

平成14年度 研究開発成果報告書

「ブロードバンド時代の高臨場感映像コンテンツ制作技術及び高品質 情報流通制御技術の研究開発」

目 次

1	研究開発課題の背景	3
2	研究開発分野の現状	4
3	研究開発の全体計画	5
3-1	研究開発課題の概要	5
3-2	研究開発目標	8
3-2-1	最終目標（平成16年3月末）	8
3-2-2	中間目標（平成15年3月末）	11
3-3	研究開発の年度別計画	15
3-4	研究開発体制	16
3-4-1	研究開発管理体制	16
3-4-2	研究開発実施体制	17
4	研究開発の概要（平成14年度まで）	19
4-1	研究開発実施計画	19
4-1-1	研究開発の計画内容	19
4-1-2	研究開発課題実施計画	24
4-2	研究開発の実施内容	25
5	研究開発実施状況（平成14年度）	35
5-1	課題ア 高臨場感ディスプレイ（IPD）用デジタル 映像コンテンツ制作に関する基盤技術の開発	35
5-1-1	課題ア-1 超大容量映像データ処理・管理技術	35
5-1-1-1	はじめに	35
5-1-1-2	実施内容	35
5-1-1-3	まとめと今後の課題	97
5-1-2	課題ア-2 超高精細・広視野角映像の撮影技術	98
5-1-2-1	はじめに	98
5-1-2-2	実施内容	98
5-1-2-3	まとめと今後の課題	144
5-1-3	課題ア-3 幾何学的整合性を考慮した映像編集処理技術	146
5-1-3-1	はじめに	146
5-1-3-2	実施内容	146
5-1-3-3	まとめと今後の課題	170
5-1-4	課題ア-4 リアルタイム映像作成技術	171

5-1-4-1	はじめに	171
5-1-4-2	実施内容	171
5-1-4-3	まとめと今後の課題	207
5-2	課題イ 超広帯域且つ超高品質ネットワーク制御 に関する基盤技術の研究開発	208
5-2-1	課題イ-1 階層フロー識別型通信品質制御技術の研究開発	208
5-2-1-1	はじめに	208
5-2-1-2	実施内容	208
5-2-1-3	まとめと今後の課題	220
5-2-2	課題イ-2 イーサネットにおける通信品質制御、管理技術 の研究開発	221
5-2-2-1	はじめに	221
5-2-2-2	実施内容	221
5-2-2-3	まとめと今後の課題	260
5-2-3	課題イ-3 IPネットワークとイーサネットワークの 連携技術の研究開発	262
5-2-3-1	はじめに	262
5-2-3-2	実施内容	262
5-2-3-3	まとめと今後の課題	267
5-3	総括	268

参考資料、参考文献

(添付資料)

1 研究発表、講演、文献等一覧

1. 研究開発課題の背景

近年 ADSL、FTTH に代表されるブロードバンドの普及により、地理的、時間的、経済的な制限を受けない映像情報サービスの提供が可能になってきている。一例をあげれば、教育コンテンツの VOD 化、遠隔地への画像ベースの医療情報の提供、交通状況のリアルタイム監視、遠隔協調作業によるバーチャルファクトリなどが身近な対象である。これらのアプリケーションにおける表示手段としては、一般のモニタディスプレイが使われることが多いが、これに替わる映像表示手段として、大きなスクリーンに高精細・広視野角な映像を投影する高臨場感ディスプレイ (IPD: Immersive Projection Display) が近年注目されつつあり、これを利用することで、上記のようなアプリケーションのさらなる高機能化や高付加価値化が期待できる。

IPD の具体的な形態としては、曲面大型スクリーンへ投影するシステムや、複数の平面から構成される多面スクリーンへ投影するシステムなどが開発されており、バーチャルリアリティ等の分野で既に活用されている。しかし、映像を表現するデータ量が膨大になるため、その実用はスタンドアロンに限られているものが多かった。

しかしながら、近年のブロードバンドの進展は、IPD のネットワーク対応型ディスプレイとしての位置付けを可能にしつつある。これにより、例えば教育分野においては、教室に居ながらにしてミクロの世界を体験させるなど科学の理解を深めるためのコンテンツを遠隔の学校に配信し講義をおこなったり、医療分野においては、患部を目視と同等以上の解像度で実寸大に映し出すことで遠隔治療を可能にするなど、様々なアプリケーションの出現が期待できる。またネットワークのオープン・双方向・超高速化の流れの中で、これらのアプリケーションを発展させていくことは、様々な新市場の創出による雇用促進、経済活性化の起爆剤としての役割も期待できる。

そこで本研究では、高精細・広視野角・非平面といった特徴を持つ IPD用デジタル映像コンテンツに関し、その撮影から、編集、配信、表示に至るまでのトータルなソリューションを提供できる基盤技術の確立を目的とし、IDP 用映像制作のための映像撮影・編集技術、大容量データ処理技術、およびこれらのデータをネットワークで結ばれた遠隔地間で円滑に実施・流通させる超広帯域・超高品質ネットワーク制御に関する技術の開発を行う。

2. 研究開発分野の現状

高臨場感ディスプレイ (IPD) に関しては、いまだ発展途上の技術であり、多くの研究が続けられているものの、同時にその実用化も広く行われている。例えばアミューズメントにおける新しい形態の映像コンテンツ、防衛や医療等におけるシミュレーション、科学技術計算の可視化などがその代表的なものである。しかしながらそこで用いられる映像は、CG (コンピュータグラフィックス) で作られたものや、CGに実写を部分的に合成したものが大半であり、実写を全面的に用いたものは少なかった。これは IPD のための高精細・広視野角な実写映像を撮影する手段が、一般に利用できるものとして提供されていない、あるいはこのような実写映像が撮影できたとしても、それをシナリオに合うコンテンツとして編集を行い、IPD 表示環境で投影できる形式に変換する画像処理を、効率的に行うような映像編集システムが提供されていなかったことによるものが大きい。

視野角の大きな映像を撮影する手段は、全方位画像センサ (Omnidirectional Sensor) と呼ばれる分野で研究が進められている。特殊光学系を用いるものや、カメラを多数用いるものなどが多く提案・開発されているが、研究等を目的として作られたものが多く、残念ながら普及に至っていないのが実状である。またこれらのシステムは、必ずしも IPD に表示する映像コンテンツの撮影を前提に開発されているわけではなく、IPD に適用するためには、解像度等の仕様の問題や品質の問題、映像の繋ぎ合わせの問題などをすべて解決させる必要がある。

以上は、IPD アプリケーションを開発するための要素技術に関する現状であるが、アプリケーション技術については、IPD を遠隔に配置し、両者間での通信同期制御を行うことで、実写やCGモデルによる映像空間を共有したりロボットなどの制御を行う技術が、テレプレゼンスやテレイクジスタンスなどと呼ばれる分野で研究されている。しかしこれらは、未だ試行段階のものが多く、より厳密なリアルタイム性の実現など、通信性能の向上や通信技術の確立が課題となっている。その意味でインターネットなどにおける一般的なベストエフォート型のサービスの利用は、通信品質が保証されないという問題がある。これに対し、高速ネット占有による高精細パッシブ映像の伝送が、試行中あるいは一部実用化されているが、IPD への適用は特に考慮されていない。また、優先制御や帯域制御を用いた、一階層の通信品質制御技術は従来技術として存在しているが、本研究開発が対象とするような IPD アプリケーションには不十分である。そこで、ヘビーユーザによる帯域占有防止機能や、複数アプリケーションの同時使用を可とするための階層フロー識別型通信制御機能などを実現させる、高度な通信制御技術が必要となっている。

3. 研究開発の全体計画

3-1 研究開発課題の概要

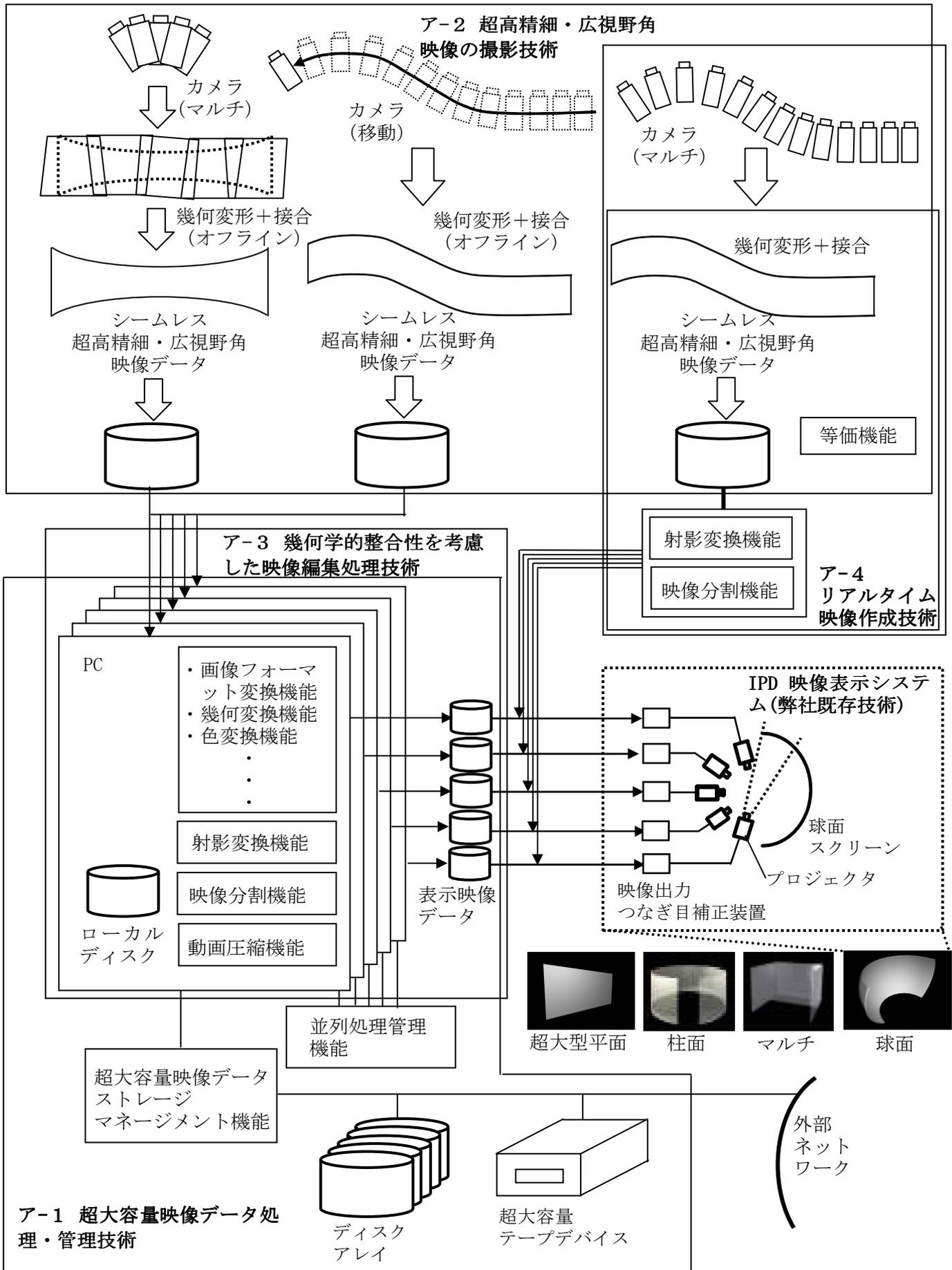
教育、医療、セキュリティ、サービス、文化・芸術、など様々な分野でのアプリケーションが期待できる、高臨場感ディスプレイ (IPD : Immersive Projection Display) 用デジタル映像コンテンツに関し、その制作、実施、及び流通を効率的・高機能に行う技術の確立のため、高付加価値・大容量データの処理技術、及びこれらのデータをネットワークで結ばれた遠隔地間で効果的に処理し、円滑に実施・流通させる超広帯域・超高品質ネットワーク制御に関する技術の研究開発を行う。

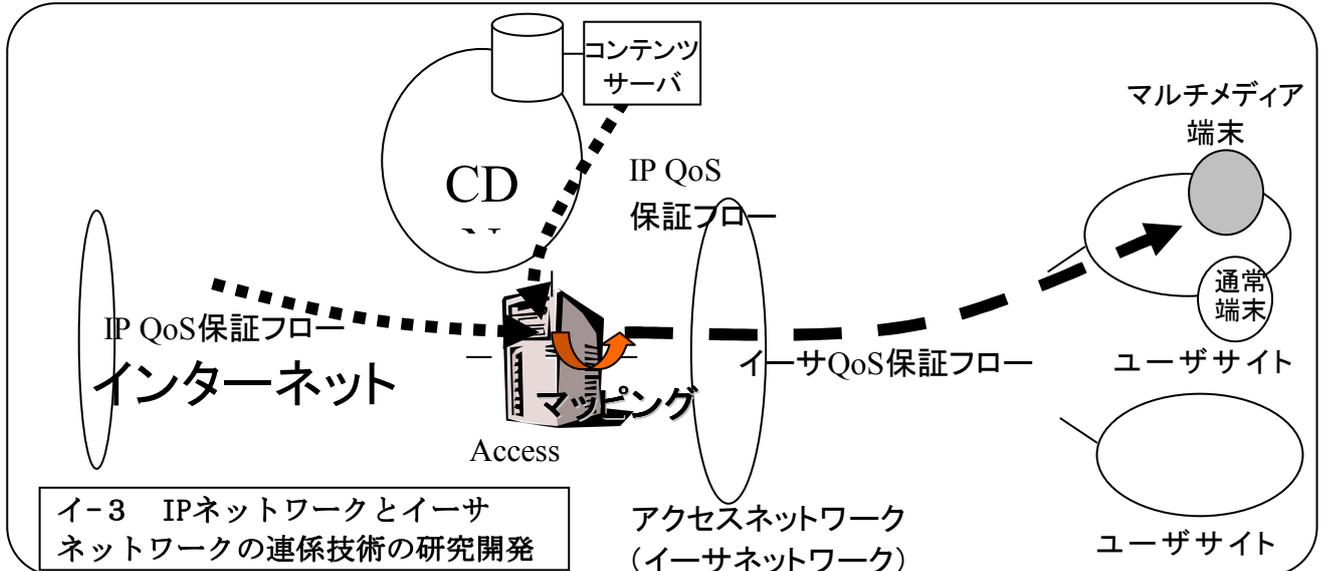
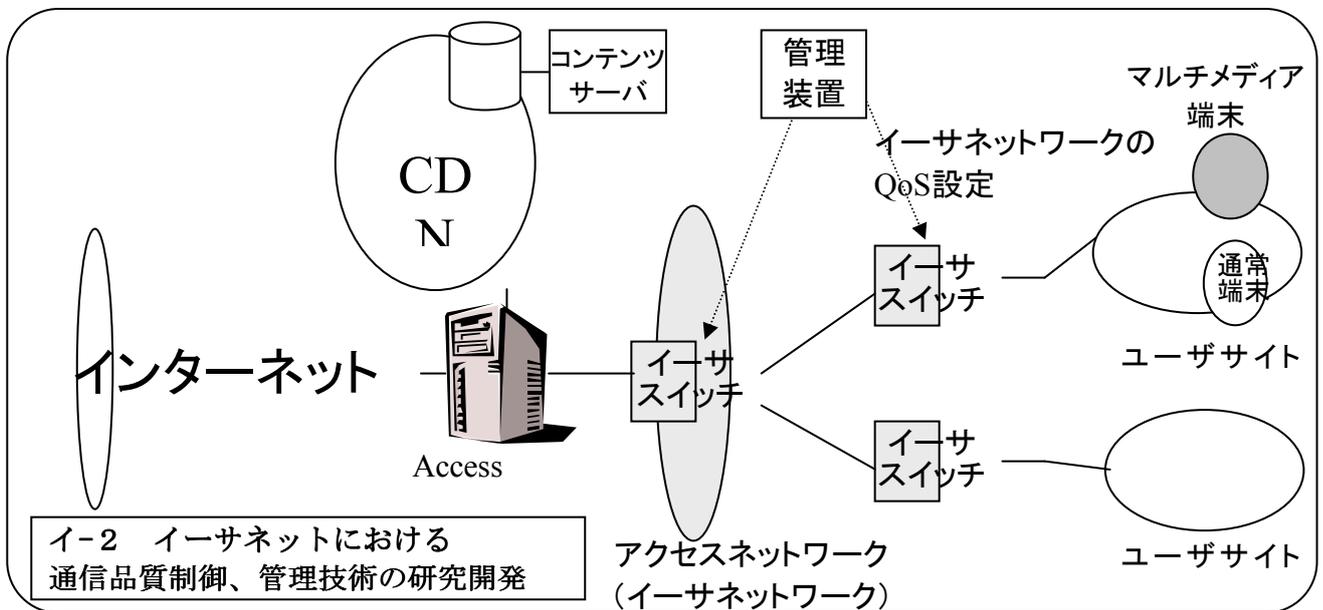
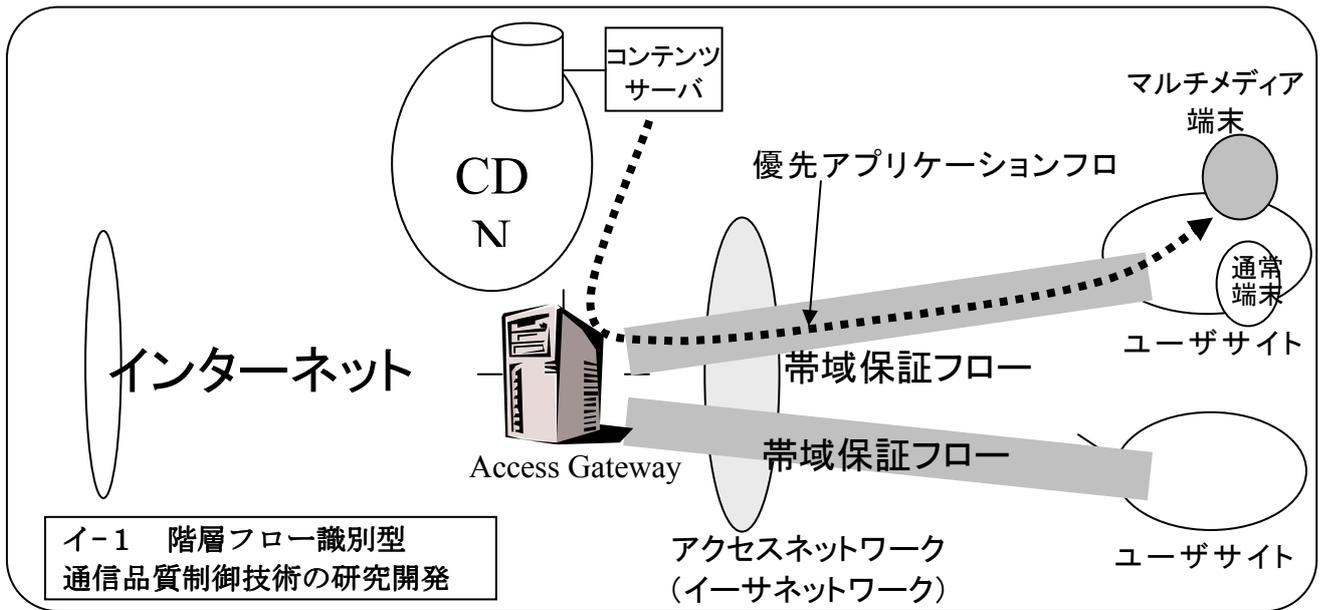
IPD 用映像制作では、IPD が持つ大画面、非平面、超高精細・広視野角などといった特徴のため、テレビや従来の劇場映画等を対象とした映像制作過程では有り得なかった新たな技術課題が生じており、撮影から編集、配信、表示に至るまでのトータルなソリューションが求められる。超高精細映像の撮影機能、映像の幾何学的整合性を考慮した編集機能、映画の 10 倍以上にも及ぶ大容量データの処理方法等、諸課題を克服してはじめて臨場感あふれる新しい価値を持った映像空間の提供が可能になる。また、このような高付加価値で且つ大容量のデータを、ネットワークで結ばれた遠隔地間で効率的に処理し、スムーズに流通させるためには、広帯域なネットワークの実現と通信品質の確立が極めて重要である。

以上を鑑み、以下の 2 テーマについて研究を行う。

課題ア 高臨場感ディスプレイ (IPD) 用デジタル映像コンテンツ制作に関する基盤技術の研究開発

課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発





3-2 研究開発目標

3-2-1 最終目標（平成16年3月末）

高臨場感ディスプレイ (IPD) 用デジタル映像コンテンツ制作環境プロトシステム構築と、それを使用したテストコンテンツの作成、および、同コンテンツを高品質に伝送可能とするネットワーク制御技術の確立を目標とする。各々の項目に対する目標を以下に示す。

(1) 課題ア 高臨場感ディスプレイ (IPD) 用デジタル映像コンテンツ制作に関する基盤技術の開発

(1.1) ア-1 超大容量映像データ処理・管理技術

(1.1.1) 画像処理並列化技術

画像の幾何変換、色変換、スクリーン形状に合わせた射影変換処理、プロジェクタ分割処理、動画圧縮処理などの画像処理が、ネットワーク接続された複数の PC 上で分散並列して行われる仕組みを実現する。プロトシステムにおいて PC は 10 台 ~ 20 台程度を想定するが、50 台以上の構成にも対応可能な設計とする。

またその実現においては、下記の技術を確認し取り入れるものとする。

- ① 処理の効率的なスケジュール管理
- ② ネットワーク転送、データ読み込み・書き出し、計算処理等の個別処理間のボトルネックを解消する最適な処理のスケジュール管理
- ③ 処理をビジュアルに一元管理できるジョブ管理とその GUI 表示

上記機能を持つプロトシステムを開発し、評価を行う。

(1.1.2) 超大容量データストレージ技術・データマネージメント技術

500GB 以上の画像データを、上記画像処理を行う各 PC から効率的に読み出し・書き込みできる仕組みを確認する。具体的には、ディスクアレイサーバと、大容量テープデバイス、ならびに前述の各 PC に置かれたローカルハードディスクの 3 デバイスを用い、実運用を考慮に入れた最適なアーキテクチャおよびその管理方法を確認する。

さらに、同アーキテクチャにおいて、ストレージ機器の故障や、操作者の操作ミス等によって映像データの一部が失われるなどの事故が生じた場合でも、速やかに現状復帰できるように機能する、フェイルセーフ技術についても合わせて検討を加え、設計に取り入れるものとする。

以上の技術を確認し、実機によるプロトシステムを開発し、評価を行う。

(1.2) ア-2 超高精細・広視野角映像の撮影技術

(1.2.1) 複数のビデオカメラを用いた超高精細・広視野角映像撮影技術

複数のビデオカメラを異なる方向を向くように配置し、これらによって撮影された映像の射影変換処理を行い、お互いをシームレスに繋ぎ合わせることで、1 枚の超高精細・広視野角の映像を取得するための基礎技術を開発する。なお、同システムは、実質的に

NTSC 解像度を持つ 5 台以上のカメラより構成されるものを想定する。

(1.2.2) 移動カメラを用いた超高精細静止画像撮影技術の研究開発

移動するカメラによって撮影された連続する複数の異地点映像を、射影変換処理を施しながら擬似的にシームレスに接続することで、目的となる超高精細な静止画像を取得する技術を確立する。具体的には下記を目標とする。

- ① 5 以上の自由度を持つハンドロボットにビデオカメラを取り付け、ロボットコントロールにより任意の位置・方向の映像を撮影し、撮影された映像を mosaicing (stitching) 処理により、擬似的にシームレスに接続する手段を開発する。本手段を実現する、システム設計、基本アルゴリズムの開発および試作を完了し、実証実験を行う。
- ② 自動走行ロボットにカメラを取り付け、あらかじめ与えられた経路に沿って映像を撮影する。撮影された映像を mosaicing (stitching) 処理により、擬似的にシームレスに接続する手段を開発する。本手段を実現する、システム設計、基本アルゴリズムの開発および試作を完了し、実証実験を行う。

(1.3) ア-3 幾何学的整合性を考慮した映像編集処理技術

(1.3.1) 映像編集情報管理方式の研究

映像制作・編集過程における、合成・変形等の処理フローを、出力映像の表示環境の特性に依存する部分と、非依存な部分に分けて定義し、処理・管理可能な映像編集システムを設計する。映像表示環境固有のパラメータデータがあたえられたときに、映像解像度、スクリーン形状、プロジェクタの特性、映像再生方式など、映像表示環境の特性に対応した仕様の映像として加工し、出力可能なものとする。

本設計内容を、課題ア-1 で開発するプロトシステムに反映させ、評価を行う。

(1.3.2) カメラワーク管理方式の研究

撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪みなど）などの情報を、映像データに合わせて記録し、映像編集時や表示に、その情報を効果的に利用できる仕組みについての検討・開発を行い、これを課題ア-1、および課題ア-2 で開発するプロトシステムに連携させ、評価を行う。

(1.3.3) 実験コンテンツの制作

課題ア-1、課題ア-2、ならびに上記 (1.3.1) (1.3.2) で開発した技術・プロトシステムや CG を使い、IPD 用ディスプレイに表示する映像コンテンツを制作し、開発技術の検証を行う。

(1.4) ア-4 リアルタイム映像作成技術

(1.4.1) 複数カメラによる撮影映像のリアルタイム変換技術

上記課題ア-2 (1.2.1) で挙げた開発技術の、リアルタイム化を試みる。複数のカメラの

撮影映像に対し、スクリーン上で擬似的にシームレスに接続されて表示されるような適切な射影変換や幾何変換を、リアルタイムに行う仕組みを開発する。以上を実現するため、各カメラの位置や方向、色特性、レンズ歪みなどをあらかじめ測定し、それらを考慮に入れた画像変換パラメータを算出するキャリブレーション技術、ならびにその変換パラメータに従い、撮影データをリアルタイムに変換する機能を開発する。

(1.4.2) マルチカメラによる遠隔監視システム プロトタイプの開発

上記開発技術の効果的利用のための、アプリケーション技術の検討およびその検証を目的とした実験アプリケーションを開発する。

本実験アプリケーションは、街路や大型店舗などにカメラを複数設置し、防犯や異常検知などを目的に遠隔監視を行うものを想定する。実際には、オフィス等カメラ設置が容易な場所を選定し、映像撮影から、IPD 映像表示システムにてリアルタイム表示できるまでのトータルな実験システムを構築する。本システムにより、開発技術の検証を行う。

(2) 課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発

(2.1) イ-1 階層フロー識別型通信品質制御技術の研究開発

ギガビットイーサネットインタフェースを有するアクセスゲートウェイのプロトタイプと検証用実験コンテンツを開発し、以下の仕様を満たす技術を確立する。

- ① 2 階層以上のフローを識別し、それぞれの階層のフローに対して通信品質制御を行うこと
- ② 通信品質保証制御は、優先制御および帯域制御が可能であること
- ③ 1000 フロー以上の通信品質制御の設定、管理が可能であること
- ④ フローの最大帯域は 100 Mbpsが可能であること

(2.2) イ-2 イーサネットにおける通信品質制御、管理技術の研究開発

イーサネットワークの通信品質制御を行う管理サーバのプロトタイプを開発し、以下の仕様を満たす技術を確立する。

- ① 管理サーバより複数のイーサスイッチの設定を行い、通信品質制御が可能であること
- ② 1 フローの設定が 5 秒以内に終了すること
- ③ 1000 フロー以上の通信品質制御の設定、管理が可能であること

(2.3) イ-3 IPネットワークとイーサネットワークの連携技術の研究開発

課題イ-1 階層フロー識別型通信品質制御技術の研究開発で開発したアクセスゲートウェイにおいて、以下の仕様を満たす技術を確立する。

- ① IP レベルの通信品質制御とイーサネットレベルの通信品質制御のマッピングが可能であること
- ② 1 物理回線上の 1000 以上のフローに対して、IPレベルの通信品質制御とイーサネ

ットレベルの通信品質制御のマッピングが可能であること

3-2-2 中間目標（平成15年3月末）

(1) 課題ア 高臨場感ディスプレイ（IPD）用デジタル映像コンテンツ制作に関する基盤技術の開発

(1.1) ア-1 超大容量映像データ処理・管理技術

(1.1.1) 画像処理並列化技術

画像の幾何変換、色変換、射影変換処理などの画像処理が、ネットワーク接続された複数の PC 上で、効率的に分散並列して行われる仕組みの開発のため、下記項目の検討・設計・開発を実施する。

- ① 全体アーキテクチャの検討、設計を完了し、概念設計書を作成する。
- ② 映像編集・制作に必要な画像処理ライブラリの検討、設計、開発を行う。最終目標に対し、1/2 程度の機能の実装を完了する。
- ③ 処理の効率的なスケジュール管理手法の検討を行い、その結果を項目① の概念設計書に盛り込む。
- ④ 処理をビジュアルに一元管理できるジョブ管理とその GUI 表示方法の検討および概念設計を行い、概念設計書を作成する。

(1.1.2) 超大容量データストレージ技術・データマネジメント技術

超大容量映像データを、上記画像処理を行う各 PC から、効率的に読み出し、および書き込みできることを最終目的とし、ディスクアレイサーバと、大容量テープデバイス、ならびに前述の PC に置かれたローカルハードディスクの3デバイスを用い、実運用を考慮に入れた最適なストレージアーキテクチャおよびその管理方法の検討を行う。全体アーキテクチャの設計を行い、プロトシステム実現のための概念設計書を完成させる。

