1 4 日
[変更理由] 新規登録
住 所 神奈川県南足柄市中沼 2 1 0 番地
氏 名 富士写真フイルム株式会社
2. 変更年月日 2 0 0 6 年 1 0 月 2 日
[変更理由] 名称変更
住所変更
住 所 東京都港区西麻布 2 丁目 2 6 番 3 0 号
氏 名 富士フイルムホールディングス株式会社