(1.2) ア-2 超高精細・広視野角映像の撮影技術

(1.2.1) 複数のビデオカメラを用いた超高精細・広視野角映像撮影技術

複数のビデオカメラを異なる方向を向くように配置し、これらによって撮影された映像の射影変換処理を行うことでお互いをシームレスに繋ぎ合わせて、1 枚の超高精細・広視野角の動画映像の取得を可能にするカメラシステムを開発するための要素技術の基礎検討を行う。

(1.2.2) 移動カメラを用いた超高精細静止画像撮影技術の研究開発

- ① ハンドロボットにビデオカメラを取り付け、任意の位置・方向の映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発する。アルゴリズムの基礎検討書を完成させる。

- ② 自動走行ロボットにカメラを取り付け、あらかじめ与えられた経路に沿って映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発する。アルゴリズムの基礎検討書を完成させる。

(1.3) ア-3 幾何学的整合性を考慮した映像編集処理技術

(1.3.1) 映像編集情報管理方式の研究

映像制作・編集過程における IPD 固有の処理技術の開発のため、平成13年度の研究にて列挙した、スクリーン形状やプロジェクタ構成の違いに対応する IPD 固有の映像編集処理についての具体処理アルゴリズムの開発と、処理全体を効率的に管理するためのシステムアーキテクチャの設計を行う。

概念設計書を完成させるとともに、最終目標に対し 1/2 程度の機能についてのアルゴリズム仕様書を作成する。

(1.3.2) カメラワーク管理方式の研究

撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪み）などの情報を、映像編集時や表示時に利用することを想定し、映像データに合わせて記録・管理する効果的な手法、ならびに編集時に利用するカメラパラメータ情報にあわせてカメラコントロールを行うための制御方法に関する基礎検討を行い、基礎検討書を完成させる。

(1.3.3) 実験コンテンツの制作

課題ア-1、2、3の検証を目的とし平成15年度に開発する実験コンテンツについて、シナリオ、撮影方法、撮影対象、表示方法などの検討を行う。

シナリオ第1バージョンを作成する。

(1.4) ア-4 リアルタイム映像作成技術

(1.4.1) 複数カメラによる撮影映像のリアルタイム変換技術

上記課題ア-2で挙げた各開発技術の、リアルタイム化を実現するためのアーキテクチャの検討、システム設計、実装方法の検討を行う。

実機による実験を行い、基礎検討書を完成させる。

(1.4.2) マルチカメラによる遠隔監視システム プロトタイプの開発

課題アおよび課題イを組み合わせたアプリケーション技術の検討ならびに検証を目的とした実験アプリケーション開発のため、全体システムの設計および、実装に際し必要となる応用技術の検討と開発を行い、概念設計書を完成させる。

(2) 課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究

開発

(2.1) イ-1 階層フロー識別型通信品質制御技術の研究開発

(2.1.1) 通信品質制御技術

通信するコンテンツに対応し、ネットワークの End-to-End での通信品質を制御する技術の開発のため、下記項目の検討・設計・プロトシステム開発を実施する。

- ① レイヤ2およびレイヤ3レベルでの通信ノード（スイッチ、ルータ等）におけるアプリケーション対応フロー識別方法の検討及び設計を完了し、設計仕様書を作成する。
- ② 通信ノード（スイッチ、ルータ等）でのアプリケーション対応フロー識別を可能にする制御プロトプログラムの開発を完了する。

(2.1.2) アプリケーション対応通信品質制御技術

アプリケーションとして超大容量映像データ配信システム（放送型）を想定し、超大容量映像データの通信トラヒックと、該映像データ放送の複数観客からの反応情報返送トラヒックが混在する場合の、各トラヒックに対する通信品質制御技術の開発のため、下記項目の検討・設計・プロトシステム開発を実施する。

- ① 従来の通信レイヤ（レイヤ3以下）の通信品質制御に加え、通信データの情報処理を含む上位レイヤでの通信品質制御方式の検討を完了する。
- ② ネットワーク上に配備する上位レイヤでの通信品質制御対応情報処理サーバの配備方式の検討及び設計を完了し、設計仕様書を作成する。
- ③ 情報処理サーバのプロトプログラム開発を完了する。

(2.2) イ-2 イーサネットにおける通信品質制御、管理技術の研究開発

今後の高速広域ネットワークの主流になると予想される広域イーサネットにおける通信品質制御、管理技術を開発するため、下記項目の検討・設計・プロトシステム開発を実施する。

- ① 通信品質保証ネットワーク管理装置からイーサネットワークを管理するために必要な、イーサスイッチのリモート制御について、ワイドエリアおよびローカルエリア・イーサスイッチの機種に依存しない制御方式の設計を完了し、設計仕様書を作成する。
- ② 通信品質保証ネットワーク管理装置のプロトシステムの開発を完了する。

(2.3) イ-3 IPネットワークとイーサネットワークの連携技術の研究開発

IP ネットワークとイーサネットワークが接続され、かつ、レイヤ3以下の通信品質制御とレイヤ4以上の通信品質制御が連携する次世代のネットワークサービス対応通信品質制御技術を開発するため、下記項目の調査・検討を実施する。

- ① 超大容量映像データを駆使する次世代ネットワークサービスの調査を完了する。
- ② IP ネットワークとイーサネットワークの通信品質制御連携方式の検討を完了する。
- ③ 情報処理機能と通信品質保証機能を有する次世代ネットワークの機能要件及び連携

方式の検討を完了し、基本検討書を作成する。

3-3 研究開発の年度別計画

(金額は非公表)

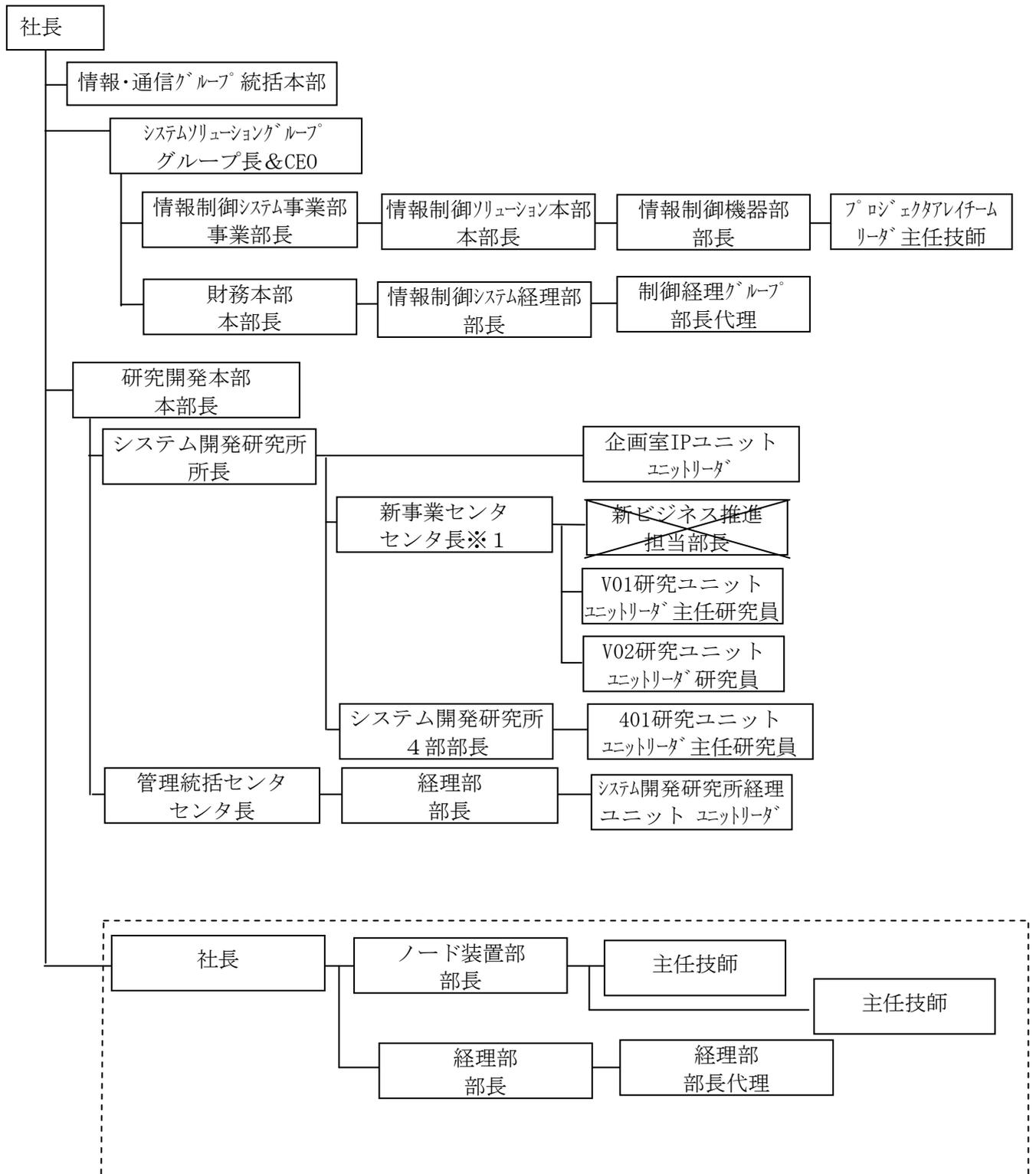
研究開発項目	13年度	14年度	15年度	計	備考
ブロードバンド時代の高臨場感映像コンテンツ制作技術および高品質情報流通制御技術の研究開発					
課題ア 高臨場感ディスプレイ(IPD)用デジタル映像コンテンツ制作に関する基盤技術の開発					
ア-1 超大容量映像データ処理・管理技術			→		
ア-2 超高精細・広視野角映像の撮影技術			→		
ア-3 幾何学的整合性を考慮した映像編集処理技術			→		
ア-4 リアルタイム映像制作技術			→		
課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発					
イ-1 階層フロー識別QoS制御技術の研究開発			→		
イ-2 EthernetにおけるQoS制御、管理			→		
イ-3 IPルータとEthernetネットワークの連携			→		(株)日立コミュニケーションテクノロジー
間接経費					
合計					

注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上(消費税を含む。)

2 備考欄に再委託先機関名を記載

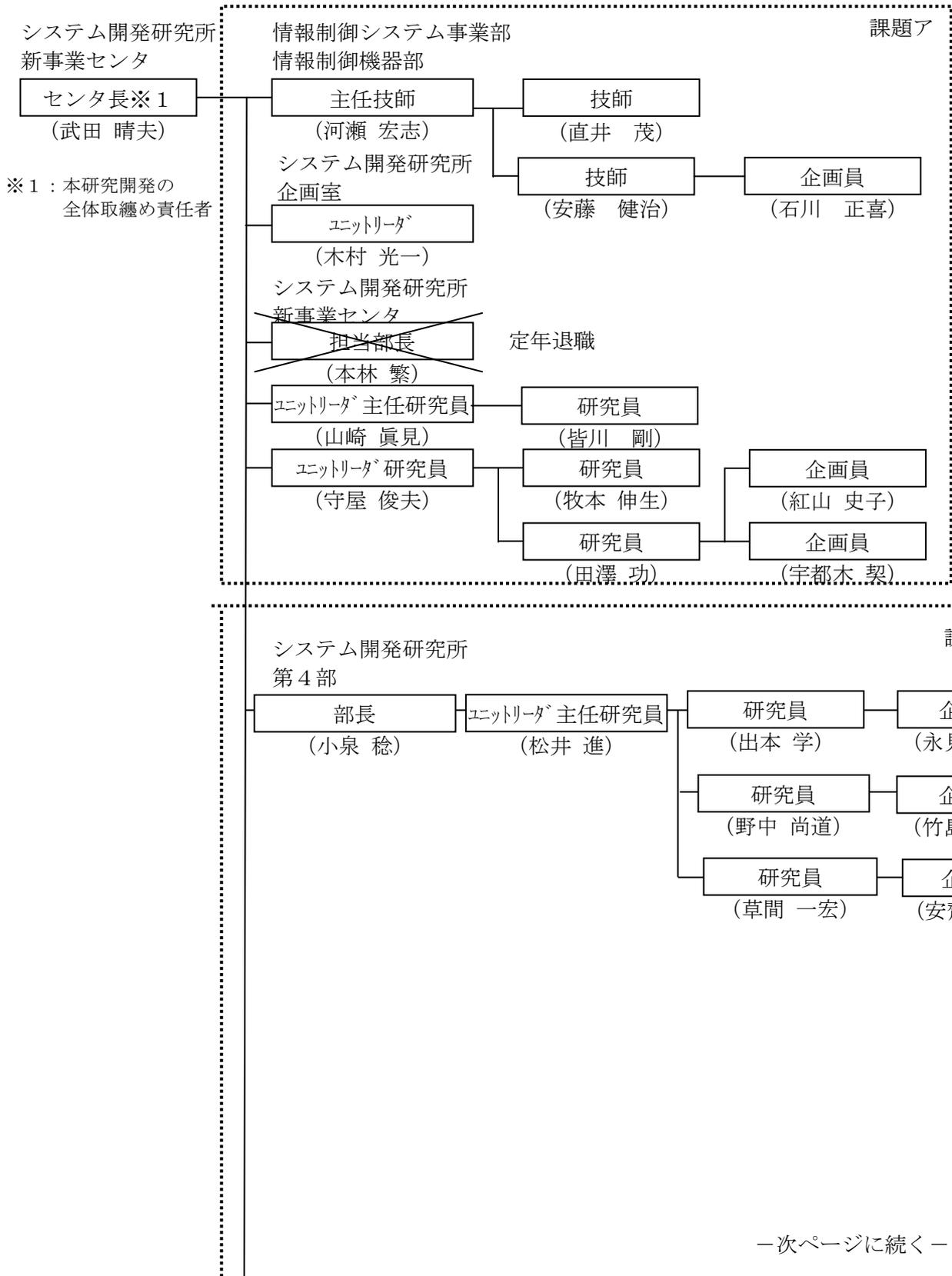
3-4 研究開発体制

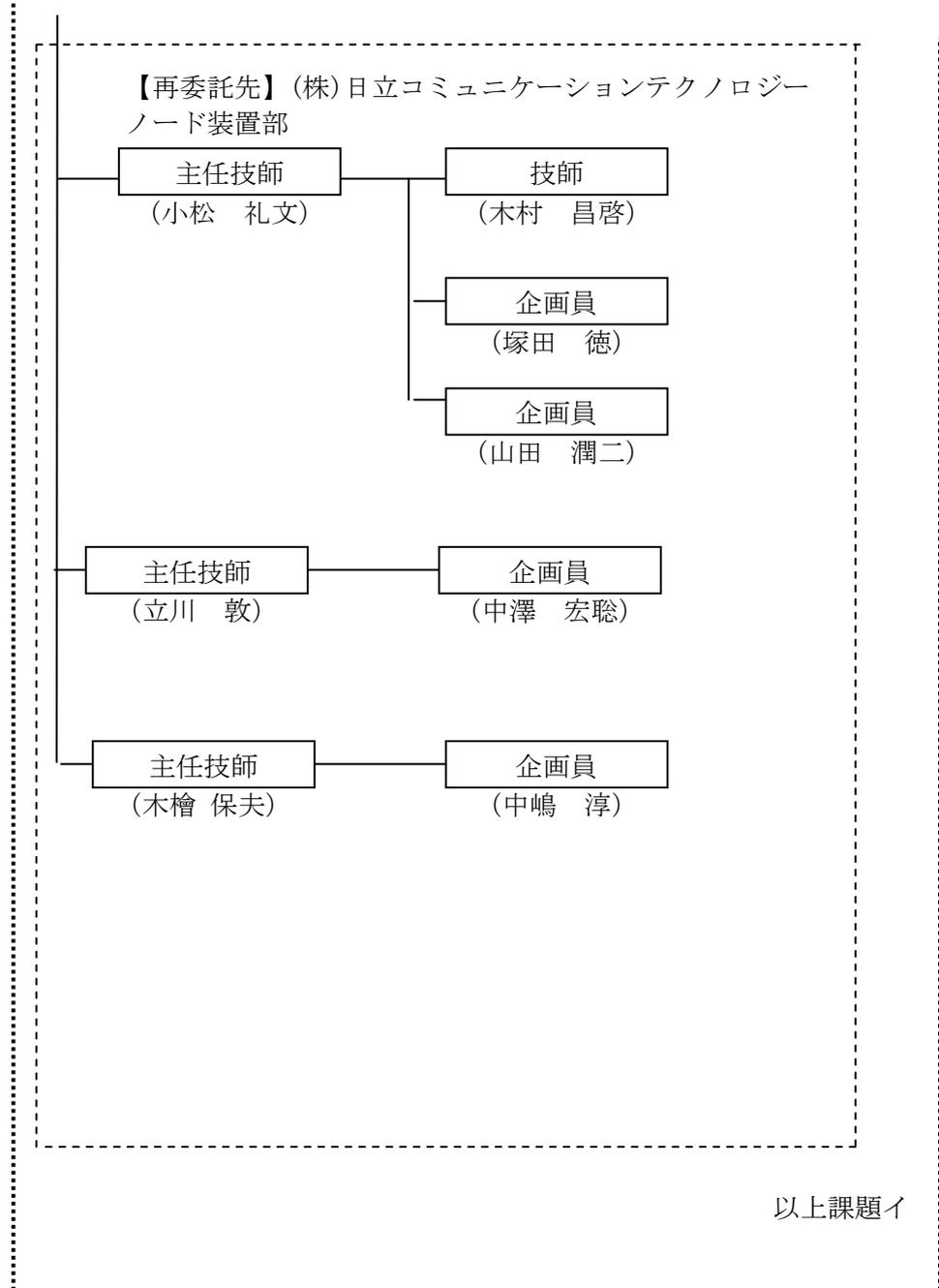
3-4-1 研究開発管理体制



※1：本研究開発の全体取り纏め責任者

3-4-2 研究開発実施体制





4. 研究開発の概要（平成14年度まで）

4-1 研究開発実施計画

4-1-1 研究開発の計画内容

(1) 課題ア 高臨場感ディスプレイ（IPD）用デジタル映像コンテンツ制作に関する基盤技術の開発

平成13年度は、平成14年度に行う中間目標までの研究開発に先立ち、課題項目の詳細化、既存技術調査、主要研究設備の調達等を行う。

平成14年度は、同年度に行う中間発表に向けて、各課題項目の詳細検討、アルゴリズム開発、プロト開発のための設計を主に行う。

各項目の詳細は、以下のとおりである。

(1.1) ア-1 超大容量映像データ処理・管理技術

(1.1.1) 画像処理並列化技術

平成13年度は、画像の幾何変換、色変換、射影変換処理などの画像処理が、ネットワーク接続された複数の PC 上で、効率的に分散並列して行われる仕組みの開発のため、下記項目の検討・準備を行う。

- ① 並列化可能な画像処理項目の列挙
- ② 同 実装に必要となるアルゴリズム等についての公知技術の調査
- ③ 同 処理を実装する PC 等設備の選定と調達

平成14年度は、画像の幾何変換、色変換、射影変換処理などの画像処理が、ネットワーク接続された複数の PC 上で、効率的に分散並列して行われる仕組みの開発のため、下記項目の検討・設計を実施する。

- ① 全体アーキテクチャの検討、設計
- ② 映像編集・制作に必要な画像処理ライブラリの開発
- ③ 処理の効率的なスケジュール管理手法の検討
- ④ 処理をビジュアルに一元管理できるジョブ管理とその GUI 表示方法の検討、設計

(1.1.2) 超大容量データストレージ技術・データマネジメント技術

平成13年度は、超大容量映像データを効率的に保存・管理する技術開発に必要な、ディスクアレイ、テープデバイス、ネットワーク機器等、設備の選定と調達を行う。

平成14年度は、超大容量映像データを、上記画像処理を行う各 PC から、効率的に読み出し、および書き込みできることを最終目的とし、ディスクアレイサーバと、大容量テープデバイス、ならびに前述の PC に置かれたローカルハードディスクの3デバイスを用い、実運用を考慮に入れた最適なストレージアーキテクチャおよびその管理方法の検討

を行う。

(1.2) ア-2 超高精細・広視野角映像の撮影技術

(1.2.1) 複数のビデオカメラを用いた超高精細・広視野角映像撮影技術

平成13年度は、複数のビデオカメラで撮影された映像の射影変換処理によって1枚の超高精細・広視野角の映像を取得する技術の開発のため、下記項目の調査・準備を行う。

- ① 複数のビデオカメラを用いた映像撮影に関する公知技術の調査
- ② ビデオカメラ等の実験設備の選定と調達

平成14年度は、複数のビデオカメラを異なる方向を向くように配置し、これらによって撮影された映像の幾何変換処理を行い、お互いをシームレスに繋ぎ合わせることで、1枚の超高精細・広視野角の動画映像の取得を可能にするカメラシステムの基礎検討を行う。

(1.2.2) 移動カメラを用いた超高精細静止画像撮影技術の研究開発

平成13年度は、移動するカメラによって撮影された連続する複数の異地点映像を、射影変換処理を施しながら擬似的にシームレスに接続する技術の開発に必要となる、カメラ、ロボット等、実験・開発設備の選定と調達を行う。

平成14年度は、下記項目を実施する。

- ① ハンドロボットにビデオカメラを取り付け、任意の位置・方向の映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発する。
- ② 自動走行ロボットにカメラを取り付け、あらかじめ与えられた経路に沿って映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発する。

(1.3) ア-3 幾何学的整合性を考慮した映像編集処理技術

(1.3.1) 映像編集情報管理方式の研究

平成13年度は、映像制作・編集過程における IPD 固有の処理技術の開発のため、下記項目の検討・調査を行う。

- ① スクリーン形状やプロジェクタ構成の違いに対応する IPD 固有の映像編集処理と、それらに関する諸課題の列挙
- ② 課題解決に必要な既存技術の調査

平成14年度は、映像制作・編集過程における IPD 固有の処理技術の開発のため、平

平成13年度の研究にて列挙した、スクリーン形状やプロジェクタ構成の違いに対応するIPD固有の映像編集処理についての具体アルゴリズムの開発と、処理全体を効率的に管理するためのシステムアーキテクチャの設計を行う。

(1.3.2) カメラワーク管理方式の研究

平成13年度は、撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪みなど）の扱い方法に関する既存技術の調査を行う。

平成14年度は、撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪み）などの情報を、映像編集時や表示時に利用することを想定し、映像データに合わせて記録・管理する効果的な手法、ならびに編集時に利用するカメラパラメータ情報にあわせてカメラコントロールを行うための制御方法に関する基礎検討を行う。

(1.3.3) 実験コンテンツの制作

平成13年度は、本項目に関する研究開発は実施しない。

平成14年度は、課題ア-1、2、3の検証を目的とし平成15年度に開発する実験コンテンツについて、シナリオ、撮影方法、撮影対象、表示方法などの検討を行う。

(1.4) ア-4 リアルタイム映像作成技術

(1.4.1) 複数カメラによる撮影映像のリアルタイム変換技術

平成13年度は、上記課題ア-2で挙げた開発技術の、リアルタイム化を試みるための、実験設備の選定と調達を行う。

平成14年度は、上記課題ア-2で挙げた各開発技術の、リアルタイム化を実現するためのアーキテクチャの検討、システム設計、実装方法の検討を行う。

(1.4.2) マルチカメラによる遠隔監視システム プロトタイプの開発

平成13年度は、アプリケーション技術の検討およびその検証を目的とした実験アプリケーションを開発するため、下記項目の検討・準備を行う。

- ① 実験デモンストレーションシステムの全体構想策定
- ② 主要設備の選定と調達

平成14年度は、課題アおよび課題イを組み合わせたアプリケーション技術の検討ならびに検証を目的とした実験アプリケーション開発のため、全体システムの設計および、実装に際し必要となる応用技術の検討と開発を行う。

(2) 課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発

平成13年度は、平成14年度の機能設計、詳細設計およびプロトタイプ実装に先立って、技術調査、方式検討、および主要研究設備の調達等を行う。

平成14年度は、平成13年度に実施した技術調査、方式検討、および主要研究設備の調達等の結果に基づき、方式検討、機能設計、詳細設計およびプロトタイプ実装を行う。

(2.1) イ-1 階層フロー識別型通信品質制御技術の研究開発

(2.1.1) 通信品質制御技術

平成13年度は、次世代ネットワークサービスの通信品質制御条件を整理するため以下の既存技術の調査・整理を行うとともに実験設備の調達ならびにネットワーク構築を行う。

- ① 今後要求されるコンテンツ対応の通信品質に関する調査と整理
- ② ネットワークの End-to-End の通信品質制御技術に関する調査と整理
- ③ 実験設備の調達ならびにネットワーク構築を行う。

平成14年度は、通信するコンテンツに対応し、ネットワークの End-to-End での通信品質を制御する技術の開発のため、下記項目の検討・設計・プロトシステム開発を行う。

- ① レイヤ2およびレイヤ3レベルでの通信ノード（スイッチ、ルータ等）におけるアプリケーション対応フロー識別方法の検討及び設計
- ② 通信ノード（スイッチ、ルータ等）でのアプリケーション対応フロー識別を可能にする制御プロトプログラムの開発

(2.1.2) アプリケーション対応通信品質制御技術

平成13年度は、通信品質制御方式の下記項目について検討を実施する。

- ① 階層フローの定義および検出手段の検討

フローの検出に用いる情報として、送信／受信回線、イーサヘッダ情報、IPヘッダ情報、TCP/UDPヘッダ情報などが考えられる。

ここでは、サービス利用者の観点より要求条件を明確にし、フローの定義および階層化方式の検討を行う。

- ② 階層フロー通信品質制御機能の実装方式案

アクセスゲートウェイのプロトタイプ実装方式案を検討する。

平成14年度は、アプリケーションとして超大容量映像データ配信システム（放送型）を想定し、超大容量映像データの通信トラヒックと、該映像データ放送の複数観客からの反応情報返送トラヒックが混在する場合の、各トラヒックに対する通信品質制御技術の開発のため、下記項目の検討・設計・プロトシステム開発を行う。

- ① 従来の通信レイヤ（レイヤ3以下）の通信品質制御に加え、通信データの情報処理を含む上位レイヤでの通信品質制御方式の検討
- ② ネットワーク上に配備する上位レイヤでの通信品質制御対応情報処理サーバの配備方式の検討及び設計

- ③ 情報処理サーバのプロトプログラム開発

(2.2) イ-2 イーサネットにおける通信品質制御、管理技術の研究開発

平成13年度は、次世代ネットワークサービスの通信品質制御条件を整理するため以下の既存技術の調査・整理を行うとともに実験設備の調達ならびにネットワーク構築を行う。

- ① IP ルータにおける通信品質制御技術の調査と整理
- ② イーサスイッチにおける通信品質制御技術の調査と整理
- ③ 実験設備の調達ならびにネットワーク構築

また、イーサネットにおける通信品質制御・管理方式の下記検討を行う。

- ① イーサスイッチ制御方式

管理装置からイーサネットワークを管理するために必要な、イーサスイッチのリモート制御について、既存イーサスイッチの通信制御技術の調査結果を受けて、適切な制御方式および使用方式について検討する。

- ② イーサネットワーク管理装置の実装方式案

イーサネットワーク管理装置のプロトタイプ実装方式を検討する。

平成14年度は、今後の高速広域ネットワークの主流になると予想される広域イーサネットにおける通信品質制御、管理技術を開発するため、下記項目の検討・設計・プロトシステム開発を行う。

- ① 通信品質保証ネットワーク管理装置からイーサネットワークを管理するために必要な、イーサスイッチのリモート制御について、ワイドエリアおよびローカルエリア・イーサスイッチの機種に依存しない制御方式の設計
- ② 通信品質保証ネットワーク管理装置のプロトシステム開発

(2.3) イ-3 IPネットワークとイーサネットワークの連携技術の研究開発

平成13年度は、次世代ネットワークサービスの通信品質制御条件を整理するため以下の既存技術調査・整理を行う。

- ① IP ネットワークとイーサネット、ATM などのレイヤ2ネットワークの連携による通信品質制御技術の調査と整理

平成14年度は、IP ネットワークとイーサネットワークが接続され、かつ、レイヤ3以下の通信品質制御とレイヤ4以上の通信品質制御が連携する次世代のネットワークサービス対応通信品質制御技術を開発するため、下記項目の調査・検討を行う。

- ① 超大容量映像データを駆使する次世代ネットワークサービスの調査
- ② IP ネットワークとイーサネットワークの通信品質制御連携方式の検討
- ③ 情報処理機能と通信品質保証機能を有する次世代ネットワークの機能要件及び連携方式検討

4-1-2 研究開発課題実施計画

(金額は非公表)

研究開発項目	第1四半期	第2四半期	第3四半期	第4四半期	計	備考
課題ア 高臨場感ディスプレイ(IPD)用デジタル映像コンテンツ制作に関する基盤技術の開発 ア-1 超大容量映像データ処理・管理技術 ア-2 超高精細・広視野角映像の撮影技術 ア-3 幾何学的整合性を考慮した映像編集処理技術 ア-4リアルタイム映像作成技術 課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発 イ-1 階層フロー識別型通信品質制御技術の研究開発 イ-2 Ethernetにおける通信品質制御、管理 イ-3 IPネットワークとイーサネットワークの連携技術の研究開発						(株)日立コミュニケーションテクノロジー
間接経費						
合計						

- 注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上(消費税を含む)。
 2 備考欄に再委託先機関名を記載

4-2 研究開発の実施内容

(1) 課題ア 高臨場感ディスプレイ（IPD）用デジタル映像コンテンツ制作に関する基盤技術の開発

平成13年度は、平成14年度に行う中間目標までの研究開発に先立ち、課題項目の詳細化、既存技術調査、主要研究設備の調達等を行った。

平成14年度は、同年度に行う中間発表に向けて、各課題項目の詳細検討、アルゴリズム開発、プロト開発のための設計を主に行った。

各項目の詳細は、以下のとおりである。

(1.1) ア-1 超大容量映像データ処理・管理技術

(1.1.1) 画像処理並列化技術

平成13年度は、画像の幾何変換、色変換、射影変換処理などの画像処理が、ネットワーク接続された複数のPC上で、効率的に分散並列して行われる仕組みの開発のため、下記項目の検討・準備を行った。

- ① 並列化可能な画像処理項目の列挙
- ② 同 実装に必要なアルゴリズム等についての公知技術の調査
- ③ 同 処理を実装するPC等設備の選定と調達

①では、一般のノンリニア編集システムの機能などを調査し、また弊社が過去に行ったデジタル映像コンテンツ制作における画像処理内容などを分析、整理した上で、各処理についての並列化可能性について検討を行った。

②では、上記検討内容に基づき、その処理を実現するのに必要となる公知技術について、画像処理技術書などを対象に調査を行った。

③では、上記検討内容に基づき、その処理を実装する複数のPC等からなるハードウェアシステムを設計し、計算能力、ネットワーク能力、記憶容量、汎用性、価格等の諸条件を考慮した上で機器を選定し、その調達を行った。

平成14年度は、IPDを対象とした映像コンテンツの編集・制作を行うための、映像編集システムプロトタイプを、プロジェクト最終年度までに開発することが、課題ア-1および課題ア-3の最終目標であるが、本項目ではその実現のための要素技術として、画像の幾何変換、色変換、射影変換処理などの画像処理が、ネットワーク接続された複数のPC上で、効率的に分散並列して行われる仕組みについて開発する。本年度は下記項目の検討・設計を実施した。

- ① 全体アーキテクチャの検討、設計
- ② 映像編集・制作に必要な画像処理ライブラリの開発
- ③ 処理の効率的なスケジュール管理手法の検討
- ④ 処理をビジュアルに一元管理できるジョブ管理とそのGUI表示方法の検討、設計

①については、コンテンツ制作の高速化と高信頼化を図ることを目的として、IPD 用コンテンツの制作過程を分析した上で、そこで必要となる画像処理をネットワーク接続された複数 PC で分散・並行して行うための全体アーキテクチャについて検討を行い、個々のハードウェア間の接続構成や全体の処理フローについて設計を行った。また、システム実装のためのクラス設計を完了させた。

②については、まず IPD 用コンテンツ制作に必要な機能抽出・分析した。さらに、画像処理フローについての詳細検討を行い、個別の画像処理に共通なフレームを設計した。その上で、AVI 動画ファイル作成機能、画像拡大機能、他、個々の画像処理機能を実装した。本年度開発を行った画像処理機能は大項目において約 10 である。

③については、全体処理時間の短縮を図るため、主にレンダリング処理と動画圧縮処理の最適配分について検討を行い、その決定方法およびそれに基づく制御方法の開発を行った。

④については、複数の画像処理フローを効率的に管理するために必要となるジョブのスケジューリング機能などについて分析を行い、その分析結果にもとづく管理方法の設計と、それを実装するための GUI の設計を行った。

(1.1.2) 超大容量データストレージ技術・データマネジメント技術

平成 13 年度は、超大容量映像データを効率的に保存・管理する技術開発に必要な、ディスクアレイ、テープデバイス、ネットワーク機器等、設備の選定と調達を行った。

データストレージ手段を、

(a) アクセス性能：中、容量：中 の特徴を持つ保存手段 → ディスクアレイ

(b) アクセス性能：小、容量：大 の特徴を持つ保存手段 → DTF デバイス

(c) アクセス性能：大、容量：小 の特徴を持つ保存手段 → ローカルディスク

の 3 タイプのデバイスに分けて考え、それぞれがネットワークにて接続されるシステムとしての全体設計を行ったのと同時に、各デバイスについて、機能、容量、汎用性、価格等の諸条件を考慮した上で機器を選定し、その調達を行った。

平成 14 年度は、超大容量映像データを、上記画像処理を行う各 PC から、効率的に読み出し、および書き込みできることを最終目的とし、ディスクアレイサーバと、大容量テープデバイス、ならびに前述の PC に置かれたローカルハードディスクの 3 デバイスを用い、実運用を考慮に入れた最適なストレージアーキテクチャおよびその管理方法の検討を行った。

具体的には、一般的なノンリニアシステムとの比較という見地から、IPD 用映像編集システムにおけるストレージに関する課題を抽出した上で、平成 13 年度に選定と調達を行ったストレージ手段を用いた効率的なストレージアーキテクチャの検討を行い、大容量データストレージにおけるデータ管理方法について検討・設計を行った。

(1.2) ア-2 超高精細・広視野角映像の撮影技術

(1.2.1) 複数のビデオカメラを用いた超高精細・広視野角映像撮影技術

平成13年度は、複数のビデオカメラで撮影された映像の射影変換処理によって1枚の超高精細・広視野角の映像を取得する技術の開発のため、下記項目の調査・準備を行った。

- ① 複数のビデオカメラを用いた映像撮影に関する公知技術の調査
- ② ビデオカメラ等の実験設備の選定と調達

①については、現在世界的に注目されている Omni-directional Camera（全方位カメラ）に関する論文などを中心に調査を行った。

また、②については、撮影映像のクオリティ、インタレス・プログレッシブの種別、パン・チルトなどの付加機能などを検討の上で機器を選定し、その調達を行った。

平成14年度は、複数のビデオカメラを異なる方向を向くように配置し、これらによって撮影された映像の幾何変換処理を行い、お互いをシームレスに繋ぎ合わせることで、1枚の超高精細・広視野角の動画映像の取得を可能にするカメラシステムの基礎検討を行った。

光学中心の一致を保証しないマルチカメラ方式として、 6×1 の配置と、 3×2 の配置を持つ2つのタイプの撮影方法を、実験システムの試作により検討・実験した。

また、光学中心を一致させるマルチカメラ方式として、ミラーを用いた撮影方法を、まず、 2×1 の配置による実験システムの試作により、基礎検討・実験を行い、そこで得られた知見をもとに、 3×2 の配置による実験システムを試作し、基本技術の検討・実験を行った。

(1.2.2) 移動カメラを用いた超高精細静止画像撮影技術の研究開発

平成13年度は、移動するカメラによって撮影された連続する複数の異地点映像を、射影変換処理を施しながら擬似的にシームレスに接続する技術の開発に必要となる、カメラ、ロボット等、実験・開発設備の選定と調達を行った。

カメラについては、上記(1.2.1)のカメラを兼用することとし、ロボットについては、カメラを回転させるためのものとして、6自由度固定型ハンドロボット、および移動可能な軽量のものとして4自由度固定型ハンドロボット、またカメラを移動させるものとして、前記ハンドロボットを取り付け可能な大型モーションロボット、および室内用小型3輪モーションロボット、ならびに凹凸のある床面を移動可能な小型4輪モーションロボットを選定し、その調達を行った。

平成14年度は、下記項目を実施した。

- ① ハンドロボットにビデオカメラを取り付け、任意の位置・方向の映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発した。

② 自動走行ロボットにカメラを取り付け、あらかじめ与えられた経路に沿って映像を撮影し、撮影された映像を mosaicing (stitching) 処理により擬似的にシームレスに接続する機能を実現するため、マーカによる画像認識やビジュアルサーボ、ジャイロ等の物理的手段によって、ロボットの姿勢パラメータの認識を行う方法の基礎検討を行い、具体アルゴリズムを開発した。

①については、5 + 1 の配置による小型マルチカメラシステムをハンドロボットに取り付け、撮影画像をシームレスに接続する基礎実験を行った。さらに、撮影と同時に、被写体の位置を3次元位置センサにより測定し、隣り合うカメラ映像の繋ぎ目が最も目立たなくなるようにハンドロボットを制御することで撮影方向を適応的に変更可能な撮影システムに関する手法を提案し、基礎実験を行った。また、ロボットやカメラの位置姿勢パラメータを視覚情報から認識するための射影幾何学に基づく基本アルゴリズムを提案した。

②については、棚に並べられた書籍背表紙の映像情報を、広範囲に、シームレスに、且つ高精細に取得可能とする、自動撮影システムの原理試作、実験を行い、そこで必要となる画像処理アルゴリズムやロボット制御技術について開発を行った。

(1.3) ア-3 幾何学的整合性を考慮した映像編集処理技術

(1.3.1) 映像編集情報管理方式の研究

平成13年度は、映像制作・編集過程における IPD 固有の処理技術の開発のため、下記項目の検討・調査を行った。

- ① スクリーン形状やプロジェクタ構成の違いに対応する IPD 固有の映像編集処理と、それらに関する諸課題の列挙
- ② 課題解決に必要な既存技術の調査

①については、弊社開発済み IPD 映像表示環境を対象とし、その映像制作・編集過程において特に必要となる具体的処理内容について検討し、その要求機能仕様、課題等の洗い出しを行った。

また、②については、SIGGRAPH コースノート 「Computer Graphics for Large-Scale Immersive Theaters」 などを中心とする文献・論文を対象として、既存技術の調査を行った。

平成14年度は、映像制作・編集過程における IPD 固有の処理技術の開発のため、平成13年度の研究にて列挙した、スクリーン形状やプロジェクタ構成の違いに対応する IPD 固有の映像編集処理についての具体アルゴリズムの開発と、処理全体を効率的に管理するためのシステムアーキテクチャの設計を行った。

まず、対象となる IPD 表示システムとしてプロジェクタアレイシステムを想定し、その仕様について分析を行った。またそれと並行して、撮影時のカメラパラメータと表示時のスクリーン形状との不一致性を、表示映像の違和感を少なく補正するために、擬似射影モデルと呼ばれる概念を導入し、スクリーンの形状に柔軟に対応可能な映像変換方式を検討した。これらの結果を踏まえ、本処理を課題ア-1 記載の映像編集システムプロトタイプ

プに実装していくためのシステムアーキテクチャの設計を行った。

(1.3.2) カメラワーク管理方式の研究

平成13年度は、撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪みなど）の扱い方法に関する既存技術の調査を行った。

特に、コンピュータビジョン（3次元画像認識技術）をビジュアルライゼーション（映像制作・表示処理）に応用する技術に関する論文などを対象とした。

平成14年度は、撮影時のカメラパラメータ（撮影位置、撮影方向、焦点距離、光学中心位置、レンズ歪み）などの情報を、映像編集時や表示時に利用することを想定し、映像データに合わせて記録・管理する効果的な手法、ならびに編集時に利用するカメラパラメータ情報にあわせてカメラコントロールを行うための制御方法に関する基礎検討を行った。

具体的には、撮影時にあらかじめ決められたパターンを撮影し、そのパターンの解析を行うことでカメラパラメータを算出することができる、カメラパラメータキャリブレーションに関するアルゴリズムの検討を行った。またこれらのデータを、課題ア-1記載の映像編集システムプロトタイプにおいて、効果的に利用していくためのシステムアーキテクチャについて検討を行った。さらに、与えられたパラメータに従うカメラコントロールを可能とするロボット制御方法についての基礎検討を行い、実機による原理実験を行った。

(1.3.3) 実験コンテンツの制作

平成13年度は、本項目に関する研究開発は、当初の予定どおり実施しなかった。

平成14年度は、課題ア-1、2、3の検証を目的とし平成15年度に開発する実験コンテンツについて、シナリオ、撮影方法、撮影対象、表示方法などの検討を行った。

具体的には、沖縄で撮影した9台のマルチカメラによる360°全方位映像を主な映像素材として、様々なタイプのIPD表示環境に映写可能な映像コンテンツの、シナリオ、エフェクトなどの検討を行い、VHSによる仮編集ビデオを作成した。

(1.4) ア-4 リアルタイム映像作成技術

(1.4.1) 複数カメラによる撮影映像のリアルタイム変換技術

平成13年度は、上記課題ア-2で挙げた開発技術の、リアルタイム化を試みるための、実験設備の選定と調達を行った。

カメラについては性能、価格等を考慮し、画像処理ボードについてはカメラとの親和性や処理速度を考慮し、またPCについては価格を考慮し、その他下記デモンストレーションを実施する際に必要となる機能等を考慮した上で機器を選定し、その調達を行った。

平成14年度は、上記課題ア-2で挙げた各開発技術の、リアルタイム化を実現するためのアーキテクチャの検討、システム設計、実装方法の検討を行った。

具体的には、マルチカメラのキャリブレーションによって得られた隣り合う映像間のシ

ームレス接合のための映像変換パラメータと、IPD 表示環境におけるマルチプロジェクタの隣り合う映像間のシームレス接合のための映像変換パラメータを統合する仕組みを開発し、この統合されたパラメータに基づき、表示映像のリアルタイム映像変換処理を行うことで、マルチカメラ・マルチプロジェクタによる、シームレスリアルタイム映像表示システムを実現させるためのシステム設計ならびに、原理実験を行った。

(1.4.2) マルチカメラによる遠隔監視システム プロトタイプの開発

平成13年度は、アプリケーション技術の検討およびその検証を目的とした実験アプリケーションを開発するため、下記項目の検討・準備を行った。

- ① アプリケーションデモシステムの全体構想策定
- ② 主要設備の選定と調達

①については、主に機器選定のためのハードウェア構成策定を目的とし、検討を行った。複数のカメラから撮影された映像を複数の PC に取り込み、これをデジタルデータとしてギガビットネットワークで別の複数の PC に送信する。データを受け取った PC は、それぞれ、IPD 映像表示環境の構成に従った適切な映像変換を行う画像ボードを介して、プロジェクタに映像を出力する。以上のような機能・構成を、アプリケーションデモシステムとして、現時点（研究開始時点）では想定するものとした。なおこれらの仕様は、上記課題ア-1～3の研究の結果によって適宜変更されるものとする。

②については、上記デモシステムの仕様・構成を考慮した上で機器の選定を行い、その調達を行った。特に、既存の IPD 映像表示環境（プロジェクタアレイシステム）との接続性について留意した。

平成14年度は、課題アおよび課題イを組み合わせたアプリケーション技術の検討ならびに検証を目的とした実験アプリケーション開発のため、全体システムの設計および、実装に際し必要となる応用技術の検討と開発を行った。

具体的には、課題ア-2で試作した3×2の配置によるマルチカメラシステムで撮影した映像を、同じく3×2の配置のプロジェクタアレイシステムによる表示環境に、ギガビットネットワークを通してリアルタイムに転送し、上記(1.4.1)で開発したリアルタイム映像変換処理を行うことで、シームレス表示可能とするアプリケーションシステムの、全体システム設計と、各コンポーネントで必要となる機能についての検討を行い、全体システムの原理実験を行った。

(2) 課題イ 超広帯域且つ超高品質ネットワーク制御に関する基盤技術の研究開発

平成13年度は、平成14年度からの技術開発、プロトタイプ開発に先立ち、課題項目の基礎検討、既存技術調査、主要研究設備の調達を行った。また、課題イのテーマ全体の技術開発傾向を把握する目的で、「イーサネット／IP ネットにおける通信品質」に関する特許調査を行った。

平成14年度は、平成13年度に実施した技術調査、方式検討、および主要研究設備の調達等の結果に基づき、方式検討、機能設計、詳細設計およびプロトタイプ実装を行った。各項目の詳細は、以下のとおりである。

(2.1) イ-1 階層フロー識別型通信品質制御技術の研究開発

(2.1.1) 通信品質制御技術

平成13年度は、本委託業務では、次世代ネットワークサービスの通信品質制御条件を整理するため、以下の既存技術の調査・整理を行うとともに実験設備の調達ならびにネットワーク構築を行った。

- ① 今後要求されるコンテンツ対応の通信品質に関する調査と整理
- ② ネットワークの End-to-End の通信品質制御技術に関する調査と整理
- ③ 実験設備の調達ならびにネットワーク構築

①では、ネットワークサービスの通信品質の要件項目を調査整理した後、次世代ネットワークで主流になると予想される、映像配信（伝送）システム、IP 電話システム、TV 会議システムの仕様及びシステム品質要件について調査した。最後に、上記の調査結果を元に、今後のコンテンツ対応の通信品質要件について検討した。

②では、ネットワークで現状使用されている、または、検討されている品質制御技術について調査整理した。具体的には、エンドホスト上及び中継ノードに実装される品質制御機能について、プロトコル技術、パケット制御技術を中心に調査、整理した。

③では、本研究で開発する通信品質制御機能、及び管理技術を評価、検証するための実験設備の調達及びネットワーク構築を行った。

平成14年度は、通信するコンテンツに対応し、ネットワークの End-to-End での通信品質を制御する技術の開発のため、下記項目の検討・設計・プロトシステム開発を行った。

- ① レイヤ2およびレイヤ3レベルでの通信ノード（スイッチ、ルータ等）におけるアプリケーション対応フロー識別方法の検討及び設計
- ② 通信ノード（スイッチ、ルータ等）でのアプリケーション対応フロー識別を可能にする制御プロトプログラムの開発

①では、前年度の調査結果に基づいて PC 上でのプロトプログラム開発を見据えたアプリケーション対応フロー識別方法の検討及び設計を行った。

②では、上記設計に基づき、プラットフォームとして FreeBSD を採用し、PC 上でプロトプログラムの開発を行った。

(2.1.2) アプリケーション対応通信品質制御技術

平成13年度は、通信品質制御方式の下記項目について検討を実施した。

- ① 階層フローの定義および検出手段の検討
- ② 階層フロー通信品質制御機能の実装方式案

①では、フロー検出に用いることができる情報を、レイヤー別に整理し、それらを用いた新しい階層フローの定義方法を検討した。

②では、そのような品質制御機能に基づいた、ユーザー間のフェアな帯域分配とフロー種別に基づいた品質制御方式案を検討した。

平成14年度は、アプリケーションとして超大容量映像データ配信システム（放送型）を想定し、超大容量映像データの通信トラヒックと、該映像データ放送の複数観客からの反応情報返送トラヒックが混在する場合の、各トラヒックに対する通信品質制御技術の開発のため、下記項目の検討・設計・プロトシステム開発を行った。

- ① 従来の通信レイヤ（レイヤ3以下）の通信品質制御に加え、通信データの情報処理を含む上位レイヤでの通信品質制御方式の検討
- ② ネットワーク上に配備する上位レイヤでの通信品質制御対応情報処理サーバの配備方式の検討及び設計
- ③ 情報処理サーバのプロトプログラム開発

①では、前年度の調査結果に基づき、また (2.1.1) ①で行ったアプリケーション対応フロー識別技術の検討結果を利用して上位レイヤでの通信品質制御方式の検討を行った。

②では、前年度の調査結果に基づき、今回プロトプログラムの開発を行った情報処理サーバの配備方式の検討及び設計を行った。

③では、①での設計に基づき、プラットフォームとして FreeBSD を採用し、PC 上でプロトプログラムの開発を行った。

(2.2) イー2 イーサネットにおける通信品質制御、管理技術の研究開発

平成13年度は、次世代ネットワークサービスの通信品質制御条件を整理するため以下の既存技術の調査・整理を行うとともに実験設備の調達ならびにネットワーク構築を行った。

- ① IP ルータにおける通信品質制御技術の調査と整理
- ② イーサスイッチにおける通信品質制御技術の調査と整理
- ③ 実験設備の調達ならびにネットワーク構築

①では、IP ルータにおける通信品質制御技術としての、包括的なフレームワークについて調査を行い、整理した。②では、イーサスイッチにおける通信品質制御技術として、IEEE での活動状況を調査し、整理した。

現在市販されている製品について、実用化されている技術の下記調査を行った。

①では、IP ルータにおける通信品質制御技術の種別を調査し、代表的なメーカーの主要な製品における実装状況を整理した。また、IP ルータの主要部品となりつつある、ネットワークプロセッサの製品動向についての調査も行った。②では、イーサスイッチにおける通信品質制御技術の種別を調査し、代表的なメーカーの主要な製品における実装状況

を整理した。

③では、本研究で開発する通信品質制御機能、及び管理技術を評価、検証するための実験設備の調達及びネットワーク構築を行った。

また、イーサネットにおける通信品質制御・管理方式の下記検討を行った。

既存イーサスイッチの通信制御技術の調査結果を受けて、管理装置からイーサネットワークを管理するために必要な、イーサスイッチのリモート制御について、下記項目について検討した。

- ① イーサスイッチ制御方式
- ② イーサネットワーク管理装置の実装方式案

具体的には前項での調査結果に基づき、管理装置からイーサスイッチ制御を実現する為の方式検討を行った。また、イーサネットワークをリモート制御する管理装置の実装方式案を検討した。

平成14年度は、今後の高速広域ネットワークの主流になると予想される広域イーサネットにおける通信品質制御、管理技術を開発するため、下記項目の検討・設計・プロトシステム開発を行った。

- ① 通信品質保証ネットワーク管理装置からイーサネットワークを管理するために必要な、イーサスイッチのリモート制御について、ワイドエリアおよびローカルエリア・イーサスイッチの機種に依存しない制御方式の設計
- ② 通信品質保証ネットワーク管理装置のプロトシステム開発

①では、前年度の調査結果に基づき、ワイドエリア、ローカルエリアで必要となる管理内容を検討し、管理対象を抽出して、イーサスイッチの機種に依存しない制御方式の設計を行った。

②では、①での設計に基づき、レイヤ3ネットワーク管理装置であるVPNサーバの拡張機能として、課題イ-1で開発したL2Shaperと、Extreme社のイーサスイッチを対象にイーサネットワークを制御するプロトシステムの開発を行い、また、イーサネット網データ管理及びサービス品質管理業務を支援する管理装置であるL2meterのプロトシステム開発を行った。

(2.3) イ-3 IPネットワークとイーサネットワークの連携技術の研究開発

平成13年度は、次世代ネットワークサービスの通信品質制御条件を整理するため以下の既存技術の調査・整理を行った。

- ① IPネットワークとイーサネットの連携による通信品質制御技術の調査と整理

IPレベルでの通信品質制御パラメータとイーサネットレベルの通信品質制御パラメータとの間の整合性の取り方について検討した。

平成14年度は、IPネットワークとイーサネットワークが接続され、かつ、レイヤ3以下の通信品質制御とレイヤ4以上の通信品質制御が連携する次世代のネットワークサービス対応通信品質制御技術を開発するため、下記項目の調査・検討を行った。

- ① 超大容量映像データを駆使する次世代ネットワークサービスの調査
- ② IPネットワークとイーサネットワークの通信品質制御連携方式の検討
- ③ 情報処理機能と通信品質保証機能を有する次世代ネットワークの機能要件及び連携方式検討

①では、市場調査を実施し、ユーザニーズの調査を行った。

②では、13年度に整理したIPレベルでの通信品質制御パラメータとイーサネットレベルでの通信品質制御パラメータ間の整合性を基に、IPネットワークとイーサネットワークの通信品質制御連携方式の検討を行った。

③では、課題イ-1、2の通信品質制御方式を弊社ブロードバンドアクセスサーバAG8000に適用するための基本検討と、さらにAG8000のリモート制御に関する基本検討を行った。

5. 研究開発実施状況（平成14年度）

5-1 課題ア 高臨場感ディスプレイ（IPD）用デジタル映像コンテンツ制作に関する基盤技術の開発

5-1-1 課題ア-1 超大容量映像データ処理・管理技術

5-1-1-1 はじめに

超大型スクリーンやマルチ（多面）スクリーン、柱面スクリーンやドーム型球面スクリーンといった特殊表示環境に対して、高精細・広視野角の映像を投影し、よりリアルで豊かな臨場感を持った映像空間を提供しようとする高臨場感ディスプレイ（IPD: Immersive Projection Display）が注目されている。

しかし、IPD用コンテンツを制作しようとした場合、投影する映像が高精細・広視野角であるため、取扱う映像素材も高解像度・大容量なものとなり、その加工時間は膨大なものとなる。通常、それら映像素材と加工後の映像データは、ハードディスクなどの記憶媒体上に置かれる。そのため、この記憶媒体の容量が不足することも少なくなく、コンテンツ制作中に記憶媒体への画像データの入替え作業が頻繁に発生することも少なくない。

本稿では、映像素材の加工時間を短縮するために、ネットワークに接続された複数の計算機を利用した画像処理の並列化技術の検討を行う。また、ネットワークで接続された複数の計算機に対する映像データの供給や、映像素材や加工後の映像デー

タが置かれる記憶媒体への画像データの入替え作業の効率化を図るために、超大容量データのマネージメント技術についての検討を行う。

5-1-1-2 実施内容

第1部 映像制作過程における画像処理並列化に関する研究

1. IPD 用コンテンツ制作作業の概要

IPD 用コンテンツ制作作業は、基本的に、ビデオや映画の制作で行われるノンリニア編集作業である。ノンリニア編集とは、映像をリニアな媒体であるテープではなく、ランダムアクセス可能なハードディスクなどの媒体上で編集することである。この場合、映像データは、映画フィルムの1コマ1コマをデジタル静止画像に置き換えたような、静止画像の時系列データとして与えられる。通常、映像データを構成する1枚1枚の静止画像には、先頭から順番にフレーム番号と呼ばれる数字が付けられる。

本稿では、静止画像の時系列データのことを映像または映像データと呼び、映像データを構成する1フレーム1フレームの静止画像のことを画像または画像データと呼んで区別することにする。

静止画像のデジタル表現には、ラスタ型データ表現とベクトル型データ表現とがある。ラスタ型データ表現とは、画像を画素の2次元配列として表現するもので、ベクトル型データ表現とは、画像を構成する点や線、面、文字などの構成要素を表すものである。一般に画像処理の対象となるデジタル静止画像は、ラスタ型データ表現によるものであり、本システムでも、ラスタ型データ表現によるデジタル静止画像を処理対象とする。

通常のビデオや映画などと異なり、IPD では映像を投影するスクリーン形状が特殊で広視野角である。これに対応するために、通常の透視投影映像だけでなく、より広角の映像が得られる柱面投影映像や魚眼投影映像などを用いてコンテンツ制作を行う。このような映像を取扱うために、特殊な射影変換が必要となる。

さらに、ビデオや映画の制作では、その編集結果をビデオテープや映画フィルムに記録するが、我々が想定している IPD 用コンテンツ制作作業では、編集結果として得られたデジタル静止画像を AVI(Audio Video Interleave) 形式などの動画圧縮ファイルに変換する。

以下、コンテンツ制作の作業の流れと、そこで使用される画像処理フローの編集方法について述べる。

1.1 コンテンツ制作作業の流れ

図 1にコンテンツ制作作業の流れを示す。

通常、コンテンツを制作する場合には、最初に絵コンテや設計書などでシナリオが作成される。そして、それを元に必要な映像データが準備される。映像データは、映画会社や映像制作会社に依頼して新規に撮影あるいはCGを作成する場合や、過去のコンテンツ制作で使用したものを再利用する場合などが考えられる。



図 1 コンテンツ制作作業の流れ

このように準備された映像データを組み合わせ、作成されたシナリオどおりのコンテンツを仕上げるために、オペレータはノンリニア編集システムを使用して映像データを加工する。以下、ノンリニア編集システムに与えられる映像データを入力映像データとし、ノンリニア編集システムで作成される映像データおよび動画圧縮データを出力映像データおよび出力動画圧縮データとする。

ノンリニア編集システムにおける映像データの加工は、画像処理フローを用いて行われることが多い。画像処理フローとは、入力映像データから出力映像データを作るための設計書にあたるもので、図 2 にその例を示す。この例は、入力映像データに対してカラーバランス調整処理を行った後、拡大処理を行い、さらに回転処理を行う画像処理フローである。通常、それぞれの画像処理は、その処理特有のパラメータを持っている。例えば拡大処理であれば、倍率を調節するパラメータを持っているだろうし、回転処理であれば回転角度を調節するパラメータを持っている。さらに、これらのパラメータは、フレームの進行に合わせて変化させることができるのが普通である。

また、例に示した画像処理フローでは、出力映像データ書込処理が1回しか無く、1フレームごとに1枚の静止画像データが出力される。しかし、1つの画像処理フロー中に複数回の出力映像データ書込処理がある場合には、1フレームごとに複数枚の静止画像データが出力される。

このように、画像処理フローの内容は、入力映像データの場所、画像処理の種類とその実行順序、個々の画像処理が持つパラメータの値などの情報からなる。通常、これら画像処理フローの内容はファイルに保存され、これを画像処理フロー設定ファイルと呼ぶ。

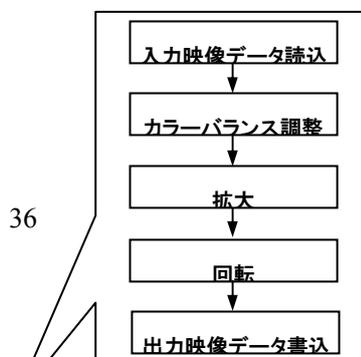


図 2 画像処理フローの例

制作しようとするコンテンツに対して、作成される画像処理フローの数は、最も単純なものでも1つ、複雑なものになれば数十個に達する。1つのコンテンツに対して複数の画像処理フローが作成される場合には、シーンごとに別々の画像処理フローが作成されることが多い。

作成した全ての画像処理フローをノンリニア編集システムで実行することによりコンテンツが作成されるが、複数の画像処理フローが作成された場合には、個々の画像処理フローの出力結果を合せて、1つのコンテンツができあがる。

ただし、オペレータの操作ミスや思い違いによって、シナリオどおりの加工ができない場合や、シナリオどおりの加工ができたとしても実際に完成したコンテンツをスクリーンに投影した段階で問題や変更が生じる場合も多い。そのような場合には、問題や変更が発生した部分の画像処理フローを修正したり、新たに作成し直すなどして再実行する。

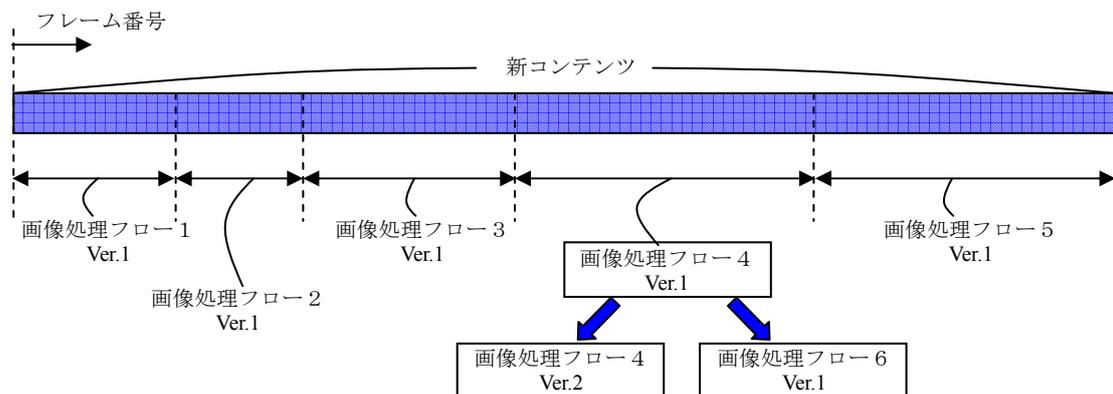


図 3 コンテンツの調整作業

上記のような試行錯誤を繰り返してコンテンツが完成する。また、作業内容のバックアップを保管することが重要で、その際に保管するデータとしては、入力画像データ、出力画像データおよび出力動画圧縮データ、画像処理フロー設定ファイルなどがある。

1.2 画像処理フローの編集

オペレータが、画像処理フローを編集する際には、画像処理の追加や削除、個々の画像処理が持つパラメータ値の変更など、画像処理フローの出力結果に変更が生じる操作を行うたびに、画像処理フローを実行して出力画像を作成する。そして、その変更内容がオペレータの意図したものであるかどうかを確認しながら作業を進める。

一般に、画像処理フローを実行して出力画像を作成することをレンダリングと呼ぶ。画像処理フローの編集時に行われるレンダリングでは、入力映像データや出力映像データの解像度を小さくしたり、変更を加えたフレームに対してだけ実行するといった簡易化を図る。このようなレンダリングを簡易レンダリングと呼んで、編集を終えた画像処理フローに基づいて最終的な出力映像データを作成する本レンダリングと区別する。

2. 画像処理並列化の全体アーキテクチャの検討、設計

ここでは、コンテンツ制作作業の高速化と信頼性の向上を図ることを目的とし、システム内の複数のPCに画像処理を分散して並列処理を行わせるための全体アーキテクチャの検討および設計を行う。

前述したように、簡易レンダリングについては、処理する映像データの解像度を小さくするなどの工夫によって、処理時間を短縮することが可能である。したがって、コンテンツ制作時間を短縮するには、本レンダリングの処理時間を短縮する必要がある。

通常、ノンリニア編集システムのレンダリングにおいて使用される画像処理には、カラーバランス調整やレベル調整などの色調整処理、平行移動や回転や拡大・縮小などの幾何変換処理、クロマキーによる合成処理など、様々なものが挙げられる。これらの処理は全てフレーム番号の順序に関係無く実行可能な処理であり、フレーム毎の処理を並列化することが可能である。

一方、フレーム番号の順序に依存する処理としては特徴点追跡処理が挙げられる。これは、あるフレーム番号の画像に写っている特徴的なオブジェクトや領域の位置がフレーム番号の進行と共に、どう変わっていくかを調べる処理であり、主に、映像の揺れを止めたり、あるオブジェクトの動きに合わせて画像の合成を行いたい場合などに使用される。しかし、この処理は、追跡を行っていたオブジェクトや領域が他のオブジェクトの裏に入り込んで見えなくなるなどにより、追跡を失敗することが多い点や、主に他の画像処理のパラメータを設定するために使用されることなどから、簡易レンダリングや本レンダリング時には、画像処理フロー上に存在していたとしても機能しないように設計されていることが多い。

従って、本稿においても、レンダリング時に有効に機能する画像処理は、フレーム番号の順序に関係なく実行可能な処理に限定することにする。

一方、先にも述べたように、我々が想定しているIPD用コンテンツ制作作業では、画像処理フローを実行することによって静止画像の時系列データを作成したあと、最終的にはこれをAVIなどの動画圧縮ファイルに変換する。

動画圧縮処理をレンダリング処理の一部と考えることもできるが、我々がこれらを明確に分ける理由は、両者には以下に述べるような特性上の違いがあるためである。

まず、並列化処理の可能性の観点からの違いを見てみると、前述したようにレンダリング処理は、フレーム番号の順序に関係なく実行でき、フレーム毎の処理を並列化できるのに対し、動画圧縮処理は、圧縮効率を上げるため隣り合うフレーム間の相関を利用して圧縮を行うために、フレーム番号の順番に処理する必要がある。ただし、例えば、フレーム番号1番～200番までの動画圧縮ファイルを作成しようとするときに、1番～100番までの動画圧縮ファイルと101番～200番までの動画圧縮ファイルを作成して、それらを接続するというように、部分的な動画圧縮ファイルを作成しておき、これを接続することで全体の動画圧縮ファイルを作成することは可能である。つまり、部分的な動画圧縮ファイルの作成処理は並列化が可能である。この際、圧縮効率を低下させないためには、数十～数百フレーム程度の部分動画圧縮ファイルとするのが現実的だと思われる。

次に、処理負荷の観点からの違いを見てみると、レンダリング処理は、個々の画像処理が持つパラメータの値をフレームごとに変化させることができるため、フレームごとに負荷が大きく変わる可能性がある。これに対して、動画圧縮処理では、処理する画像の解像度が決まってしまうと、フレームごとに負荷が大きく変わることはないと考えられる。

以降、2.4節において、レンダリング処理を並列実行する方法について検討し、2.5節において、レンダリング処理と動画圧縮処理を複数の PC に配分する方法について検討する。

2.1 システム構成

本システムの構成を図 4に示す。合計21台のPCが1000BASE-TのLANに接続され、後述するPCクラスタを構成している。うち1台は制御用(制御PC)であり、他の20台(レンダリングPC)は、主にレンダリング処理や動画圧縮処理を行う。制御 PC は Pentium3 1.4GHzの CPU 2台、256MByteのメモリ、36.8GByteの内臓ハードディスクを有し、OS はWindows 2000 Serverである。また、この制御 PC には、1.3TByteの RAID5 ハードディスクとテープライブラリが接続されている。テープライブラリは、内部にアクセス速度24MB/secのテープドライブを搭載し、さらに、1巻あたり200GByteの記憶容量を有するテープを35本搭載可能である。一方、レンダリング PC は、Pentium4 2.0GHzの CPU 、160GByteの内臓ハードディスクを有し、OS はWindows 2000 Professionalである。20台のレンダリング PCのうち6台には2GByte、残りの14台には512MByteのメモリがそれぞれ搭載されている。

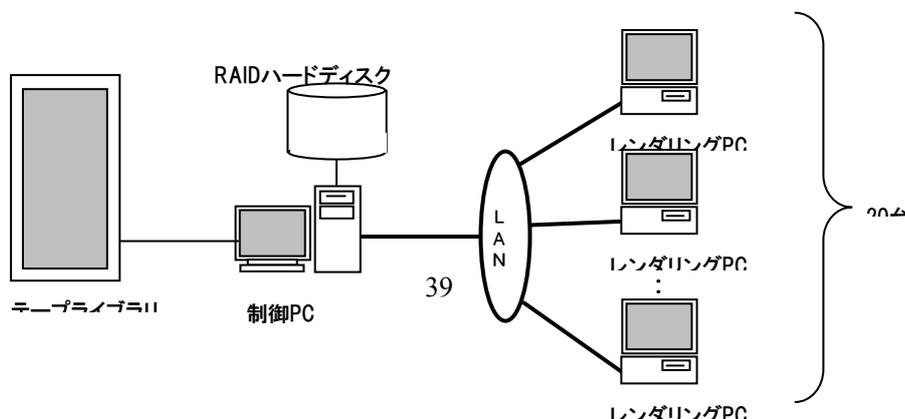


図 4 システム構成

2.2 PC クラスタ

クラスタとは、複数の計算機をネットワークで接続して構築した並列計算機の一つである。クラスタでは、ネットワークで接続された1台1台の計算機のことをノードと呼び、このノードにパーソナルコンピュータ（PC）を使用したものをPCクラスタという。

近年、PCで使用される汎用部品は、著しく高性能化、低価格化しており、スーパーコンピュータにおいても開発コストの問題により、こうした汎用部品が使われるようになってきている。したがって、PCクラスタでもチューニング次第では、スーパーコンピュータと同程度の性能を発揮させることが可能で、現に世界のスーパーコンピュータのランキングを紹介している Super Computer Top 500 において、PCクラスタがランキング入りすることも珍しくない[Doshisha2]。

ノードにPCを採用することによる利点として、新規ハードウェアの開発が不要であること、量産されていることにより低価格であること、アーキテクチャが標準化されているためハードウェアを選択する際の幅が広いこと、さらには、比較的簡単にCPUなどを交換することができ、短期間にシステムが陳腐化してしまうのを防げることなどが挙げられる[手塚 1996]。

2.3 処理を並列化する際の問題点

一般に、処理を並列化する際には、各プロセッサに与える処理の粒度が問題となる。粒度とは複数のプロセッサに与えるタスクの大きさのことである。この処理の粒度を決める場合、以下に示す点に注意する必要がある。

2.3.1 Amdahl の法則[SSE] [Doshisha1]

Amdahl の法則とは、元 IBM のエンジニアであった Gene Amdahl が1967年に提唱した法則であり、並列可能な処理と不可能な処理との割合によって、並列化による処理速度向上の限界が決まってしまうというものである。

今、ある処理全体の $s\%$ を n 倍高速化することができたと仮定する。これにより、処理速度が v から v' になり、処理時間が t から t' になったとする。この時、速度向上率 e_{ff} は次式で表される。

$$e_{ff} = \frac{v'}{v} = \frac{t}{t'} = \frac{t}{t((1-s) + \frac{s}{n})} = \frac{1}{(1-s) + \frac{s}{n}} \quad \dots (1)$$

全体の処理に対する並列化可能な処理の割合 s を0.5～0.95の間で変化させ、並列化可能な処理の高速化倍率 n を1～10000倍まで変化させた時の速度向上率 e_{ff} を図 5に示す。

処理を並列化したことによるオーバーヘッドが全く無い理想的な状況を考えると、並列化可能な処理の高速化倍率 n は、使用するプロセッサ数に等しいと考えられる。

しかし、このような理想的な状況においても、並列化不可能な処理がわずか5%あるだけで10000個のプロセッサを使用して、処理速度が約20倍向上するだけである。

このことから、全体の処理に対する並列化可能な処理の割合ができるだけ多くなるように、処理の粒度を決める必要がある。

一般に並列処理では、処理を並列化したことによるオーバーヘッドが生じ、並列化可能な処理の高速化倍率 n は、使用するプロセッサ数に等しくならない。オーバーヘッドの主な原因は、以下に示すプロセッサ間通信とロードバランスである。

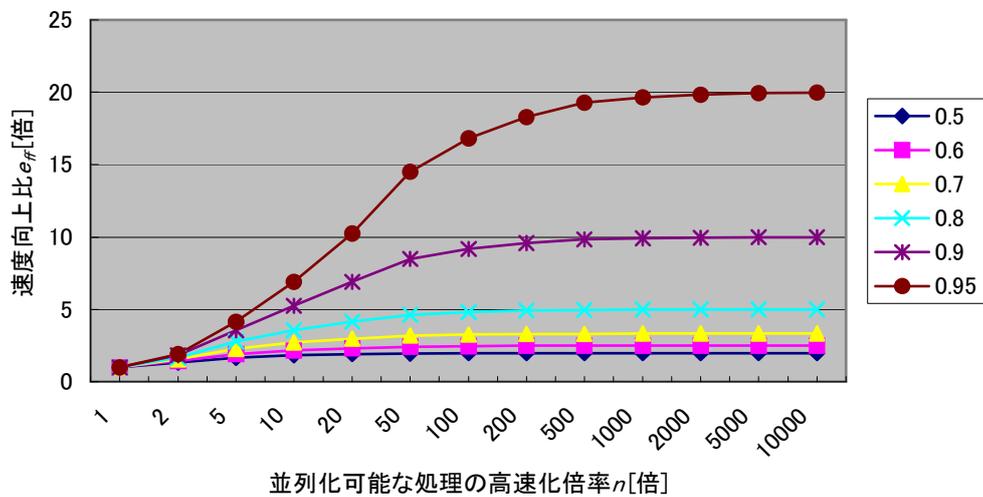


図 5 Amdahl の法則

2.3.2 プロセッサ間通信

プロセッサ間通信とは、分割した処理を各プロセッサに送ったり、処理結果などの情報を各プロセッサ間でやり取りするために必要な通信である。

プロセッサ間通信の頻度は、各プロセッサに与える処理の粒度に関係する。処理の粒度を小さく設定すれば、プロセッサ間通信は多くなり、処理の粒度を大きく設定すれば、プロセッサ間通信は少なくなる。

一般に PC クラスタでは、プロセッサ間通信速度が遅いため、処理の粒度を小さく設定すると複数のプロセッサを使用しているにもかかわらず、1台のプロセッサを用いたときよりも遅くなることもある[Doshisha1]。

2.3.3. ロードバランス

ロードバランスとは、並列計算機の各プロセッサに与えられる処理負荷の均一性のことである。各プロセッサに与えられる処理負荷が不均一で、各プロセッサで行われる処理の終了タイミングに違いがあると、負荷が軽いプロセッサは、負荷の重いプロセッサに待たされることになり、効率を著しく悪化させてしまう。

ロードバランスの取り方は、処理粒度の取り方によって変わってくる。そのため、ロードバランスを取りやすい処理粒度を選ぶことが重要となる。

2.4 レンダリング処理の並列処理方法

前回の報告の中で、レンダリング処理を、クラスタ内の複数の PC で並列処理する場合、以下の4つの方法があることを示した。そして前回の報告では、それぞれの方法について主にプロセス間通信量に着目した検討を行った。本報告では、処理全体に対する並列化可能な処理の割合や、ロードバランスの取り易さを含めて再検討を行う。

[レンダリング方法1] 各 PC は、1つの画像処理の実行を担当する。処理対象データは、入力画像1フレームを単位とする。

[レンダリング方法2] 各 PC は、1つの画像処理の実行を担当する。処理対象データは、入力画像1フレームを適当な大きさに分割した部分領域を単位とする。

[レンダリング方法3] 各 PC は、複数の画像処理の実行を担当する。処理対象データは、入力画像1フレームを単位とする。

[レンダリング方法4] 各 PC は、複数の画像処理の実行を担当する。処理対象データは、入力画像1フレームを適当な大きさに分割した部分領域を単位とする。

それぞれの方法について図 3に示した画像処理フローを用いて述べる。

レンダリング方法1は、PCごとに実行する画像処理を決め、個々の画像フレームに対する当該画像処理だけを行わせる。例えば、図 6に示すように、ある PC がカラー

バランス調整処理を行い、他の PC が拡大処理を行うというように決められる。したがって、この方法では、カラーバランス調整処理を行うPCに入力画像データが与えられ、その処理結果が、拡大処理を行うPCに渡されるというように、各 PC 間を画像データが移動しながら、画像処理フローが進行する。この方法は、複数フレームのレンダリングを行う場合に、実行速度を向上させる可能性がある。

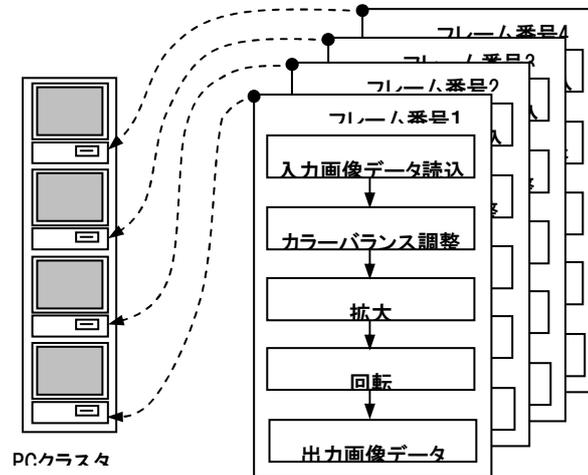


図 6 レンダリング方法 1

レンダリング方法2は、レンダリング方法1と基本的に同様であるが、各 PC を移動するデータが、レンダリング方法1に比べて小さくなったものである。レンダリング方法1では、各 PC にフレーム番号の異なる画像が与えられるのに対して、この方法では、フレーム番号が同じ画像の部分領域が与えられる。この方法は、1フレームのレンダリング実行速度を向上させる可能性がある。

レンダリング方法3は、1つの PC に1フレームの画像データに対する複数の画像処理を行わせる（図 7）。レンダリング方法1および2では、画像データを PC 間で移動させながら画像処理フローを進行させるが、この方法では、画像データの移動は行わず、ある PC に与えられた入力画像データに対して、最初にカラーバランス調整処理を行い、同じ PC 上で、今度は、その出力画像データに対する拡大処理を行うというように、画像処理を次々と変えることにより画像処理フローを進行させる。この方法も、レンダリング方法1と同様、複数フレームのレンダリングを行う場合に、実行速度を向上させる可能性がある。

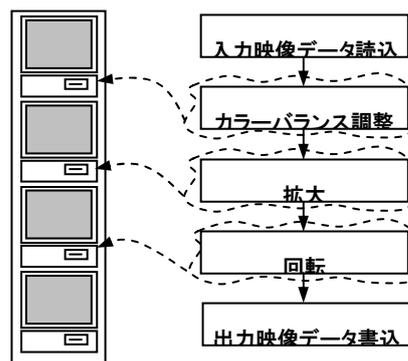


図 7 レンダリング方法3

レンダリング方法4は、基本的に同様であるが、各 PC に与えられるデータが、レンダリング方法3に比べて小さくなったものである。レンダリング方法3が、複数の PC を用いて複数フレームの画像を処理するのに対して、この方法は、複数の PC を用いて1フレームの画像を処理する。この方式は、1フレームのレンダリング実行速度を向上させる可能性がある。

2.4.1 処理全体に対する並列化可能な処理の割合

ここでは、それぞれのレンダリング方法で、処理全体に対する並列化可能な処理の割合にどのような違いがあるかを考える。

前回の報告では、ノンリニア編集においてよく使用される画像処理のアルゴリズムを示し、それら1つ1つの画像処理を単体で並列化しようとした場合、SPMD (Single Program Multiple Data) 型、および SPMD 型とパイプライン型[松山1995]の組み合わせで実現されることを述べた。

レンダリング方法1および3では、画像処理フロー中の各画像処理は、1フレームの画像データを1つの単位として行なわれる。このため、その画像処理実行中に他の PC と情報をやり取りする必要はなく、各 PC の処理は完全に独立して行われる。したがって、全ての画像処理が並列化可能である。

レンダリング方法2および4では、画像処理フロー中で使われる全ての画像処理が、SPMD 型で並列化される場合、各部分領域の処理は完全に独立であり、全てが並列化可能である。しかし、SPMD 型とパイプライン型の組み合わせで並列化される画像処理が含まれていた場合、並列化不可能な部分が生じる可能性がある。例えば、画像のヒストグラムを利用してコントラストを調整する画像処理を考えた場合、各 PC に与えられた部分領域を用いて局所的なヒストグラムを作成する。この局所的なヒストグラムの作成処理は、SPMD 型の並列化で実現できる。次に、複数の局所ヒストグラムから大局的なヒストグラムを作成する。この処理はパイプライン的な並列化で実現できる。次に、この大局的ヒストグラムを用いて、変換前と変換後の画素値の対応表を作成する。しかし、この対応表の作成は、並列化不可能だと考えられる。最後にこの対応表を用いて、各 PC に与えられた部分領域の画素値の変換を行う。この処理は、SPMD 型の並列化で実現できる。

このように、レンダリング方法2および4では、画像処理フロー中で使用される画像処理の種類により、並列化可能な処理の割合が変わってくる。

2.4.2 プロセス間通信の頻度

ここでは、各レンダリング方法のプロセス間通信の頻度について考える。一般に処理の粒度が小さいほどプロセス間通信は多くなり、大きいほどプロセス間通信は少なくなる。

1フレームの画像データを1つの単位として処理するレンダリング方法1および3と、1フレームの画像データを複数の部分領域に分けて、その部分領域を単位として処理するレンダリング方法2および4を比べた場合、1フレームの画像を処理するために発生するプロセス間通信は、データの引き渡し頻度が増す後者の方が多くなることは容易に理解できる。

4つのレンダリング方法のうち、レンダリング方法1, 2は、PC ごとに実行する画像処理を決め、各 PC 間を画像データが移動することによりレンダリングが進行する。そのため、画像処理フロー中で使用される画像処理が増えると、その分、プロセス間通信が増えることになる。

また、レンダリング方法2および4のように、1フレームの画像を部分領域に分けて処理する場合、画像処理をアルゴリズムレベルで並列化することになり、その画像処理が、SPMD 型とパイプライン型の組み合わせで実現される場合が出てくる。パイプライン的な処理を実行する場合には、プロセス間通信が必要である。そのため、画像処理フロー中で使用される画像処理の種類によっても、プロセス間通信の頻度が変わってくる。

以上から、プロセス間通信の頻度が多いと考えられるものから順に並べると、レンダリング方法2、レンダリング方法4、レンダリング方法1、レンダリング方法3となる。

2.4.3 ロードバランス

ここでは、各レンダリング方法におけるロードバランスについて考えてみる。まず、処理粒度を小さくしたほうが、きめ細かな調整が可能であるが、制御は複雑になると考えられる。

同程度の処理粒度で考えた場合、ロードバランスの取り易さを判断することは難しく、レンダリング方法1と3、2と4を比べた場合、どちらが良いかを明確に判断することはできない。

以上の結果をまとめると、表 1、表 2のようになる。表 1より、複数フレームの処理に対する高速化を行う場合、レンダリング方法3が良いと考えられる。また、表 2より、1フレームの処理に対する高速化を行う場合、レンダリング方法4が良いと考える。

複数フレームの画像を処理することが多い本レンダリングでは、レンダリング方法3が効果的だと考える。

表 1 検討結果（複数フレームのレンダリングの高速化）

	全体の処理に対する並列化可能な処理の割合	プロセス間通信の頻度	ロードバランス
レンダリング方法1	○	×	△
レンダリング方法3	○	○	△

表 2 検討結果（1フレームのレンダリングの高速化）

	全体の処理に対する並列化可能な処理の割合	プロセス間通信の頻度	ロードバランス
レンダリング方法2	×	×	△
レンダリング方法4	×	○	△

2.5 レンダリング処理と動画圧縮処理の配分方法

以下では、複数フレームの画像を処理する場合に、本レンダリングの処理時間短縮を目的として、レンダリング処理と動画圧縮処理を、効率的に実行する方法について考える。複数フレームを対象とするので、ここでのレンダリング処理とは、レンダリング方法3を採用したものである。

レンダリング処理と動画圧縮処理の配分方法には、以下の2つが考えられる。

[処理の配分方法1] 全ての PC にレンダリング処理と動画圧縮処理を行わせる。

[処理の配分方法2] いくつかの PC にはレンダリング処理を行わせ、残りの PC には動画圧縮処理を行わせる。

配分方法1と2では、各 PC で行わせる処理の違いにより、入力画像の与え方が変わってくる。

配分方法1では、レンダリング処理と動画圧縮処理の両方を1つの PC 上で行う。あるPC でレンダリング処理された画像は、同じ PC 上で動画圧縮ファイルに変換した方がプロセス間通信も少なく効率的である。動画圧縮が圧縮効率を上げるために隣り合うフレーム間の相関を利用することを考慮すると、それぞれの PC に、フレーム番号が連続する複数の画像フレームを割当てるのが得策である。なお、配分方法1では、レンダリング処理と動画圧縮処理を同時に実行することも可能であるが、メモリの不足などが生じ処理効率を落とすことも考えられるため、ここでは、シリアルにレンダリング処理終了後に動画圧縮処理を実行するものとする。

これに対して、配分方法2では、レンダリング処理と動画圧縮処理が別々の PC で行われる。したがって、ある PC 上でレンダリング処理された画像は、動画圧縮処理を行う他の PC に移動することが前提となる。そのため、それぞれの PC には、画像を1フレームずつ与える。また、前述したように、動画圧縮処理が隣り合うフレーム間の相関を利用することを考えると、動画圧縮処理を行うPCには、フレーム番号順に画像が送られてくるのが望ましい。そのため、レンダリング処理を行っている PC には、処理を早く終えたものから順にフレーム番号の若い画像を与えるものとする。

以下では、2.5.1 において、レンダリング処理に負荷変動がない場合の検討を行い、その結果を踏まえて 2.5.2 において、レンダリング処理に負荷変動がある場合の検討を行う。

2.5.1 レンダリング処理に負荷変動がない場合

一般に、フレーム毎のレンダリング処理には負荷変動がある。しかし、ここでは、そうした負荷変動がないものとして、どちらの配分方法が処理時間の面から有利であるかを考える。

プロセス間通信にかかる時間を無視すると、各 PC の処理終了時間は次式で概算できる。

$$T_{node} = (T_i \times F_i) + (T_m \times F_m) \quad \dots(2)$$

T_i : 処理終了時間(sec)

T_i : 単位レンダリング処理時間(sec/frame)

F_i : レンダリング処理を行うフレーム数(frame)

T_m : 単位動画圧縮処理時間(sec/frame)

F_m : 動画圧縮処理を行うフレーム数(frame)

全ての PC の処理能力が等しいとすると、配分方法1では、レンダリング処理および動画圧縮処理を行う各PCでの担当フレーム数を等分すれば、すべて同時終了して処理終了時間は最短となる。これに対して、配分方法2では、レンダリング処理を行う PC と動画圧縮処理を行う PC の台数配分によって、レンダリング処理、動画圧縮処理を行うそれぞれの PC の担当フレーム数が変わり、処理終了時間も変化する。動画圧縮処理は、レンダリング処理の出力結果を入力とする処理である。したがって、実際には、レンダリング処理終了前に動画圧縮処理が終了することはない。しかしここでは、レンダリング処理と動画圧縮処理を単純に比較し、大きいほうを終了時間と考えることにする。

単位レンダリング処理時間を50sec/frame、単位動画圧縮処理時間を10sec/frameとし、20台の PC を用いて2000枚の画像を処理させた場合、配分方法2におけるレンダリング処理と動画圧縮処理を担当する PC の台数の変化により、処理終了時間が表 3のように変化する。

処理終了時間が最も早くなるのは、レンダリング処理を16台の PC に行わせ、動画圧縮処理を4台の PC に行わせたときであり、処理終了時間は6250secである。

表 3 配分方法2における処理終了時間の変化の様子

単位レンダリング 処理時間 (sec/frame)	単位動画作成 処理時間 (sec/frame)	レンダリング 処理台数(台)	動画圧縮 処理台数(台)	配分方法2の処理時間		
				レンダリング 処理時間(sec)	動画圧縮 処理時間(sec)	終了 時間(sec)
50	10	19	1	5263	20000	20000
50	10	18	2	5556	10000	10000
50	10	17	3	5882	6667	6667
50	10	16	4	6250	5000	6250
50	10	15	5	6667	4000	6667
50	10	14	6	7143	3333	7143
50	10	13	7	7692	2857	7692
50	10	12	8	8333	2500	8333
50	10	11	9	9091	2222	9091
50	10	10	10	10000	2000	10000
50	10	9	11	11111	1818	11111
50	10	8	12	12500	1667	12500
50	10	7	13	14286	1538	14286
50	10	6	14	16667	1429	16667
50	10	5	15	20000	1333	20000
50	10	4	16	25000	1250	25000
50	10	3	17	33333	1176	33333
50	10	2	18	50000	1111	50000
50	10	1	19	100000	1053	100000

次に、使用する PC の台数を等しくして、配分方法1と2の終了時間を比較する。表 4は、単位動画圧縮時間が10sec/frameで、単位レンダリング処理時間が10～160sec/frameであったときの配分方法1と2の終了時間を示した表である。使用する PCは20台で、2000フレームの画像を処理させる。表 3に示したように、配分方法2では、レンダリング処理を行う PC 台数と動画圧縮処理を行う PC 台数の組み合わせによって、終了時間が変化するが、その中で最も早いものを終了時間とした。両者を比較すると分るように、ほとんどの場合で、配分方法1が早く終了する。

表 4 配分方法 1 と 2 の終了時間

単位レンダリング処理時間(sec/f)	単位動画処理時間(sec/f)	配分方法1			配分方法2				
		レンダリング時間(sec)	動画圧縮時間(sec)	終了時間(sec)	レンダリング処理台数(台)	動画圧縮処理台数(台)	レンダリング時間(sec/f)	動画圧縮時間(sec/f)	終了時間(sec)
10	10	1000	1000	2000	10	10	2000	2000	2000
15	10	1500	1000	2500	12	8	2500	2500	2500
20	10	2000	1000	3000	13	7	3077	2857	3077
25	10	2500	1000	3500	14	6	3571	3333	3571
30	10	3000	1000	4000	15	5	4000	4000	4000
35	10	3500	1000	4500	15	5	4667	4000	4667
40	10	4000	1000	5000	16	4	5000	5000	5000
45	10	4500	1000	5500	16	4	5625	5000	5625
50	10	5000	1000	6000	16	4	6250	5000	6250
55	10	5500	1000	6500	17	3	6471	6667	6667
60	10	6000	1000	7000	17	3	7059	6667	7059
65	10	6500	1000	7500	17	3	7647	6667	7647
70	10	7000	1000	8000	17	3	8235	6667	8235
75	10	7500	1000	8500	17	3	8824	6667	8824
80	10	8000	1000	9000	17	3	9412	6667	9412
85	10	8500	1000	9500	18	2	9444	10000	10000
90	10	9000	1000	10000	18	2	10000	10000	10000
95	10	9500	1000	10500	18	2	10556	10000	10556
100	10	10000	1000	11000	18	2	11111	10000	11111
105	10	10500	1000	11500	18	2	11667	10000	11667
110	10	11000	1000	12000	18	2	12222	10000	12222
115	10	11500	1000	12500	18	2	12778	10000	12778
120	10	12000	1000	13000	18	2	13333	10000	13333
125	10	12500	1000	13500	18	2	13889	10000	13889
130	10	13000	1000	14000	18	2	14444	10000	14444
135	10	13500	1000	14500	18	2	15000	10000	15000
140	10	14000	1000	15000	18	2	15556	10000	15556
145	10	14500	1000	15500	18	2	16111	10000	16111
150	10	15000	1000	16000	18	2	16667	10000	16667
155	10	15500	1000	16500	18	2	17222	10000	17222
160	10	16000	1000	17000	18	2	17778	10000	17778

次に、使用する PC の台数と終了時間差の関係を示す。図 8、図 9は、単位動画圧縮時間をそれぞれ10sec/frame、20sec/frameとし、単位レンダリング処理時間を10sec/frame～400sec/frameの間で変化させたときの、配分方法1と2の終了時間差を示した図である。2000フレームの画像を処理させ、PC 台数は、20台、60台、100台の場合について示してある。なお、縦軸の終了時間差は、配分方法2の終了時間

から配分方法1の終了時間を引いた値である。したがって、グラフ上で正の値になるということは、配分方法1が配分方法2よりも早く終わることを意味している。

図 8、図 9から、終了時間差は、PC 台数が少ない場合に大きくなっていることが分る。これは、配分方法2では、レンダリング処理と動画圧縮処理とのロードバランスをとる必要があるが、それは、PC 台数が少ないほど困難であり、レンダリング処理、動画圧縮処理の一方がもう一方を待たせ、効率を落としているものと考えられる。さらに、PC 台数が少ないほど、1台の PC で処理しなければならないフレーム数が増える。そのため、1フレームの処理時間差がより大きく終了時間の差に現れるということも考えられる。

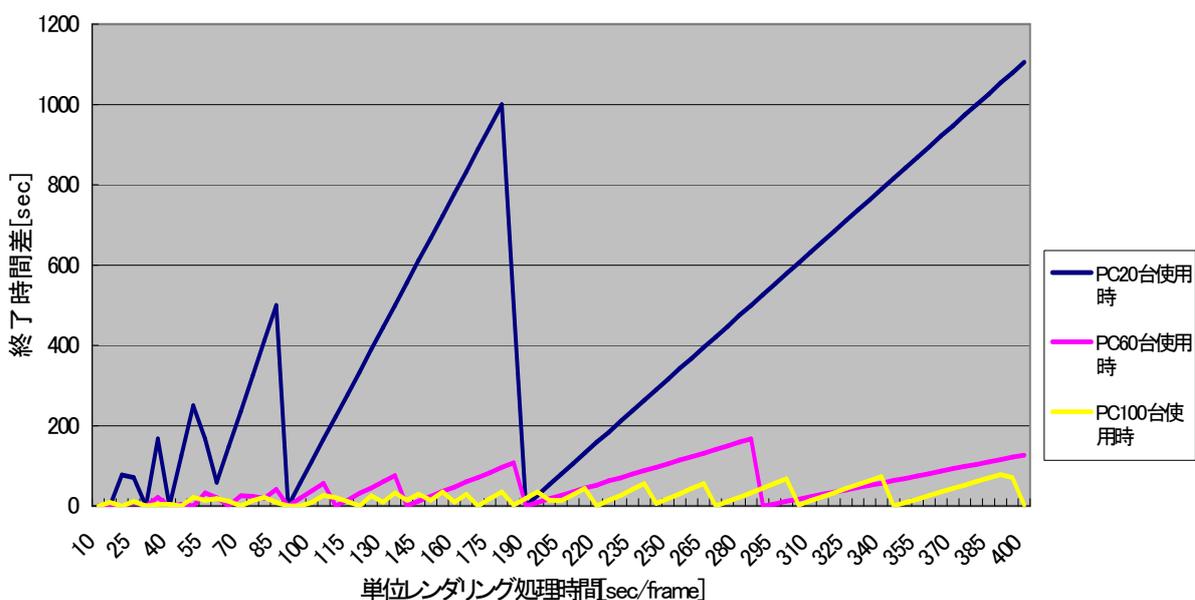


図 8 配分方法 1 と 2 の終了時間差 (単位動画圧縮処理時間 10sec/frame)

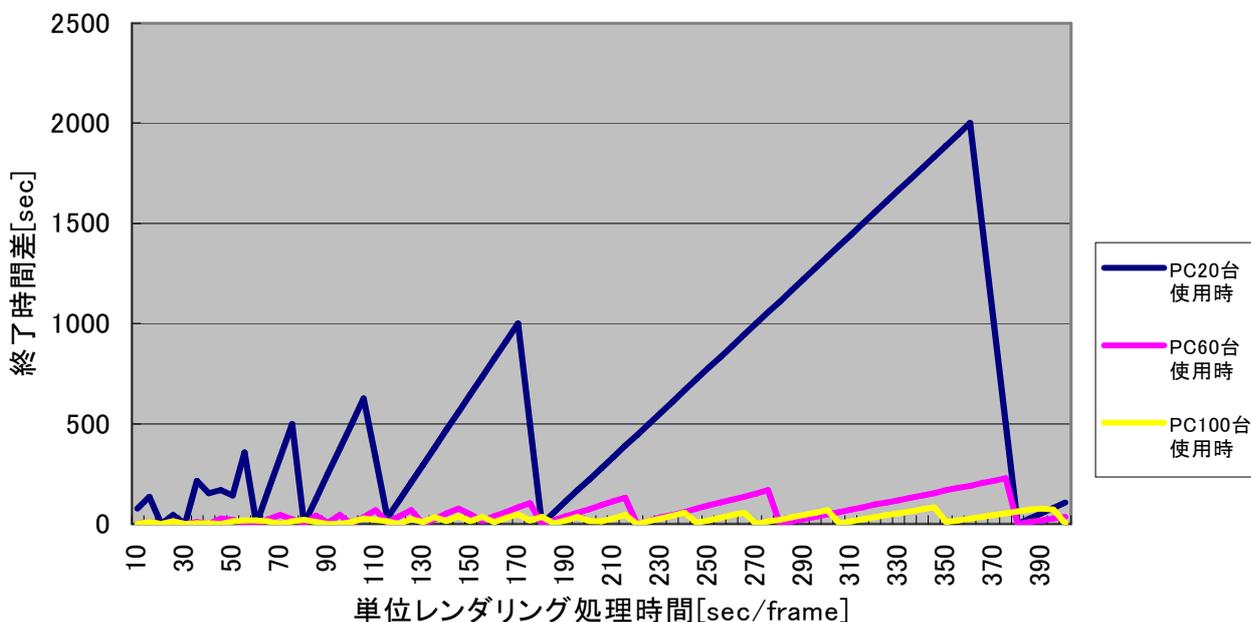


図 9 配分方法 1 と 2 の終了時間差 (単位動画圧縮処理時間 20sec/frame)

本システムの場合を考えると、制御 PC とレンダリング PC 合わせて21台の PC がある。ただし、制御 PC は、レンダリング PC やテクニライブラリの処理などを行う必要があるため、レンダリング PC と同じように取り扱うことはできないが、比較的処理の負荷が軽い動画圧縮処理であれば実行可能であると考えられる。そこで、次に、配分方法1で使用した台数よりも1台多い PC を使用して、配分方法2を実行した場合を考えてみる。

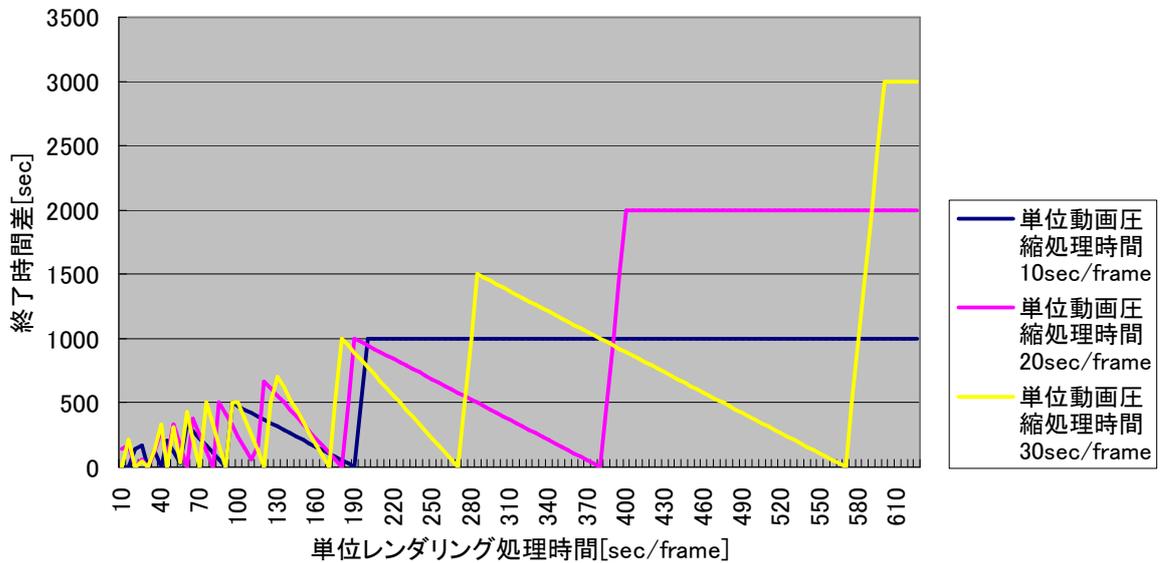


図 10 配分方法 1 と 2 の終了時間差 (配分方法 1: 計算機 20 台、配分方法 2: 計算機 21 台)

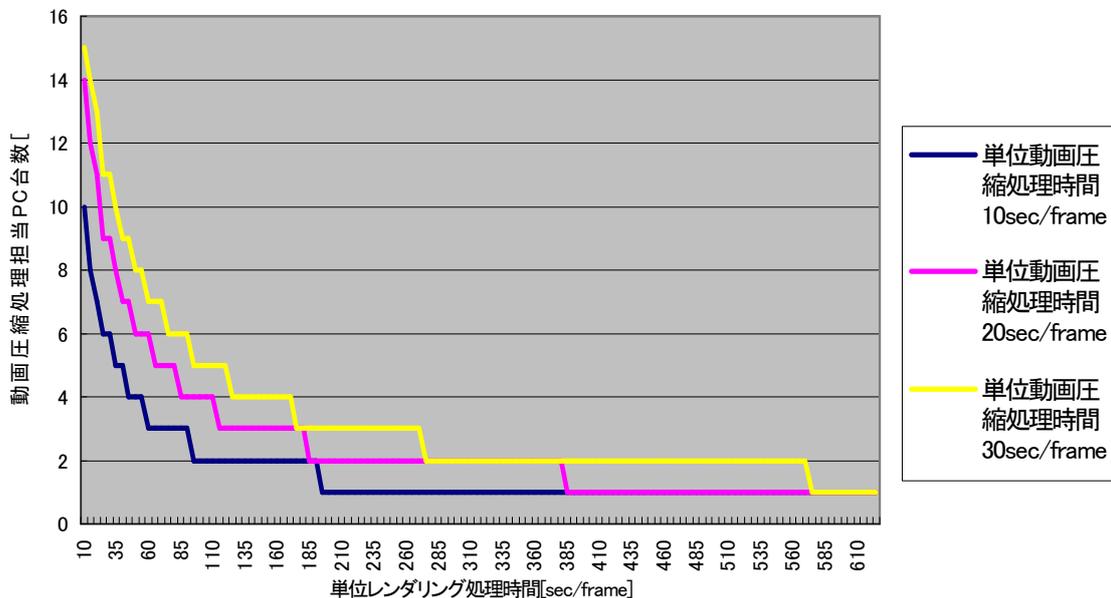


図 11 動画圧縮処理担当 PC の台数推移 (配分方法 2)

図 10は、20台の PC を使用して配分方法1を実行したときと、21台の PC を使用して配分方法2を実行したときの終了時間差を示した図である。ただし、ここでの終了時間差は、図 8、図 9とは逆で、配分方法1の終了時間から配分方法2の終了時間

を引いた値となっている。したがって、グラフ上で正の値になるということは、配分方法2が配分方法1よりも早く終わることを意味している。また、図 11は、動画圧縮処理を担当する PC の台数推移を示した図である。

図 10から、PC 台数が1台増えたことにより、ほとんどの場合において、配分方法2が早く終了している。ただし、これは、あくまでも、レンダリング処理時間および動画圧縮処理時間を用いて、それぞれの処理を担当する PC の台数を最適なものに調節した場合の結果である。また、単位動画圧縮処理時間が10sec/frameの時には、200sec/frame以上、単位動画圧縮処理時間が20sec/frameの時には、400sec/frame以上、単位動画圧縮処理時間が30sec/frameの時には、600sec/frame以上かかるレンダリング処理に対して、終了時間差が一定値となっていることも分かる。これは図 11から分るように、動画圧縮処理が1台の PC で行われるようになったためである。このとき、配分方法1と2でレンダリング処理を行う PC 台数は共に20台で等しい。配分方法2では、レンダリング処理と動画圧縮処理を同時に行うため、動画圧縮処理時間の分だけ配分方法2が早く終わる。

また、1台の PC で動画圧縮を行うようになる単位レンダリング時間は、単位動画圧縮処理時間のレンダリング処理担当 PC 台数倍になっている。これは、1台の PC で20フレーム分の動画圧縮を行う時間と、20台の PC が1フレーム分のレンダリング処理を行う時間とが等しくなるということである。

以上の結果から、処理能力が等しい同数の PC を使った場合、処理にかかる時間の面から、配分方法2よりも配分方法1の方が有利である。しかし、配分方法2において、制御 PC を含めた21台の PC を使用できると考えた場合、20台の PC を使用して配分方法1を実行するよりも配分方法2を実行する方が有利である。

2.5.2 レンダリング処理に負荷変動がある場合

本システムの構成を考えた場合、レンダリング処理に負荷変動がないとすると、制御用PC を入れた21台の PC で配分方法2を実行するのが処理時間の面から有利であることが分った。ここでは、レンダリング処理に負荷変動がある場合について考えてみる。

図 11に示すように、配分方法2では、単位レンダリング処理時間と単位動画圧縮処理時間の組み合わせによって、レンダリング処理と動画圧縮処理の最適な台数配分が決まる。

そして、この最適な台数配分で処理している場合には、配分方法2の方が1よりも終了時間が早い。しかし、最適な台数配分で処理しなかった場合、配分方法1より多くの PC を使用しているにもかかわらず、その終了時間は遅くなってしまふ。

また、単位動画圧縮処理時間が10sec/frameのとき、動画圧縮処理を5台で行うのが最適となる単位レンダリング処理時間の範囲が35～40sec、4台で行う範囲が45～55sec、3台で行う範囲が60～90sec、2台で行う範囲が95～190sec、1台で行う範囲が195sec以上というように、動画圧縮処理を行う PC 台数が多ければ多いほど、その台数配分が最適なものとなる単位レンダリング処理時間の範囲は狭くなる。

しかし、動画圧縮処理を行う PC 台数を動的に変えることは困難であり、台数配分を頻繁に変えることは現実的ではない。このことから、レンダリング処理の負荷変動を予測し、もっとも有効な台数配分を事前に決めることが必要になる。

したがって、レンダリング処理時間をどの程度予測できるかによって、制御方法は変わってくる。例えば、レンダリング処理時間がある程度予測できるのであれば、その予測時間を用いて、配分方法1および配分方法2の複数の台数配分における終了時間を求め、最も短時間に処理を完了するものを採用することが可能である。しかし、単位レンダリング処理時間がほとんど予測できない場合には、配分方法1および配分方法2を切り替えて使用することは困難である。

本稿では、後述する 4章 において、レンダリング処理時間、動画圧縮処理時間の予測方法について検討を行い、与えられた画像処理フローを最も効率的に処理する配分方法を選択させることを考える。

2.6 クラス構成

以上の検討を踏まえ、本システムでは、配分方法1と2を下図のようなクラス構成により切り替えることとする。

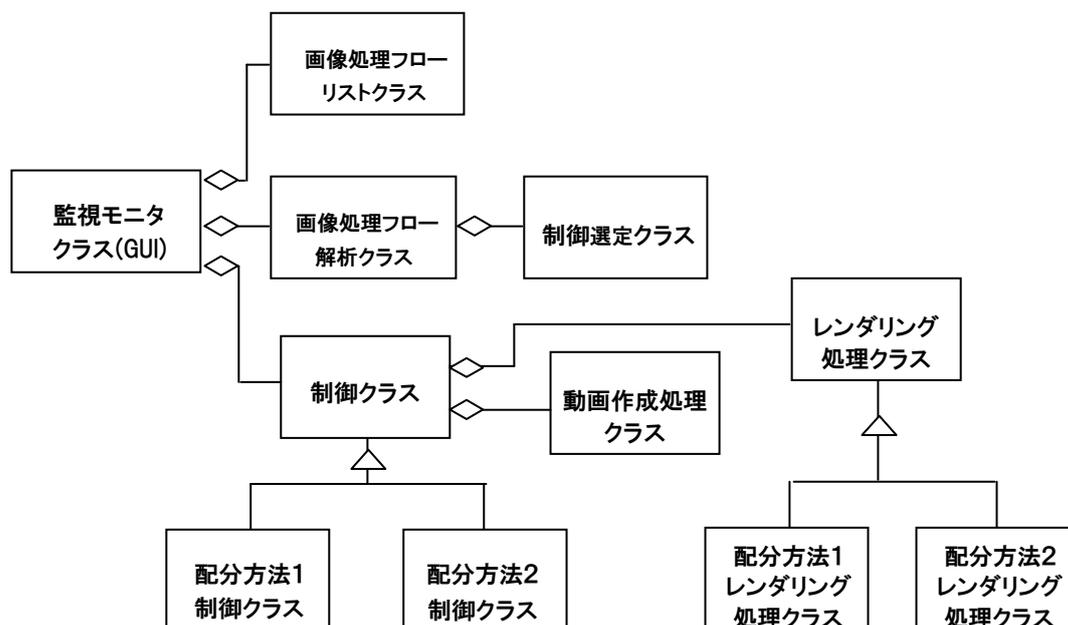


図 12 クラス構成

3. 映像編集・制作に必要な画像処理ライブラリの開発

ここでは、図 12に示すレンダリング処理クラスが、実際に画像処理を行うために開発した画像処理ライブラリについて述べる。

3.1 画像処理ライブラリの機能仕様

我々がこれまで、IPD 用のコンテンツを制作するために使用してきたノンリニア編

集システムを参考に、画像処理ライブラリの機能仕様を抽出した。以下に抽出した機能仕様を示す。

- ① 画像処理は、入力ポートと出力ポートを持つものとする。入力ポートとは、その画像処理で処理される画像の入口であり、出力ポートとは、その画像処理で処理された画像の出口である。入力ポート、出力ポートには、0 から始まるポート番号を付け、この番号で個々のポートを特定する。図 13は、入力ポートと出力ポートをそれぞれ2つ持つ画像処理の例である。複数の入力ポートや複数の出力ポートを持つ画像処理の例としては、2つの画像を合成する画像処理とか、画像を RGB 各色の色成分に分ける画像処理などが考えられる、以後、画像処理を図 13のような形で示す。ただし、画像処理のポート番号は省略可能とし、省略した場合、上から0、1…と考える。

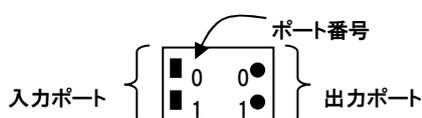


図 13 画像処理の入力ポートと出力ポート

- ② ある画像処理の入力ポートと出力ポートを接続することにより、画像処理フローを構築する。例えば図 14に示すように、A と B の2つの画像処理があった場合、画像処理 A の出力ポートと画像処理 B の入力ポートを接続することにより、画像処理 A の実行後に、その出力画像が画像処理 B で処理されるようになる。

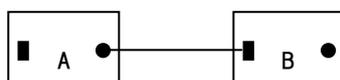


図 14 画像処理の接続

- ③ 画像処理フロー中で用いられる画像処理は、大きく次の3つに分類する。1つは、画像ファイルをハードディスクからメモリに読み込む画像処理（画像読み込み処理）、もう1つは、画像ファイルをメモリからハードディスクに書き込む画像処理（画像書き込み処理）、最後は、メモリ上で画像データを加工する画像処理（画像編集処理）である。なお、動画（圧縮）ファイルを作成する処理も、画像書き込み処理とみなすものとする。
- ④ 有効な画像処理フローは、画像読み込み処理から始まり、画像書き込み処理で終わるものとする。
- ⑤ 入力ポートには、一つの出力ポートしか接続できないものとするが、出力ポートには、複数の入力ポートが接続できるものとする(図 15)。

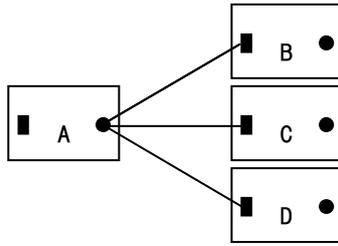


図 15 画像処理の接続例

- ⑥ 各画像処理のポート数として、画像読み込み処理は、入力ポートを持たず、出力ポートを1つだけ持つ。画像書き込み処理は、出力ポートを持たず、入力ポートを1つだけ持つ。画像編集処理には、入力ポート数、出力ポート数に対する制限は設けず、画像処理設計者が必要なポート数を定義できるものとする。
- ⑦ 画像編集処理が持つ入力ポートおよび出力ポートの意味は、画像処理設計者が決めるものとする。例えば、2つの画像を合成する画像処理であれば、2つの入力ポートを設け、入力ポート 0 に背景画像、入力ポート 1 に前景画像を入力するように設計することができる。逆に、入力ポート 0 に前景画像、入力ポート 1 に背景画像を入力するように設計することも可能である。
- ⑧ 画像処理には、任意の数のパラメータを定義できるものとする。パラメータに使用できるデータ型式として、整数型、倍精度型、色型、テキスト型の4つを用意する。
- ⑨ パラメータの値は、オペレータにより設定されるが、設定されていないフレームのパラメータ値は補間により算出する。補間には、図 16に示すような Constant , Step , Linear , Spline の4つを設ける。

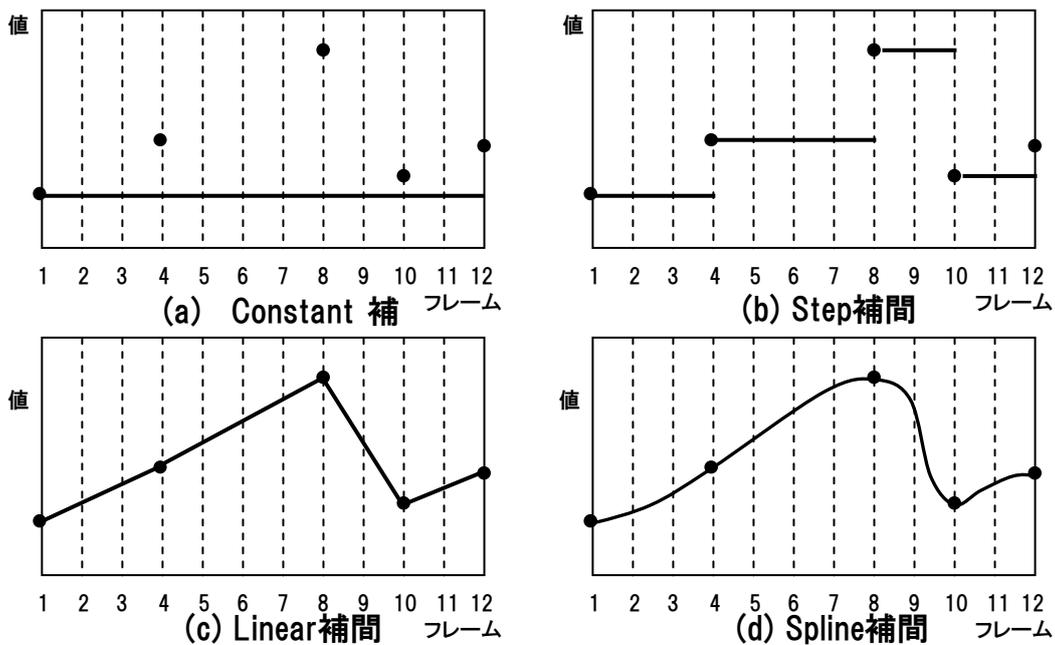


図 16 パラメータの補間モード

- ⑩ 画像処理は容易に追加できるようにする。追加された画像処理は、プログラムを

再コンパイルせずに、使用できるようにする。

3.2 画像処理フロー設定ファイルの記述方法

画像処理フロー設定ファイルは Extensible Markup Language (XML) 形式で記述する。その主な理由を以下に示す。

- ① Microsoft 社からパーサーが供給されており、プログラム中での取扱が比較的簡単である。
- ② 任意のタグ名が定義でき、そのタグで挟まれた文字や数字に意味を持たせられる。このため、各設定項目の意味が分り易くなる。
- ③ 設定項目をタグ名で識別するので、設定項目の追加によるプログラムの修正が不要である。

画像処理フロー設定ファイルには、入力映像データから出力映像データを作成するために、どの画像処理をどの順番で使用し、個々の画像処理が持つパラメータの値がフレームごとにどのように変化するかが定義される。

図 17に図 2で示した画像処理フロー設定ファイルの記述例を示す。この例を用いて画像処理フロー設定ファイルの記述方法を説明する。画像処理フローの内容は、ImageProcessFlow タグ(図 17-(1))を用いて定義する。

以下、ImageProcessFlow タグの構成を説明する。画像処理フロー内で使用される個々の画像処理は、ImageProcess タグ(図 17-(2))で定義する。ImageProcess タグは、ImageProcessFlow タグの中でだけ使用することのできるタグである。また、個々の画像処理が持つパラメータの値は、Parameter タグ(図 17-(3))を用いて定義し、画像処理を実行する順番は、Connect タグ(図 17-(4))を用いて定義する。Parameter タグ、Connect タグは、ImageProcess タグの中でだけ使用できるタグである。

ImageProcess タグには2つの属性がある。1つは、ID 属性であり、もう1つは Name属性である。ID 属性は、画像処理フロー中で、個々の画像処理を識別するために使用する番号である。そのため、ImageProcessFlow タグの中で同じID属性値を持つ ImageProcess タグが複数存在しないようにする必要がある。Name 属性は、画像処理の名称を設定する属性で、この設定により、使用する画像処理が決められる。

Parameter タグには、No , Interpolation という2つの属性がある。No 属性は、その画像処理が持つパラメータの識別番号であり、Interpolation 属性は、そのパラメータの補間モードを指定する属性である。Interpolation 属性に設定する属性値は、Constant , Step , Linear, Splineのいずれかである。また、この属性の記述を省略した場合には、Constant 補間モードが選択される。各フレームのパラメータ値は、Data タグを用いて設定する。Data タグは、Parameter タグの中でだけ使用できるタグである。Data タグの FrameNo 属性によってフレーム番号を、Data タグで値を設定する。

Connect タグは、InPortNo , OutNodeID , OutPortNo という3つの属性を持つ。この

3つの属性を用いて、自分のどの入力ポートに、どの画像処理のどの出力ポートが接続されるかを設定する。InPortNo 属性には、入力ポート番号を設定する。また、OutNodeID 属性と OutPortNo 属性に、その入力ポートに接続される画像処理の ID 番号とポート番号を設定する。

```

<ImageProcessFlow>
  <ImageProcess ID = "1" Name = "BMP Read" >
    <Parameter No = "0" >
      <Data FrameNo = "1" >g:¥patyo¥</Data>
    </Parameter>
    <Parameter No = "1" >
      <Data FrameNo = "1" >froc_</Data>
    </Parameter>
    <Parameter No = "2" >
      <Data FrameNo = "1" >bmp</Data>
    </Parameter>
  </ImageProcess>
  <ImageProcess ID = "2" Name = "ColorAdjust" >
    <Parameter No = "0" Interpolation = "Step" >
      <Data FrameNo = "1" >0.2</Data>
      <Data FrameNo = "10" >0.5</Data>
    </Parameter>
    <Parameter No = "2" >
      <Data FrameNo = "1" >-0.1</Data>
    </Parameter>
    <Connect InPortNo = "0" OutNodeID = "1" OutPortNo = "0" ></Connect>
  </ImageProcess>
  <ImageProcess ID = "3" Name = "Zoom" >
    <Parameter No = "0" >
      <Data FrameNo = "1" >1.25</Data>
    </Parameter>
    <Parameter No = "1" >
      <Data FrameNo = "1" >1.25</Data>
    </Parameter>
    <Connect InPortNo = "0" OutNodeID = "2" OutPortNo = "0" ></Connect>
  </ImageProcess>
  <ImageProcess ID = "4" Name = "Rotate" >
    <Parameter No = "0" Interpolation = "Spline" >
      <Data FrameNo = "1" >0.0</Data>
      <Data FrameNo = "20" >30.0</Data>
      <Data FrameNo = "40" >60.0</Data>
    </Parameter>
    <Connect InPortNo = "0" OutNodeID = "3" OutPortNo = "0" ></Connect>
  </ImageProcess>
  <ImageProcess ID = "5" Name = "BMP Write" >
    <Parameter No = "0" >
      <Data FrameNo = "1" >g:¥Output¥1¥nodegraph¥Output Image¥0¥</Data>
    </Parameter>
    <Parameter No = "1" >
      <Data FrameNo = "1" >Output_</Data>
    </Parameter>
    <Connect InPortNo = "0" OutNodeID = "4" OutPortNo = "0" ></Connect>
  </ImageProcess>
</ImageProcessFlow>

```

図 17 画像処理フロー設定ファイルの例

3.3 画像処理ライブラリの概要

画像処理ライブラリは、主に CImageProcessController クラス、CImage クラス、画像処理の実体が定義された DLL の3つから構成される。

CImageProcessController クラスは、画像処理フロー設定ファイルを読み込み、そこに記述された画像処理フローを構築や、その画像処理フローを実行するなどの制御を行うクラスである。このクラスが図 12に示すレンダリング処理クラスの実体から制御され、レンダリングが進行することになる。

CImage クラスは、画像処理フロー中の個々の画像処理に対応するクラスであり、そこには、各画像処理に共通するインターフェースや機能が定義される。CImage クラスの主なメンバー関数を表 5に示す。また機能としては、パラメータ管理、出力画像データ管理、入力ポートの設定情報管理などがある。

表 5 CImage クラスの主なメンバー関数

No	メンバー関数名	概要
1	Initial	画像処理の初期化を行う。主に入力ポート、出力画像データ領域の作成とパラメータ管理領域の作成を行う。
2	Process	画像処理を実行する。

1つの画像処理に対して1つの DLL が作成され、そこには、表 6に示すメンバー関数を持ったクラスの派生クラスが定義される。各メンバー関数の中で、画像処理の実際のアルゴリズムや、入出力ポート数、パラメータの数やその内容などを定義する。

表 6 DLL で定義するクラスの主なメンバー関数

No	関数名	概要
1	WhatIsYourName	画像処理名称を返す。
2	WhatIsYourKind	画像処理の種別(画像読み込み処理、画像書き込み処理、画像編集処理)を返す。
3	GetInputTotal	画像処理の入力ポート数を返す。
4	GetOutputTotal	画像処理の出力ポート数を返す。
5	GetParamTotal	画像処理が持つパラメータの数を返す。
6	GetParamInfo	パラメータの値を返す。パラメータの定義も行う。
7	SetParam	パラメータに値を設定する。
8	Process	画像処理を実行する。

以下、画像処理フローの構築および画像処理フローの実行の動作について、その簡単な流れを示す。

3.3.1 画像処理フローの構築

画像処理フローの構築は、次の2つのステップからなる。

- ① 画像処理フロー設定ファイルの ImageProcess タグで示された全ての画像処理を生成し、初期化する。
- ② 画像処理フロー設定ファイルの Connect タグの指示に従い、①で生成した画像処理のポートを接続する。

①のステップでは、CImageProcessController オブジェクトが、画像処理フロー設定ファイル中の ImageProcess タグの Name 属性に設定された画像処理名称を用いて、その名称に対応する DLL のディレクトリパスをレジストリから検索する。検索されたディレクトリパスは、CImage オブジェクトの生成時に渡され、その CImage オブジェクト内に読み込まれる。これにより、このオブジェクトにおいて DLL で定義されたメンバー関数が利用可能になる(図 18)。

DLL で定義されたメンバー関数は、CImage オブジェクトのメンバー関数を通じて利用できるようになっている。例えば、BMP 画像読み込み処理用 DLL を読み込んだ CImage オブジェクトがあった場合、そのオブジェクトの Process 関数を実行すると、BMP 画像読み込み処理用 DLL の Process 関数が実行される。そこには、指定された BMP 画像ファイルをメモリに読み込むアルゴリズムが定義されており、そのアルゴリズムに従って画像ファイルの読み込みが行われる。CImage オブジェクトのメンバー関数とその関数内で実行される DLL で定義されるクラスのメンバー関数との対応を図 19に示す。

次に、生成された CImage オブジェクトのInitialメンバー関数を実行することにより、その初期化が行われる。このとき、入力ポートの設定を保存するためのテーブルが作成されたり、出力ポート数分の出力画像データ領域などが作成されたりする。

CImageProcessController オブジェクトには、画像読み込み処理用、画像書き込み処理用、画像編集処理用それぞれのリストがあり、生成された CImage オブジェクトは、その種類に応じたリストに入れられ、管理される。

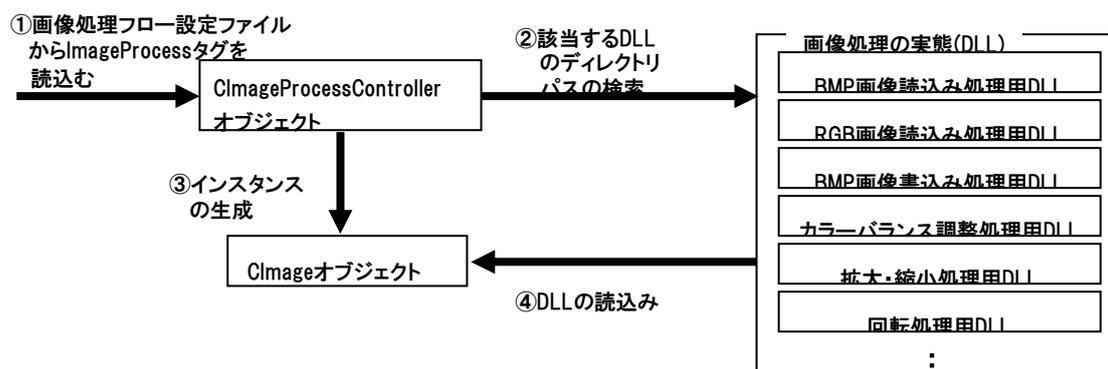


図 18 画像処理フロー設定ファイル読み込み時の動作概要

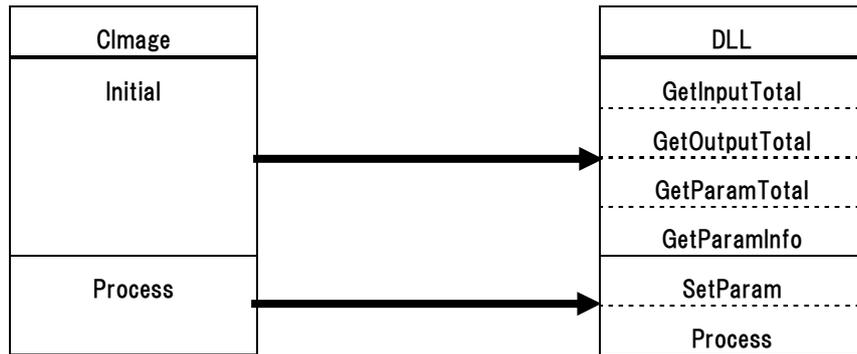


図 19 CImage のメンバー関数と DLL のメンバー関数との関係

②のステップでは、CImageProcessController オブジェクトが、画像処理フロー設定ファイル中の Connect タグのInPortNo , OutNodeID , OutPortNo 属性を用いて、ポートの接続を行う。CImage オブジェクトには、その初期化時に、入力ポートの設定を行うためのテーブルが作成されている。このテーブルには、接続先画像処理 (CImage オブジェクト) のアドレスとその出力ポート番号が設定される。接続先画像処理のアドレスは、OutNodeID 属性に設定された番号を元に、画像読み取りリスト、画像書き込みリスト、画像編集用リストの要素を検索することにより取得される。

3.3.2 画像処理フローの実行

画像処理フローの実行は、画像処理フロー中で、画像読み取り処理から、ある1つの画像書き込み処理に至る経路を求め、この経路上に存在する全ての画像処理の Process メンバー関数を順番に実行することにより行われる。Process メンバー関数では、入力ポートの設定情報を用いて接続先の画像処理オブジェクトを求め、そこから該当する出力画像データを取り出す。そして、この画像データに対して処理を行い、その処理結果で自分自身の出力画像データを作成する。前述したように、このとき行われる処理内容は、その CImage オブジェクトの生成時に読んだ DLL の Process 関数で定義されている。ただし、入力ポートを持たない画像読み取り処理に関しては、ハードディスク上にある画像ファイルから画像データを読み込むことになる。

以下に簡単な処理の流れを示す。図 20に示すような画像処理フロー設定ファイルで記述された画像処理フローを実行する場合を考える。ただし、図 20は、説明用の画像処理フロー設定ファイルであるため、ImageProcess タグと Connect タグしか記述していない。この画像処理フロー設定ファイルを CImageProcessController オブジェクトが読んだ場合、CImageProcessController オブジェクトの画像読み取り処理用リスト、画像書き込み処理用リスト、画像編集処理用リストの状態は、図 21のようになる。このとき、A～E に対応する CImage オブジェクトの入力ポートの設定には、図 22の矢印が指し示す CImage オブジェクトのアドレスと出力ポートが設定されている。例えば、画像処理 E の入力ポート設定には、接続先アドレスに画像処理 D が設定され、出力ポート番号に0が設定されている。また、画像処理 D の入力ポート設定は、ポート番号0の設定として、接続先アドレスが画像処理 C 、出力ポート番号に0が設

定される。また、ポート番号1の設定として、接続先アドレスが画像処理 B、出力ポート番号に0が設定される。

CImageProcessController オブジェクトは、画像書き込み処理用リストから、その要素である CImage オブジェクトを取り出し、その入力ポート設定を用いて、画像処理フローを画像書き込み処理から画像読み込み処理の方向にたどって行く。そのとき、図 23-(a)に示すような探索経路をたどる。この処理によって見つけられた CImage オブジェクトは、順番に、画像処理実行用リストの先頭に追加される(図 23-(b))。

次に、画像処理実行用リストの要素を先頭から順番に取り出し、Process メンバ関数を実行する。

画像処理フロー中に複数の画像書き込み処理があった場合、上記動作を画像書き込み処理の分だけ繰り返すことになる。ただし、2回目以降は、既に処理を終えている画像処理の Process 関数を実行しないなど、処理の効率化を図っている。

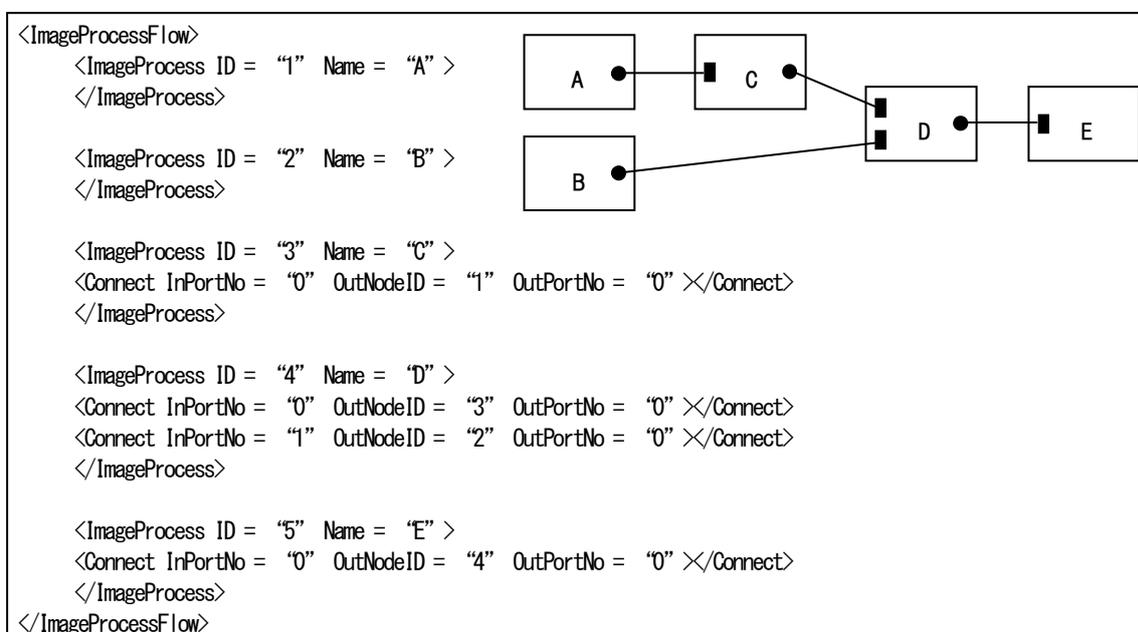


図 20 ジョブファイル例



図 21 CImageProcessController での CImage オブジェクト管理状態

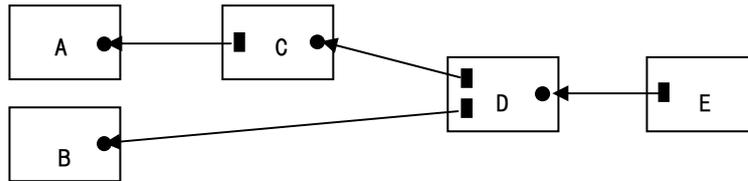
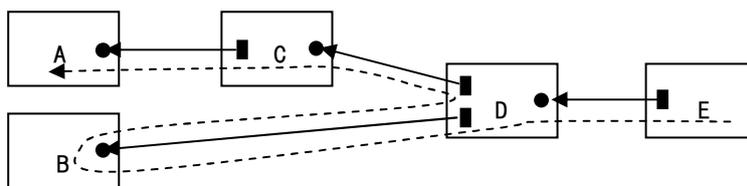
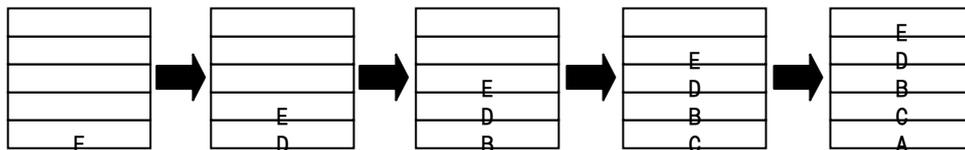


図 22 各 CImage オブジェクトの入力ポート設定



(a) 経路探索手順



(b) 画像処理実行リストの更新状態

図 23 画像処理フローの経路探索手順と画像処理実行リストの更新状態

3.4 使用可能な画像処理

ここでは、現在使用可能な画像処理を列挙する。

3.4.1 画像入力

[Bmp Reader]

BMP形式の画像ファイルをハードディスクからメモリに読み込む。出力ポートを1つ持つ。

Attribute No	名称	型	概要
0	Directory Path	文字型	読み込む画像のディレクトリパス
1	Prefix	文字型	読み込む画像のプレフィックス名

[Rgb Reader]

RGB形式の画像ファイルをハードディスクからメモリに読み込む。出力ポートを1つ持つ。

Attribute No	名称	型	概要
0	Directory Path	文字型	読み込む画像のディレクトリパス
1	Prefix	文字型	読み込む画像のプレフィックス名

3.4.2 画像出力

[Bmp Writer]

BMP形式の画像ファイルをメモリからハードディスクに書き出す。入力ポートを1つ持つ。

Attribute No	名称	型	概要
0	Directory Path	文字型	出力画像のディレクトリパス
1	Prefix	文字型	出力画像のプレフィックス名
2	Return On/Off	整数型	出力画像を制御用PC上に作成するかどうかを示すフラグ。0または1の値を取り、1で制御用PC上に出力画像を作成する設定となる。

[AVI Writer]

AVI形式の動画ファイルを作成する。入力ポートを1つ持つ。

Attribute No	名称	型	概要
0	Directory Path	文字型	出力動画画像のディレクトリパス
1	Prefix	文字型	出力動画画像のプレフィックス名

3.4.3 Cut

画像の一部を切り取る。入力ポート、出力ポートをそれぞれ1つ持つ。

Attribute No	名称	型	初期値	概要
0	Origin X	整数型	0	出力画像の原点位置(x座標)(pixel)
1	Origin Y	整数型	0	出力画像の原点位置(y座標)(pixel)
2	Width	整数型	1	出力画像の水平方向解像度(pixel)
3	Height	整数型	1	出力画像の垂直方向解像度(pixel)

3.4.4 ColorAdjust

R・G・B各色のレベルを調節する。

Attribute No	名称	型	初期値	概要
0	R Level	倍精度型	0.0	$(r, g, b) \rightarrow (r + \Delta r, g, b)$ $\Delta r : \max \times \text{R Level設定値}(0.0 \sim 1.0)$
1	G Level	倍精度型	0.0	$(r, g, b) \rightarrow (r, g + \Delta g, b)$ $\Delta g : \max \times \text{G Level設定値}(0.0 \sim 1.0)$
2	B Level	倍精度型	0.0	$(r, g, b) \rightarrow (r, g, b + \Delta b)$ $\Delta b : \max \times \text{B Level設定値}(0.0 \sim 1.0)$

3.4.5 Zoom

画像の拡大または縮小を行う。ただし、解像度は変更されない。入力ポート、出力ポートをそれぞれ1つ持つ。

Attribute No	名称	型	初期値	概要
0	X Zoom	倍精度型	10	水平方向倍率
1	Y Zoom	倍精度型	10	垂直方向倍率
2	Source X	整数型	0	拡大・縮小の原点(x座標)
3	Source Y	整数型	0	拡大・縮小の原点(y座標)

3.4.6 Resize

解像度を変更する。それに伴い、画像が拡大または縮小される。入力ポート、出力ポートをそれぞれ1つ持つ。

Attribute No	名称	型	初期値	概要
0	Width	整数型	1	出力画像の水平方向解像度
1	Height	整数型	1	出力画像の垂直方向解像度

3.4.7 Fade

画像をフェードイン、フェードアウトする。2つの入力ポート、1つの出力ポートを持つ。

Attribute No	名称	型	初期値	概要
0	Fade Value	倍精度型	0	入力画像1のフェードアウト。入力画像2のフェードイン。 (値:0.0~1.0)

3.4.8 Curtain

画像を中央から左右に、別画像に変換する。2つの入力ポート、1つの出力ポートを持つ。

Attribute No	名称	型	初期値	概要
0	Curtain Value	倍精度型	0	入力画像1の非表示領域。入力画像2の表示領域。 (値:0.0~1.0)

3.4.9 Shutter

画像を中央から上下に、別画像に変換する。2つの入力ポート、1つの出力ポートを持つ。

Attribute No	名称	型	初期値	概要
0	Shutter Value	倍精度型	0	入力画像1の非表示領域。入力画像2の表示領域。 (値:0.0~1.0)

3.4.10 Character

画像に文字列を表示する。入力ポート、出力ポートをそれぞれ1つ持つ。

Attribute No	名称	型	初期値	概要
0	Chara_Fade	倍精度型	0	文字列のフェードイン (値:0.0~1.0)
1	Chara_Cut_Width	倍精度型	0	文字列の左側からの表示割合 (値:0.0~1.0)
2	Chara_Cut_Height	倍精度型	0	文字列の上下からの表示割合 下から表示(値:0.0~-1.0) 上から表示(値:0.0~1.0)
3	Chara_Width	整数型	0	文字列書き込み位置(pixel) (左:0)
4	Chara_Height	整数型	0	文字列書き込み位置(pixel) (上:0)
5	Chara	文字型	—	表示する文字列

6	Font_Height	整数型	100	フォント高さ(pixel)
7	Font_Width	整数型	0	フォント平均文字幅(pixel)(0:自動設定)
8	Font_Escapement	整数型	0	文字送り方向の回転(1/10度)
9	Font_Orientation	整数型	0	ベースラインとX軸の角度(1/10度)
10	Font_Weight	整数型	0	フォントの太さ(0~1000)(0:デフォルト)
11	Font_Italic	整数型	FALSE	斜体にする(TRUE)
12	Font_Underline	整数型	FALSE	下線をつける(TRUE)
13	Font_StrikeOut	整数型	FALSE	取り消し線をつける(TRUE)
14	Font_CharSet	整数型	SHIFTJIS_CHARSET	文字セット
15	Font_OutputPrecision	整数型	OUT_DEFAULT_PRECIS	出力精度
16	Font_ClipPrecision	整数型	CLIP_DEFAULT_PRECIS	クリッピング精度
17	Font_Quality	整数型	DEFAULT_QUALITY	出力品質
18	Font_PitchAndFamily	整数型	VARIABLE_PITCH FF_SCRIPT	ピッチとファミリー
19	Font_Face	文字型	—	フォント名
20	Character_Color	整数型	0xFFFFFFFF	文字色(RGB)

4. 処理の効率的なスケジュール管理手法の検討

2.5 において、コンテンツ制作に必要なレンダリング処理、動画圧縮処理をシステム内の PC に配分する方法についての検討を行った。ここでは、与えられた画像処理フローの実行に適した処理配分を自動的に選択する方法について検討を行う。また、処理効率および信頼性を考慮し、処理のスケジューリングについての検討も行う。

4.1 レンダリング処理時間の予測方法

レンダリング処理時間を予測しようとした場合、以下の2つの方法が考えられる。1つは、あらかじめシステムで使用可能な全ての画像処理の処理時間を計測しておき、それを足し合わせることによって画像処理フローの処理時間を算出する方法である。もう一つは、画像処理フローのレンダリングを実際に行い、その実測値を用いる方法である。

前者の方法は、画像処理フローを実際に行わずとも、レンダリング処理時間の算出ができる。このため、後者の方法に比べて、短時間に算出が可能であるという利点が考えられる。しかし、画像処理の処理時間は、対象画像の特徴や、画像処理が持つパラメータの値などに影響される場合がある。このため、全ての画像処理について、それらの影響を考慮した正確な処理時間を算出することは困難だと考えられる。

一方、後者の方法は、上記のような問題がなく、前者の方法に比べて、より正確な処理時間の算出が可能である。しかし、計測に時間がかかるため、あまり多くのフレームで処理時間を測定することはできない。

2章 で述べたように、オペレータが画像処理フローの編集を行う場合、画像処理フローの出力結果に変更が生じる操作を行うたびに、画像処理フローを実行する（簡易レンダリング）。このように、簡易レンダリングは、通常、オペレータが設定した画像処理パラメータの妥当性を確認するために行われるため、画像処理パラメータを明示的に調節したほとんどのフレームに対して、簡易レンダリングが行われることが期待される。したがって、簡易レンダリングは、レンダリング処理時間の変化を予測する有力な情報を与える可能性が高い。また、簡易レンダリングは、コンテンツを編集する上で必ず行われるものであり、オペレータの負担も少ないと思われる。そこで、画像処理フロー編集時に行った簡易レンダリングの処理時間を画像処理フロー設定ファイルに保存しておき、実際に処理時間を計測していないフレームに関しては、補間により処理時間を予測することにする。

3章 で述べたように、画像処理フロー中の画像処理のパラメータには、図 16に示すような補間モードが設けられる。したがって、処理時間の予測に使用される補間モードも画像処理フロー中の画像処理の補間モードに合わせる必要がある。そのため、画像処理設計者が画像処理作成時に処理時間に影響を与えるパラメータを指定できるような仕組みを設ける必要があると考えられる。

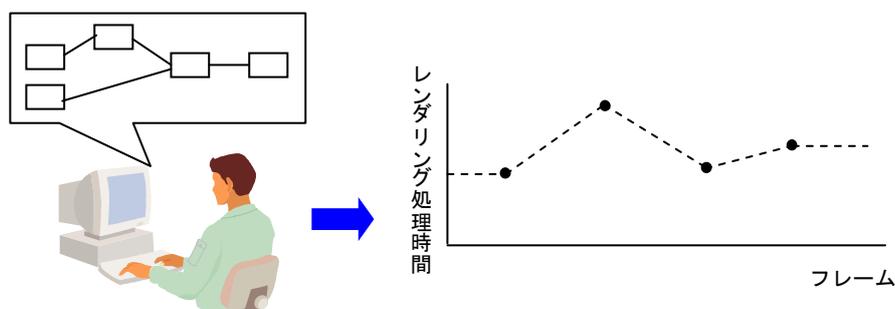


図 24 レンダリング処理時間の予測方法

一般に画像処理フローの編集は、GUI を用いて行われる。そのため、このようなレンダリング処理時間の取得は、その GUI で吸収し、オペレータに意識させることなく行われるのが理想的である。ただし、気をつけなければならない点は、オペレータが行った全ての簡易レンダリングが、有効なレンダリング処理時間を与えるわけではないということである。有効なレンダリング処理時間の条件とは、(a)本レンダリングで使用される画像処理およびその実行順序が決まり、オペレータの作業が各画像処理のパラメータ調節のみになっている段階にあり、しかも、(b)そのフレームで画像処理フローが出力するはずの全ての出力画像が実際に作成される場合のみである。すなわち、(a)の条件は、画像処理の追加や削除を繰り返している段階での簡易レンダリングは、有効なレンダリング処理時間を与えないということである。そのため、画像処理フロー編集用の GUI を考える場合には、オペレータの編集が自然に以下のような手順となるように導き、有効なレンダリング処理時間の計測が、より多く行われるような工夫をする必要がある。

- ① 画像処理フローで使用する全ての画像処理とその実行順序を決定させる。
- ② 画像処理のパラメータを調節する。

一方、(b)の条件が必要な理由は、編集用 GUI を使用した簡易レンダリングでは、メモリ上に作成した画像データをファイルに出力せず、ディスプレイに表示することで十分に編集が可能であり、最終的な出力画像の作成を行わないことが考えられるからである。この場合には、レンダリング処理時間の2つの予測方法を併用することが考えられる。すなわち、メモリ上に作成した画像データを BMP などの画像ファイルに変換する処理の処理時間についてはあらかじめ計測しておき、その他の画像処理については実測値を使用するということである。図 2 に示した画像処理フローの例を用いれば、出力映像データ書込処理の処理時間は、あらかじめ計測したものを使用し、その他の入力映像データの読込処理、カラーバランス調整処理などは実測値を用い、両者を足し合わせることでレンダリング処理時間を算出するのである。

また、先に述べたように、簡易レンダリングでは処理する画像の解像度を本レンダリングで行うものよりも小さくして、処理時間の短縮を図る場合がある。レンダリング処理でメモリが不足するなどの事態が発生しない限り、処理時間は、処理画像の面積比に比例すると考えられる。そのため、画像処理フロー設定ファイルに記述する内容は、処理時間と計測時の入力画像解像度とする。

4.2 動画圧縮処理時間の予測方法

これまでの経験から、動画圧縮処理時間に影響を与えるものは、処理画像の解像度、作成する動画形式であると予想された。そのため、これまでに IPD 用コンテンツとしての作成実績があり、現在、本システムで作成可能な動画形式について、画像解像度と動画圧縮処理時間の関係を調べた。その結果を図 25 に示す。ただし、ここでは、圧縮されていない AVI についての処理時間も調べている。これは、本システム

で未対応の動画圧縮ファイルを作成したい場合、無圧縮 AVI を作成し、これを他のアプリケーションで処理することも考えられるためである。

図 25を見ると分るように、画像解像度と動画圧縮処理時間は、ほぼ比例関係にある。そのため、現時点では、図 25から求めた以下の関係式を用いて、動画圧縮処理時間を求めることとする。

$$y = 0.0038x + 0.5 \quad \dots (3)$$

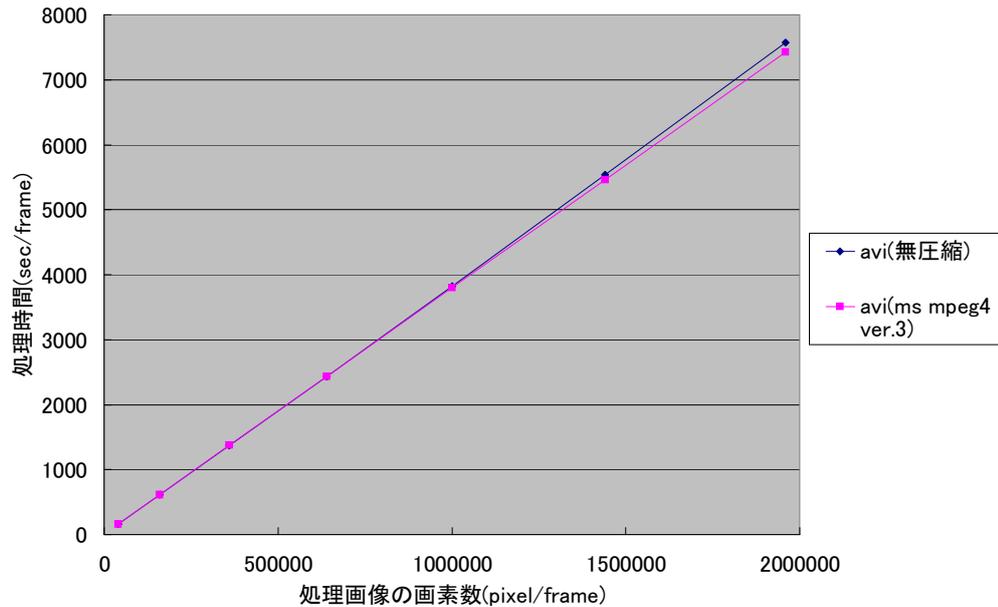


図 25 画像解像度と動画圧縮処理時間の関係

4.3 レンダリング処理と動画圧縮処理の配分決定方法

ここでは、レンダリング処理、動画圧縮処理の予測時間を用いて、与えられた画像処理フローに適したレンダリング処理、動画圧縮処理の処理配分を決める方法について述べる。

プロセス間通信に要する時間を無視した場合、各PCの処理終了時間は式(2)で概算できる。2.5.1の検討では、レンダリング処理の処理時間がどのフレームに大しても一定であるとしたが、この式における単位レンダリング処理時間に4.1節で述べたレンダリング処理予測時間を用いることによって、各配分方法の処理終了時間を予測することができる。ただし、各フレームでレンダリング処理時間が異なるため、どのレンダリング PC にどのフレームの処理が割振られるかというフレームの実行順序も考慮する必要がある。

このようなシミュレーションを事前に行うことにより、システムで使用可能な全ての処理配分に対して、与えられた画像処理フローの処理終了時間を概算する。そして、最も処理時間の短い処理配分を実行することにする。(図 26)

シミュレーションは、使用可能なレンダリング PC の台数、処理フレーム数、レンダリング処理予測時間、動画圧縮処理予測時間を用いて行い、フレームの実行順序については、4.4.2、4.4.3に示す各配分方法の制御手順を元に求めることとする。なお、使用可能なレンダリング PC とは、電源が投入され、CPU を本システム以外の処理に使用されていないレンダリング PC を意味する。

また、動画圧縮処理は、レンダリング処理で作成された出力画像データを入力とする処理である。このため、レンダリング処理の進行状況によっては、処理が待たされる場合があるが、このような影響も考慮して行う。

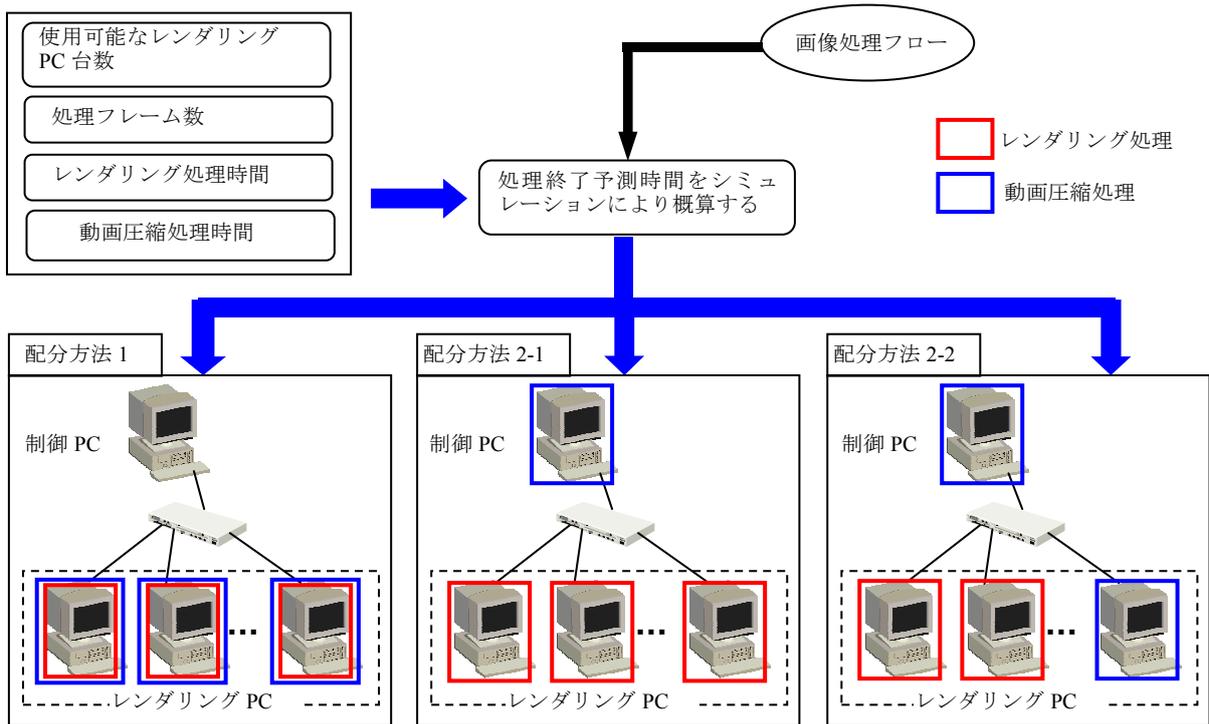


図 26 実行する処理配分の決定方法

以下、各処理配分の制御方法について述べる。

4.4 各処理配分における制御方法

4.4.1 画像データの流れとレンダリング PC に対するディスクアレイへの同時アクセス制限

第2部で後述するように、本システムでは、入力画像データは、テープからディスクアレイに読み込まれ、そこを経由して、レンダリング PC のローカルハードディスクにコピーされる。また、レンダリング PC で処理され、作成された出力画像データもディスクアレイに集められる。

そして、制御 PC が、全てのレンダリング PC に対して、ディスクアレイ上にある入力画像データの送信を行うとともに、レンダリング PC で作成された出力画像データを受信し、ディスクアレイに書き込む。

図 27 は、本システムにおいて、ディスクアレイ上にある画像データを各レンダリング PC のローカルハードディスクにコピーしたときの転送速度を調べた結果である。このようにディスクアレイに対するアクセスを同時に行うことによって、単位時間に転送できるデータ量を増やすことが可能である。

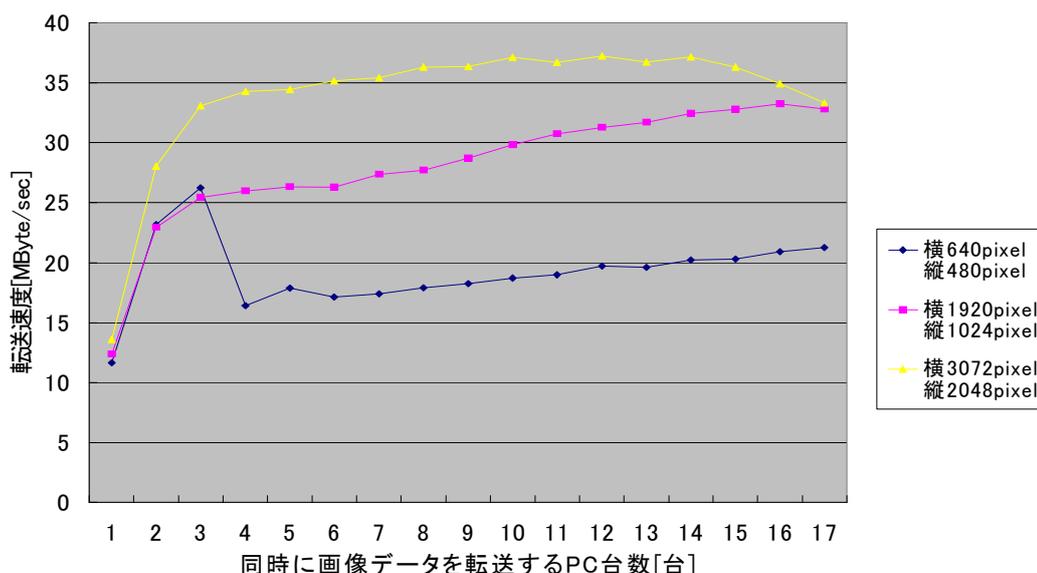


図 27 ディスクアレイからレンダリング PC に対する画像データ送信速度

しかし、同時にディスクアレイにアクセスする PC の台数を増やせば増やすほど、ディスクアレイが接続されている制御 PC の CPU 負荷も増加する。

制御 PCは、レンダリング PC で行われる処理の制御のほか、テープライブラリおよびテープドライブの制御、処理過程のディスプレイへの表示などを行う。そのため、画像データの送受信処理ばかりに CPU を使用することはできない。そこで、本システムでは、同時にディスクアレイにアクセスできるレンダリング PC の台数に制限を設ける。現在のところ、その台数は、2または3台と考えている。

4.4.2 配分方法 1 の制御方法

配分方法 1 では、レンダリング処理と動画圧縮処理を同じレンダリング PC で行う。動画圧縮処理は、フレーム番号の連続する画像データを必要とするため、各レンダリング PC には、連続するフレーム番号の処理を割当てる。各レンダリング PC に割当てるフレーム範囲の決め方としていくつかの方法を考えることができる。最も単純な方法は、処理されるフレーム数が各レンダリング PC で等しくなるように割当てる方法である。これは、例えば、2000フレーム分の処理を20台のレンダリング PC で行う場合に、レンダリング PC0 には、フレーム番号1番～100番までの処理を、レンダリング PC1 には、101番～200番までの処理を行わせるということである。ただし、この方法ではレンダリング処理時間がフレームごとに変化する場合に、各レンダリング PC での処理終了時間にばらつきが出てしまう。そのため、レンダリング処理が早く終了した PC に、他の PC が担当するフレームのレンダリング処理を手伝わせるようにしたり、4.1 で述べた予測時間を元に、あらかじめ各 PC の処理終了時間が均等になるように処理範囲を決めることが考えられ

る。しかし、現時点では、処理フレーム数が等しくなるように割当ての方法を採用する。

次に各レンダリング PC の動作を図 28に示す。各レンダリング PC では、レンダリング処理終了後に動画圧縮処理を行う。それぞれの処理は、同数のフレームを処理する。レンダリング処理は、1フレームごとに入力画像データを制御 PC から受信し、出力画像データを制御 PC に返信する形式で処理を行う。ただし、動画圧縮処理のために、レンダリング処理の出力画像データは、レンダリング PC 側のハードディスクにも残す。

この方法では、各レンダリング PC の処理範囲が事前に分るため、制御PCからレンダリングPCに対して、必要な全ての入力画像データを一度に与えることも可能である。しかし、データ送受信処理は、CPU 負荷の高い処理であり、データ送受信中に並行してレンダリング処理を行うことはしない。前述した、ディスクアレイへのアクセス制限を考慮すると、制御 PC と1つのレンダリング PC が画像データを送受信している間、他のレンダリングPCが画像データの送受信のために待たされる状況が考えられる。このとき、画像データの送受信時間が長いほど、待ち時間が長くなると考えられる。このため、このような待ち時間を最小限に抑えるために、入力画像データを1フレーム分ずつ与えることとする。

また、出力画像データができ次第、これを制御 PC に接続されたハードディスク（RAID 5 構成）にコピーしておくことで、作成したデータの安全性を確保する。

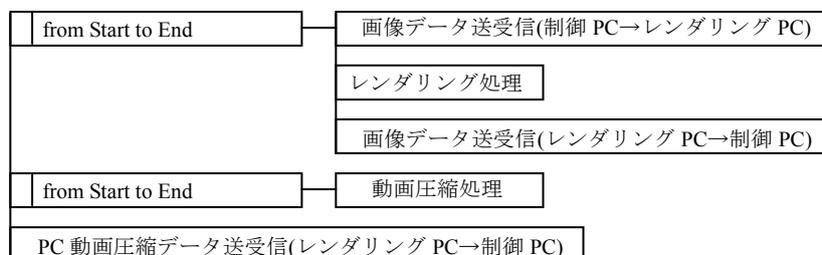


図 28 配分方法 1 の制御手順 (PAD 図)

4.4.3 配分方法 2 の制御方法

配分方法 2 では、各 PC でレンダリング処理か動画圧縮処理のどちらかが行われる。そのため、レンダリング処理で作成された画像データは、動画圧縮処理を行う PC に転送されることになる。動画圧縮処理を行う PC には、フレーム番号の若い画像データから順に転送されてくることが望ましい。また、特定のレンダリング PC に負荷が集中してしまうことは好ましくない。そのため、レンダリング処理を行う PC には、1 回に 1 フレーム分の処理を担当させ（図 29）、処理が早く終わったものから順に次の処理を割振るこ

とにする。

図 26の制御方法2-2にあるように、動画圧縮処理を複数の PC で行う場合には、動画圧縮処理を行う各 PC の処理フレーム数ができるだけ均等になるように処理を配分する。これは、例えば2000フレーム分の動画圧縮処理を2台の PC で行う場合に、1つの PC には、フレーム番号1番～1000番までの動画圧縮ファイルを作成させ、もう1つの PC には、1001番～2000番までの動画圧縮ファイルを作成させるということである。

この場合、レンダリング処理を行う PC は、動画圧縮処理のために分けられたフレーム範囲のうち、最もレンダリング処理の進行が遅れている所の処理を行うように制御し、各フレーム範囲の動画圧縮ファイル作成開始ができるだけ均一になるよう配慮する。

さらに、レンダリング処理を担当している PC は、その出力画像をそのフレームの動画圧縮を行う PC と制御 PC の両方に送信する。これは、配分方法 1 の場合と同様に、出力画像をできるだけ早く RAID 5 構成のハードディスクに移動し、作成したデータの安全性を確保するためである。

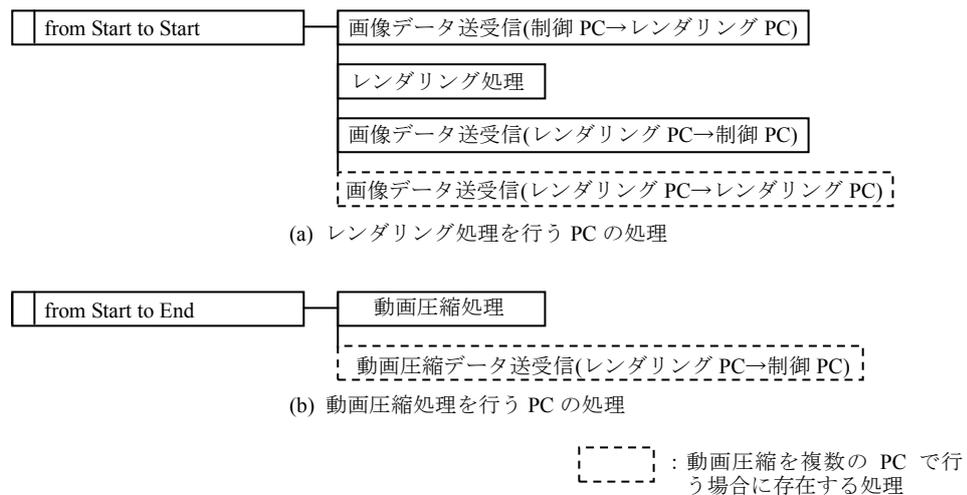


図 29 配分方法 2 の制御手順 (PAD 図)

図 30に 3 台のレンダリングPCが使用可能な場合のシミュレーション例を示す。この例の場合では、配分方法 2-1 が実行されることになる。

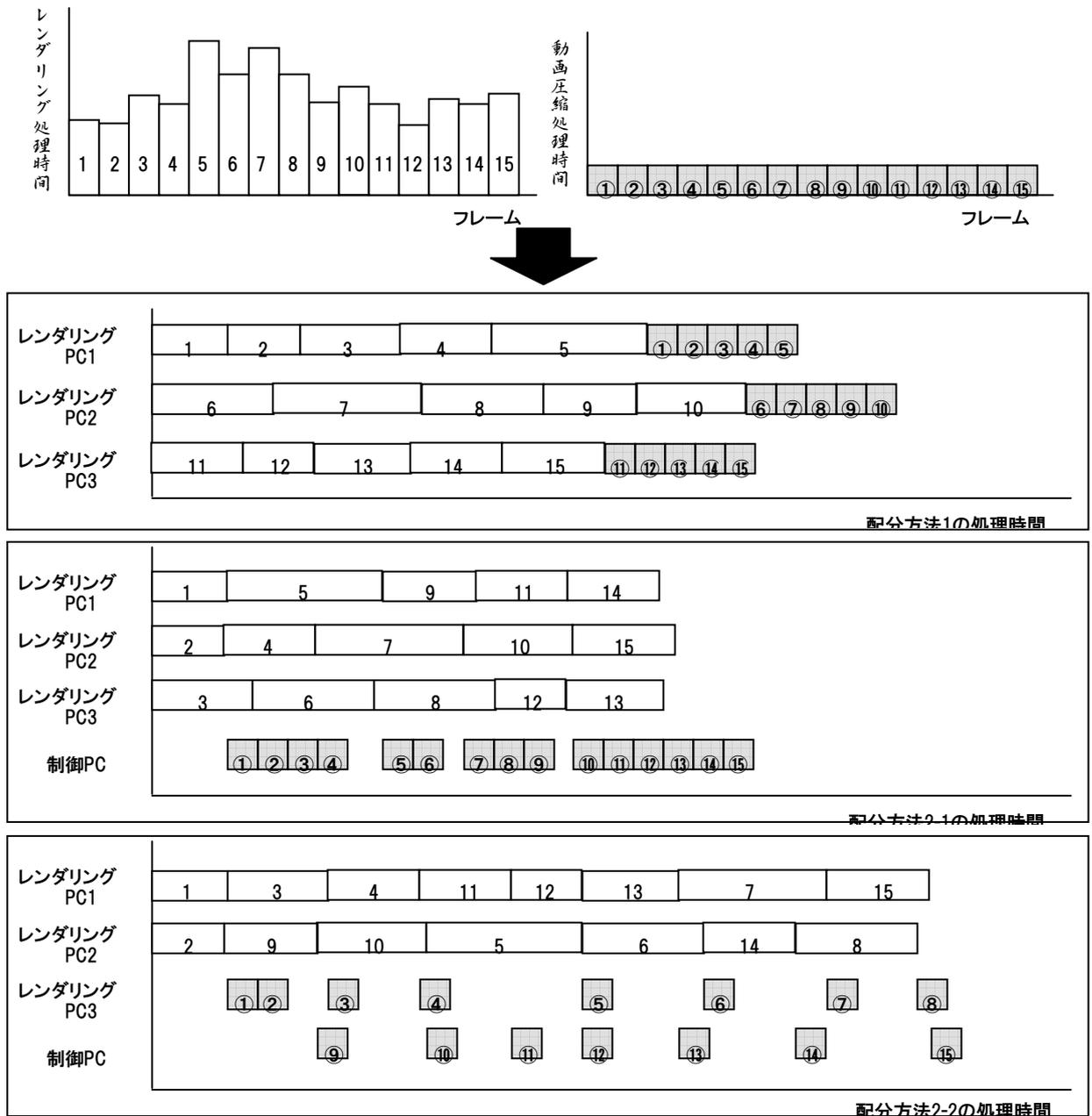


図 30 3台のレンダリング PC が使用可能なときのシミュレーション例

4.5 レンダリングPCのエラー回避方法

ここでは、画像処理フローの実行中にレンダリング PC が何らかの原因で処理または制御不能な状態に陥った場合の回避方法について検討する。

図 31にその基本的な考え方を示す。画像処理フロー実行中にレンダリング PC でエラーが発生した場合(図 31-(b))、システムは、他のレンダリング PC の正常終了を待ち(図 31-(c))、エラーが発生したレンダリング PC に担当させていた処理を実行させる。

このとき、図 28、図 29に示したように、配分方法1、配分方法2のどちらを実行した

場合でも、レンダリング処理で作成された画像データは、速やかに制御 PC にコピーまたは移動される。このため、レンダリング PC がエラーを起こす直前までに作成されたデータは、全て制御 PC 側にあることになる。そのため、例えば、図 31で、PC19 に PC2 の処理を実行させる場合、既に PC2 で終了しているフレームの処理は PC19 において実行しないようにする。ただし、配分方法1で実行している場合には、レンダリング処理の後に、動画圧縮処理を行う必要があるため、PC2 ですでに作成済みの画像データを制御 PC から PC19 にコピーする必要がある。また、配分方法2を実行している場合で、エラーを起こしたレンダリング PC が動画圧縮処理を行っていた場合には、前述の場合と同様、制御 PC から必要な画像データを処理依頼先のレンダリング PC にコピーする必要がある。

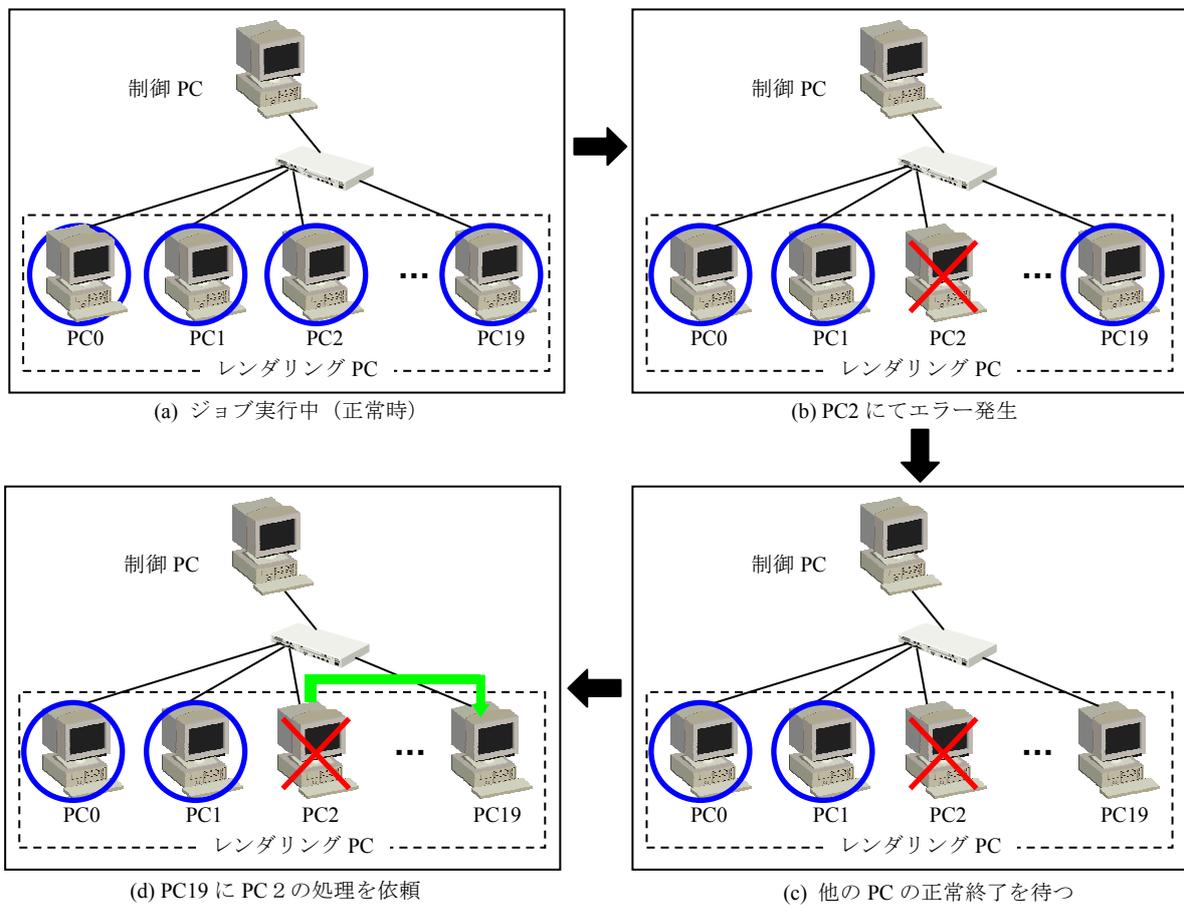


図 31 エラーの回避方法

また、4.4.2 で述べたように、配分方法1では、各レンダリング PC の担当範囲を各レンダリング PC の処理フレーム数が等しくなるように割り当てる。しかし、図 31に示すエラー回避を考えると、あるレンダリング PC でエラーが発生した時、そこで実行を予定していた処理は、最も早く正常終了した PC が1台で担当することになる。このとき、その後処理を終えた他のレンダリング PC は、ただエラー回避処理を行っている PC の処理終了を待つことになり、効率が悪くなる。

そこで、ジョブ実行開始時には、各レンダリング PC で処理されるフレーム数が等しくなるように担当フレームを割り当てが、レンダリング PC でエラーが発生した場合には、そこで実行する予定だった範囲を、もう少し細かく分割し、正常終了した複数のレンダリング PC が同時に処理分担できるようにする(図 32)。

ただし、動画圧縮処理の特徴を考えると、1つの分割領域は、それほど小さくできない。最低でも20フレーム程度にする必要があると考える。

なお、エラーが発生したレンダリング PC は自動で再起動させ、正常に起動した場合には、次の画像処理フローの実行に参加させるようにする。

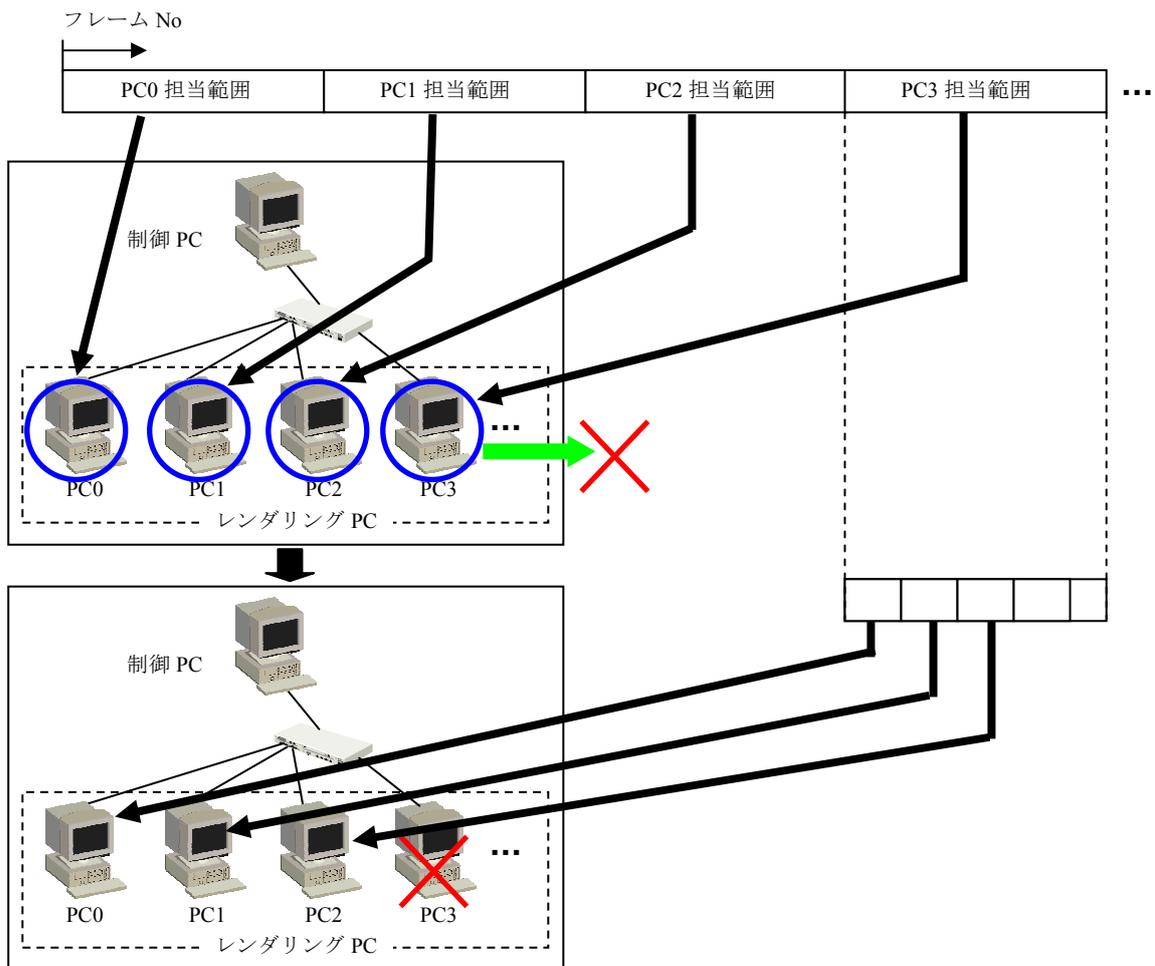


図 32 エラーの回避方法 (配分方法 1 の場合)

5. 処理をビジュアルに一元管理できるジョブ管理とそのGUI表示方法の検討

これまで、1つの画像処理フローの実行を考えてきた。本章では、複数の画像処理フローを効率的に実行することを考える。

クラスタ上で実行されるジョブは、その性質によってインタラクティブジョブとバッチジョブの2つに分類することができる[Doshisha3]。

インタラクティブジョブとは、ディスプレイへの出力や、ユーザによる入力が多く必要なジョブである。このジョブは、素早いレスポンスが要求されるが、一般に計算機への

リソース要求は小さい。また、バッチジョブとは、キューに投入されたジョブが、リソースが空くのを待ち、順番に実行されるものである。こちらは、素早いレスポンスはあまり必要でないが、1つのジョブが大きなメモリ空間や、CPUタイムを占有して処理される。

本システムのようなノンリニア編集システムでは、画像処理フローの編集作業がインタラクティブジョブに分類され、画像処理フローの実行（本レンダリング）がバッチジョブに分類されると考えられる。

以下で検討する内容は、全てバッチジョブに関するものであり、本章で用いるジョブとは、画像処理フローの実行のことを意味する。

5.1 ジョブのスケジューリング

複数のジョブを処理しようとした場合、オペレータが置かれた状況により、システムに対する要求は変わるものと考えられる。例えば、既に出荷したコンテンツの手直しなど、急ぎの作業が入ってきた状況では、現在行っているコンテンツ制作に優先させて、それを実行したいと考えるだろう。また、このような急ぎの作業がない場合には、投入する複数のジョブをできるだけ早く処理して欲しいと考えられる。

そこで、本システムでは、ジョブに対して優先順位を設け、基本的に、その優先順位に従ってジョブを実行する。ただし、同じ優先順位に投入されたジョブが複数存在する場合、それらの総所要時間が最小になるように、システムが実行順序を決定することにする。（図 33）

これにより、オペレータが実行順序を決めたい場合であれば、各ジョブに異なる優先順位を割り当てることによって、ジョブの実行順序を制御することができるし、投入した複数のジョブをできるだけ早く終了させたいと思うのであれば、複数のジョブに同じ優先順位を設定すれば良いことになる。

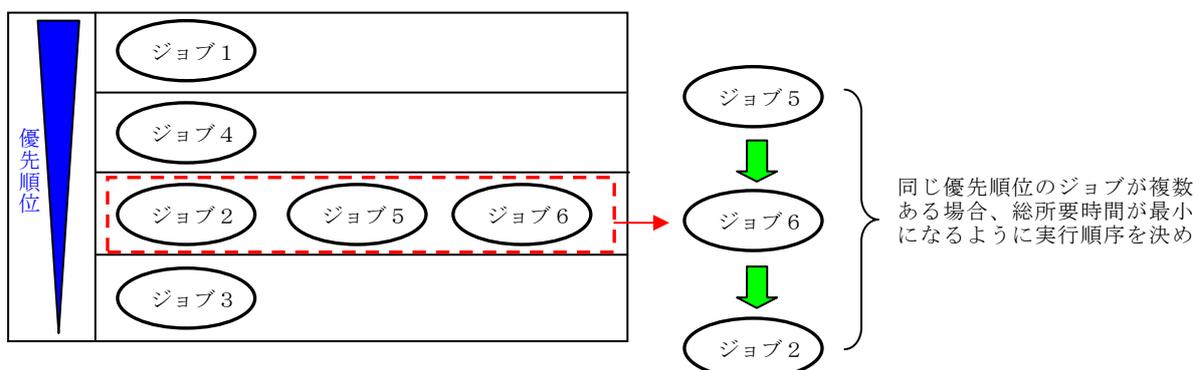


図 33 画像処理フロー実行におけるスケジューリング

5.2 ジョブの状態遷移

ジョブには、初期状態、実行可能状態、実行状態の3つの状態を設ける。初期状態

とは、ジョブの実行順序が確定していない状態を意味し、実行可能状態とは、実行順序が確定した状態を意味する。また、実行状態とはシステムがそのジョブを実行中であることを意味する。

ジョブが投入されると、投入されたジョブは初期状態になる。また、この時、投入されたジョブ以下の優先順位を持つジョブが実行可能状態から初期状態に戻され、初期状態にある複数のジョブに対して再度スケジューリングが実行され、それぞれのジョブの実行順序が決定される。また、既に投入されているジョブの優先順位が変更された場合にも、変更されたジョブ以下の優先順位を持つジョブが初期状態に戻され、再度スケジューリングが行われる。

スケジューリングを終えたジョブは、実行可能状態に移行される。また、実行可能状態にあるジョブは、他のジョブの終了により、実行状態に移行される。

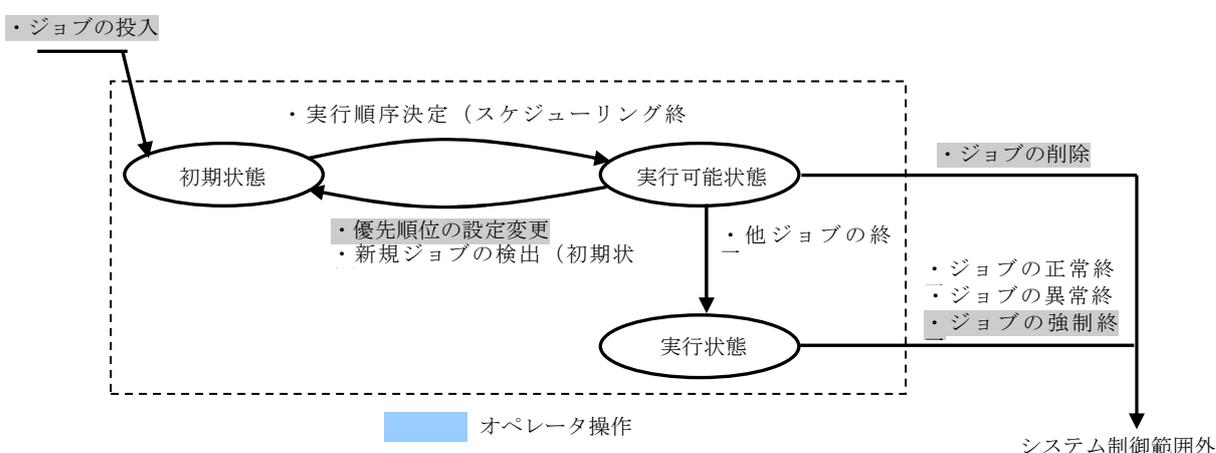


図 34 ジョブの状態遷移

5.3 総所要時間が最小となる実行順序の決定方法

スケジューリング問題を分類すると図 35のようになる[圓川 1996]。まず、それぞれのジョブを完了するために、ジョブの中で行われる各工程に決められた順序（順序制約）があるかどうかで分類される。このような順序制約がないものは、巡回型スケジューリングと呼ばれ、順序制約があるものは、資源配置型スケジューリングと呼ばれる。さらに、どちらのスケジューリング問題も、対象とするジョブの数によって単一ジョブと複数ジョブに分けられる。

巡回型スケジューリングで、単一ジョブを対象とするものは巡回セールスマン問題と呼ばれ、複数ジョブを対象とするものは車両配送計画問題と呼ばれる。

また、資源配置型スケジューリングで、単一ジョブを対象とするものはプロジェクトタイプのスケジューリングと呼ばれ、複数ジョブを対象とするものはジョブショップタイプのスケジューリングと呼ばれる。さらに、ジョブショップタイプのスケジューリングは、ジョブショップスケジューリングとフローショップスケジューリングに分類される。ジョブショップスケジューリングとは、ジョブの中で行われる各工程の順序が、ジョブごとに異なる場合のスケジューリング問題である。また、フローショップスケジューリングと

は、各工程の順序が同じ場合のスケジューリング問題である。

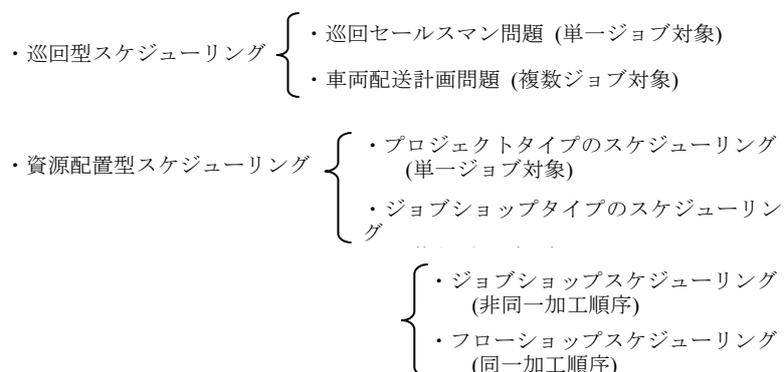


図 35 スケジューリングの分類

第2部において後述するが、本システムにおけるジョブである画像処理フローの実行での基本的なデータの流れは、テープ上にある入力画像データがディスクアレイにコピーされ、これをシステム内の複数のPCでレンダリング処理および動画圧縮処理を実行して出力画像データおよび出力動画圧縮データを作成する。そのため、大まかにジョブの工程を分けるとすれば、テープからディスクアレイ上に画像データを読み込む入力画像データ準備処理と、その入力画像データを用いて出力画像データや出力動画データの作成を行うレンダリング処理および動画圧縮処理の2つに分けることができる。また、必ず、入力画像データ準備処理の後にレンダリング処理および動画圧縮処理を実行する。そのため、本システムで取り扱うジョブは、工程数2のフローショップスケジューリングであると考えられる。

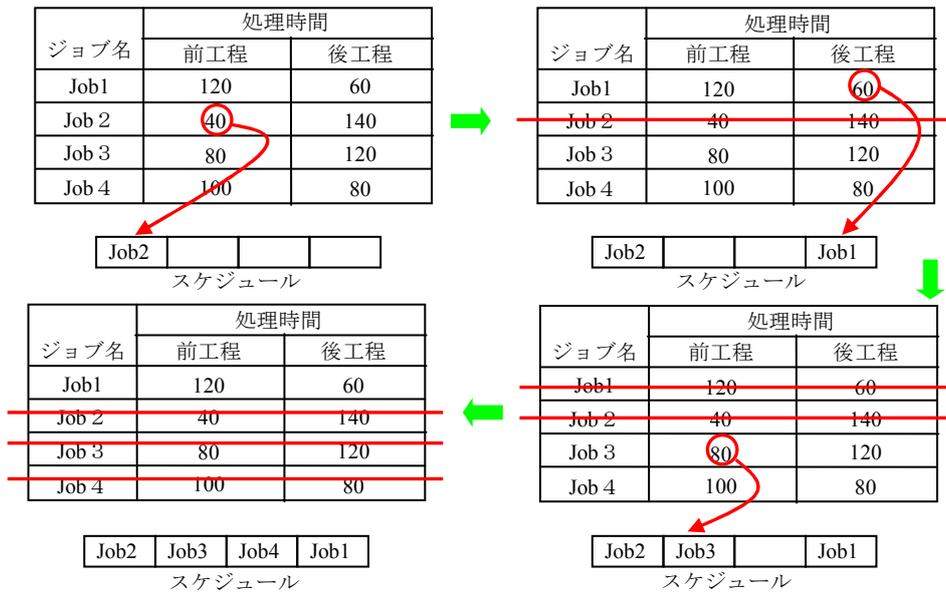
スケジューリングの目標が総所要時間最小で、工程数が2のフローショップの場合に、簡単に最適解を求める方法としてジョンソン法が知られている。本システムでは、優先順位が同じジョブの実行順序を決める際、このジョンソン法を使用するものとする。

5.4 ジョンソン法

ジョンソン法とは、2つの工程を前工程、後工程とした場合、前工程での処理終了に対する後工程での待ち時間を最小にするために、前工程は処理時間の短いジョブの順に、後工程は処理時間の長い順にジョブを割り付けるものである。そのため、まだ実行順序の決まっていない全てのジョブの中から、最も処理時間の短い工程を含むものを探し、それが前工程のものであれば、スケジュールの前の方に割り付け、それが後工程のものであれば、スケジュールの後の方に割り付けるというものである(図 36)。

ジョブ名	処理時間(min)	
	前工程	後工程
Job1	120	60
Job2	40	140
Job3	80	120
Job4	100	80

(a) スケジューリングするジョブと処理時間



(b) スケジューリング決定手順



(c) スケジューリング結果

図 36 ジョンソン法

5.5 GUI表示方法

ここでは、実行中のジョブの状態をディスプレイ上に表示することを考える。実行中のジョブに対して、オペレータに知らせる必要のある情報を抽出し表 7、表 8にまとめる。

表 7 ジョブの全体内容

No.	表示項目	概要
1	処理開始フレーム番号	そのジョブで処理される開始フレーム番号を示す。
2	処理終了フレーム番号	そのジョブで処理される終了フレーム番号を示す。
3	レンダリング処理進行状況	レンダリング処理の進行状況を示す。
4	動画圧縮処理進行状況	動画圧縮処理の進行状況を示す。
5	終了予測時間	そのジョブの終了予測時間を示す。

表 8 ジョブの詳細内容

No.	表示項目	概要
1	状態	PCの正常・異常を表示する
2	処理内容	そのPCで行われる処理がレンダリング処理であるか動画圧縮処理であるか、その両方であるかを示す。
3	処理開始フレーム番号	そのPCで行われる処理の開始フレーム番号を示す。
4	処理終了フレーム番号	そのPCで行われる処理の終了フレーム番号を示す。
5	実行中の処理内容	そのPCが現在実行している処理がレンダリング処理であるか動画圧縮処理であるかを示す。
6	実行中のフレーム番号	そのPCが現在処理しているフレーム番号を示す。
7	CPU負荷率	そのPCのCPU負荷率

本システムでは、21台ものPCが処理を行う。オペレータが1つの画面上で全てのPCの処理状況を見れるように考慮し、ジョブの全体内容、詳細内容の表示を図 37、図 38のように考えている。

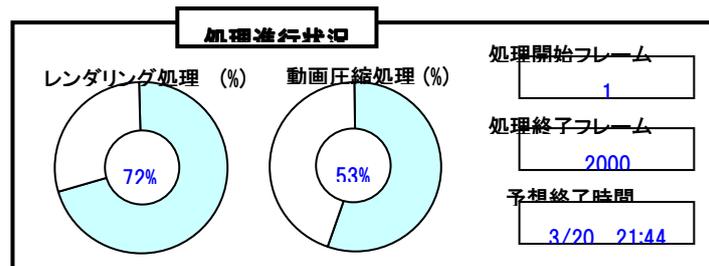


図 37 ジョブの全体内容の表示

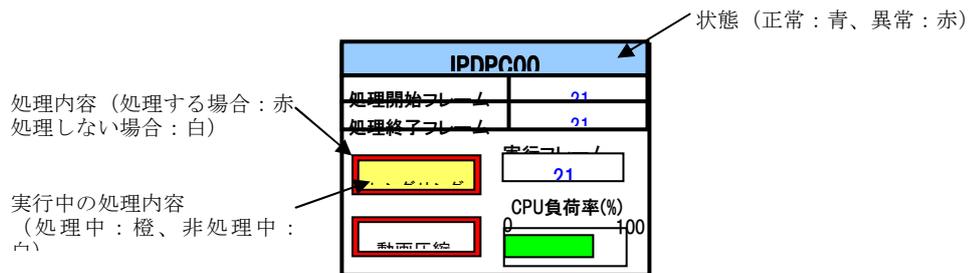


図 38 ジョブの詳細内容の表示

第 2 部 超大容量データマネージメント技術

ここでは、超大容量の画像データをネットワーク接続した複数の PC から効率的に読み書きするために、ディスクアレイ、大容量テープデバイスならびにネットワーク接続した複数の PC に置かれたローカルハードディスクの3つの記憶装置を用いて、実運用を考慮に入れた最適な管理方法について検討を行う。

1. 一般的なノンリニア編集システムの構成と IPD 用コンテンツ制作における問題点

一般的なノンリニア映像編集システムでは、図 39に示すように、映像編集を行う計算機、映像編集の素材となる入力画像データや映像編集後の出力画像データを格納するハードディスク、さらに、映画会社や映像制作会社などから供給される入力画像データや、出力画像データを保管するための補助記憶装置(主にテープ)などから構成される。



図 39 一般的なノンリニア映像編集システムの構成

これまで我々も、このようなシステムを用いて、IPD 用コンテンツを制作してきた。しかし、IPD 用コンテンツ制作では、投影する映像が高精細・広視野角であるため、扱う入力画像データも高解像度・大容量なものとなり、ハードディスク容量の不足が

問題となっていた。

例えば、横6000ピクセル、縦3000ピクセルの入力画像データに何らかの処理を加え、解像度を変更せずに出力画像データを作成する場合を考えると、1フレームの出力画像データのデータ量は、約72MByte(非圧縮、1ピクセル4Byte表現)となる。このサイズの画像を用いて3分間のコンテンツを制作しようとした場合、コンテンツのフレームレートを30frame/secと仮定すると、出力画像だけで約389GByte(72MByte×30frame/sec×180sec)となる。

通常、合成やフェードアウトなど、1フレームの出力画像データを作成するために複数の入力画像を必要とすることを考えると、3分程度のコンテンツ制作に1TByte以上の容量を持ったハードディスクが必要となる場合も少なくない。

そのため、最小限必要な画像データのみをハードディスク上に置き、必要のないものは一時的にテープに保存するなどして、限られたハードディスク容量を有効に活用する工夫が人手で行われてきた。例えば、事前に、作成される出力画像のデータ量の予測および、ハードディスク空き容量の確認を行い、空き容量が足りない場合には不要なデータを削除したり、編集済や編集途中の出力画像データを一時的にテープに移動したりして、ハードディスクの空き容量を確保するのである。

また、出力画像データを作成する場合、それに必要な全ての入力画像データをハードディスク上に置く必要がある。そのため、入力画像データをテープなどの補助記憶装置から読み込み終わるまで、出力画像データの作成は開始しない。これらの処理を同時に行うことも可能であるが、これらの処理を1台の計算機で行った場合、非常に処理効率が落とす場合がある。この時、必要な空き容量がハードディスク上に確保できない場合、上で述べた不要なデータの削除や必要なデータのテープへの移動を行い、ハードディスク上に空き容量を確保することになる。そのため、オペレータは、このような出力画像作成のための準備作業が終わるまで、システムの状態を監視する必要がある。使用する映像編集システムによっては、テープなどからのデータ読み込みやテープなどへのデータ書き込みと出力画像データの作成開始をバッチファイルなどで制御する手段が用意されている場合も考えられる。しかし、その場合でも、必要な入力画像データが複数のテープに記録されていたり、テープに移動しなければならないデータ量が多く1本のテープに入りきらない時などのように、テープの入替え作業が発生する場合には、その作業が終了するまで、オペレータはシステムの状態を監視しなければならない。

さらに高解像度・大容量の映像データを処理するには、膨大な時間を必要とする。上記の例で考えると、仮に1フレームの出力画像データの作成に2分かかったとして、全ての出力画像データを作成するには、180時間も必要なのである。そのため、夜間に計算機を無人で運転させる場合も少なくない。しかし、ハードディスクの空き容量の管理を人手で行う場合、必要なハードディスク空き容量の見積を誤ることがしばしば起きていた。これにより、出力画像データの作成は中断され、時間が浪費されてしまう。映像編集は、コンテンツ制作作業の最終段階に位置し、締め切りまでの時間が少ないことから、このような時間の浪費が許されない状況であることが多い。

2. 各記憶装置の特徴

本システムには、大容量テープドライブ、ディスクアレイ、レンダリングPCのローカルハードディスクという3つの大容量記憶装置がある。以下に、その特徴を示す。

① 大容量テープドライブ

大容量テープドライブは、ハードディスクに比べて低速である。また、シーケンシャルアクセスであるため、記録した順番にアクセスできない場合、非常に効率が悪くなる。しかし、使用頻度や保存状態にもよるが、一般にハードディスクの寿命が5年から7年といわれるのに対して[Dnki]、テープの寿命は、20年以上といわれ[Sony c]、データの長期保存に適している。また、精密機器であるハードディスクに比べ、テープは振動や衝撃に強く、持ち運びが便利である。そのため、映画会社や映像制作会社などとのやり取りや、顧客への納入などにおいて、テープが使用される場合が多い。

さらに、テープライブラリと組み合わせることによって、テープの自動入替えが可能となる。空テープの在庫と、テープライブラリ内のテープおよびデータの管理を確実に行うことができれば、仮想的に無限の容量を持った記憶装置と見なすことが可能であると考えられる。

② ディスクアレイ

ディスクアレイは、大容量で高速にランダムアクセス可能である。また、RAID5構成にしているため、ディスクアレイを構成する複数台のハードディスクのうち、1台が故障しても、データを復旧することができる。そのため、レンダリングPCのローカルハードディスクに比べて信頼性が高い。

また、一般に、RAID5では、データの読み込みは高速であるが、書き込みは、読み込みに比べて低速である[Ascii24]。これは、書き込みを行う場合、データ復旧時に使用するパリティビットの計算を必要とするためである。

③ レンダリングPCのローカルハードディスク

レンダリングPCのローカルハードディスクは、ディスクアレイと同様にランダムアクセスが可能であるが、容量が小さく低速である。また、故障した場合、データを復旧することは困難である。

ただし、レンダリングPCは、その名称の通り、レンダリング処理などの出力画像データの作成を主として行うPCである。これらの処理の作業領域を考えた場合、ローカルハードディスクは、アクセス速度が速く、大容量な記憶装置といえる。

以上より、各記憶装置の特徴を表 9にまとめる。

表 9 各大容量記憶装置の特徴

項目	ディスクアレイ	大容量テープドライブ	レンダリングPCのローカルハードディスク
信頼性	RAID5構成 △	一般に20年以上 ○	一般に5年～7年 ×
保存容量	1.3TB △	200GB (テープライブラリとして最大7TB) ○	160GB ×
データ読み込み速度	ランダムアクセス RAID5構成 ○	シーケンスアクセス ×	ランダムアクセス △
特記事項		<ul style="list-style-type: none"> ・テープライブラリと組み合わせることによってテープドライブへのテープ自動入れ替えが可能 ・テープを入れ替えることで、仮想的に無限の記憶容量を持った記憶装置と見なせる 	

本システムでは、テープで供給された入力画像データを各レンダリング PC で処理し、出力画像データをテープに保存するというデータの流れを基本に考える。

以下、テープ上にある入力画像データを、効率的に各レンダリング PC に供給する方法および各レンダリング PC で作成された出力画像データをテープに書き込む方法について、実運用を考慮し、3章および4章で検討する。

3. 入力画像データの供給方法

3.1 供給経路

テープに記録されている入力画像データを各レンダリング PC に供給する経路として、図 40に示す2通りの方法が考えられる。

1つは、テープから直接レンダリング PC に供給する方法(図 40-(a))であり、もう1つは、ディスクアレイを経由して供給する方法(図 40-(b))である。

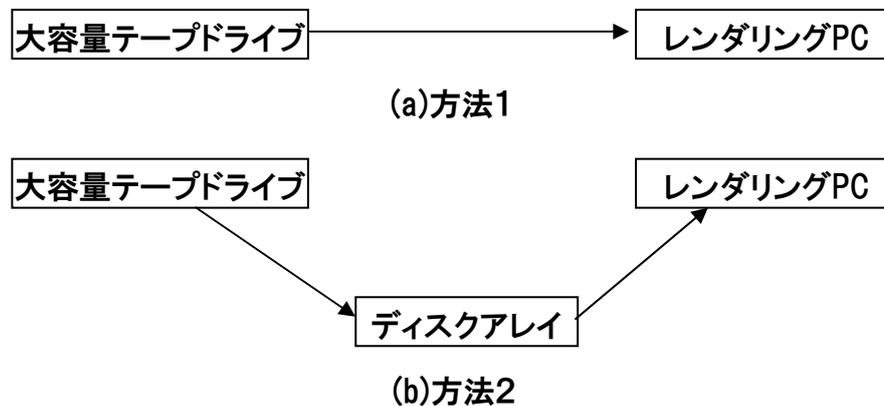


図 40 入力画像データの供給経路

この2つの方法を比較すると、前者の方法は、ディスクアレイの記憶容量を消費しない。また、ディスクアレイを経由せず、レンダリング PC に対して直接データを供給するため、1台のレンダリング PC に対して入力画像データを供給することを考えた場合、その処理時間は後者の方法に比べて少ないと考えられる。しかし、テープはシーケンシャルアクセスである。したがって、複数のレンダリング PC に対して同時にデータを供給することはできない。また、画像処理を行っていたレンダリング PC で異常が発生し、そのレンダリング PC で実行する予定であった画像処理を他のレンダリング PC に振り替えて行おうとした場合、入力画像データの再読み込みが発生し、テープの巻き戻しや早送り、さらにはテープの入替えなどの動作が入る。そして、その間、全てのレンダリング PC への入力画像の供給が停止してしまう。

これに対して、後者の方法は、ディスクアレイの記憶容量が消費されると共に、ディスクアレイを経由することによって、1台のレンダリング PC に入力画像データを供給するために必要な時間が、前者の方法に比べて多く必要である。

しかし、シーケンシャルアクセスであるテープからランダムアクセス可能なディスクアレイにデータを格納することにより、複数のレンダリング PC に対して、同時にデータを供給することが可能となる。また、この方法では、以前に処理した入力画像データがディスクアレイ上に残っていることが考えられる。そのため、テープからの読み込みを開始する前に、ディスクアレイに対してデータの存在確認を行うことによって、低速のテープからのデータ読み込み頻度を減らすことが可能である。さらに、入力画像データをテープからディスクアレイに読み込む処理と、ディスクアレイからレンダリング PC に供給する処理は、独立して行うことが可能である。そのため、出力画像の作成を、コンテンツのシーンごとに分割して行う場合のように、複数のジョブに分けて実行

する場合、シーン1の出力画像作成中に、シーン2の入力画像データをテープからディスクアレイに読み込むといった、入力画像データの先読みを行うことができる。これにより、ディスクアレイを経由することで生じる時間の増加を抑えることが可能だと考える。

以上から、本システムにおいては、テープに記録されている入力画像データをレンダリング PC に供給する場合、ディスクアレイを経由する方法で行うことにする。

3.2 供給方法

次に、ディスクアレイからレンダリング PC に入力画像データを供給する方法について検討を行う。レンダリング PC 側で行う画像処理は、レンダリング PC のメモリを用いて行われる。そのため、レンダリング PC に入力画像データを供給する場合、レンダリング PC のメモリに直接データを書き込む方法(図 41-(a))と、レンダリング PC のローカルハードディスクに書き込んだ後、メモリに読み込む方法(図 41-(b))が考えられる。

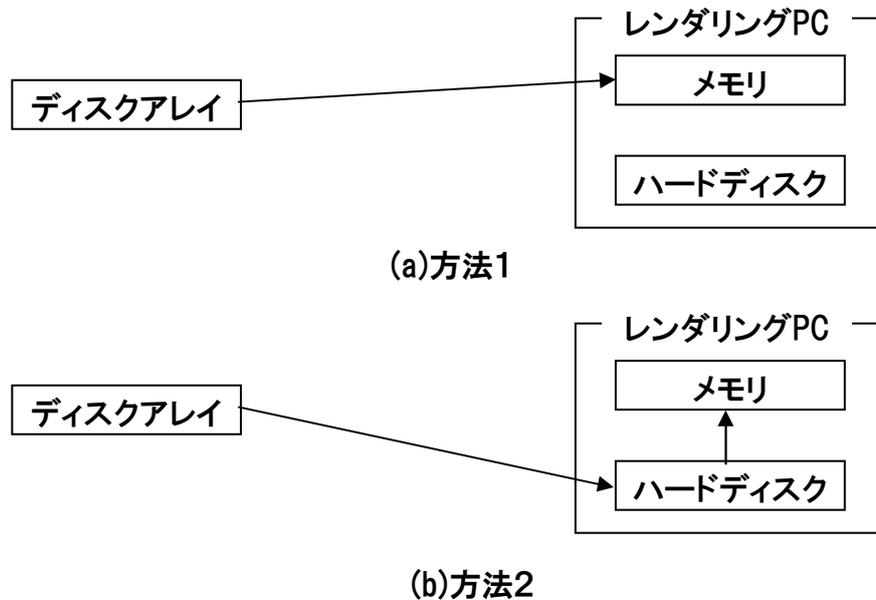


図 41 入力画像データの供給方法

この2つの方法を比較すると、前者の方法は、後者の方法に比べて高速であると考えられる。しかし、メモリに比べてハードディスクは大容量であるため、後者の方法では、そのレンダリング PC に対して供給された過去の入力画像データを保管しておくことが可能である。レンダリング PC のローカルハードディスクは、ディスクアレイに比べて信頼性が低い、そのため、重要なデータの長期保存には適していないと考えられる。しかし、入力画像データの場合、テープ上に元となるデータが存在し、レンダリング PC のローカルハードディスクにあるデータは、あくまでもそのコピーである。そのため、故障によるデータの損失が大きな問題となることはないと考えられる。

通常、あるコンテンツの作成を行っている期間は、同じ入力画像データを使用する

確率が高い。そのため、過去にレンダリング PC で使用した入力画像データをローカルハードディスク上に保管しておき、ディスクアレイからレンダリング PC に対してデータの供給を行う前に、そのデータの存在確認を行うことによって、ディスクアレイからレンダリング PC に対して供給しなければならないデータ量を減らすことが可能である。

本システムの場合、20台ものレンダリング PC に対して、1台のディスクアレイから入力画像データの供給を行う。そのため、供給しなければならないデータ量を減らすことは、非常に重要であると考えられる。

以上から、レンダリング PC に入力画像データを供給する場合、レンダリング PC のローカルハードディスクに書き込むこととする。さらに、レンダリング PC で過去に使用した入力画像データは、ハードディスクの容量が許す限りにおいて保管し、これを積極的に使用するものとする。

4. 出力画像データの取得方法

各レンダリング PC で作成された出力画像データは、できるだけ速やかにディスクアレイに移動または、コピーすることにする。これは、レンダリング PC のローカルハードディスクの信頼性が低く、大切な出力画像データの保存には適さないと考えるためである。

また、ディスクアレイ上の出力画像データは、オペレータが確認したあとにテープへ格納するものとする。これは、作成された出力画像データが、必ずしも、オペレータの思い描く結果だという保証がないためである。オペレータは、細心の注意を払ってジョブを作成する。しかし、操作ミスや思い違いによって、作成結果が自分の思い描いていたものと異なる場合があるのである。

5. 各記憶装置のデータ管理方法

5.1 管理方針

ここでは、大容量テープドライブ（テープライブラリ含む）、ディスクアレイ、レンダリングPCのローカルハードディスクにおけるデータの管理方針について述べる。

3.1、3.2 で述べたように、本システムでは、テープからディスクアレイ、ディスクアレイからレンダリング PC のローカルハードディスクへデータを移動またはコピーする場合、事前にそのデータが相手先にあるかどうかの存在確認を行う。そして、データが存在しない場合のみデータの移動またはコピーを行う。この存在確認をスムーズに行うためには、各記憶装置が共通の方法でデータを管理することが望ましい。

通常、コンテンツ制作は、シーンごとに分けて編集される。そのため、本システムでは、入力画像データ、出力画像データをコンテンツごとに分け、1つのコンテンツを複数のシーンに分けて管理可能なディレクトリ構成(図 42)でデータを管理する。

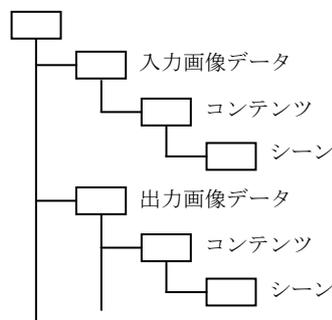


図 42 ディレクトリ構成

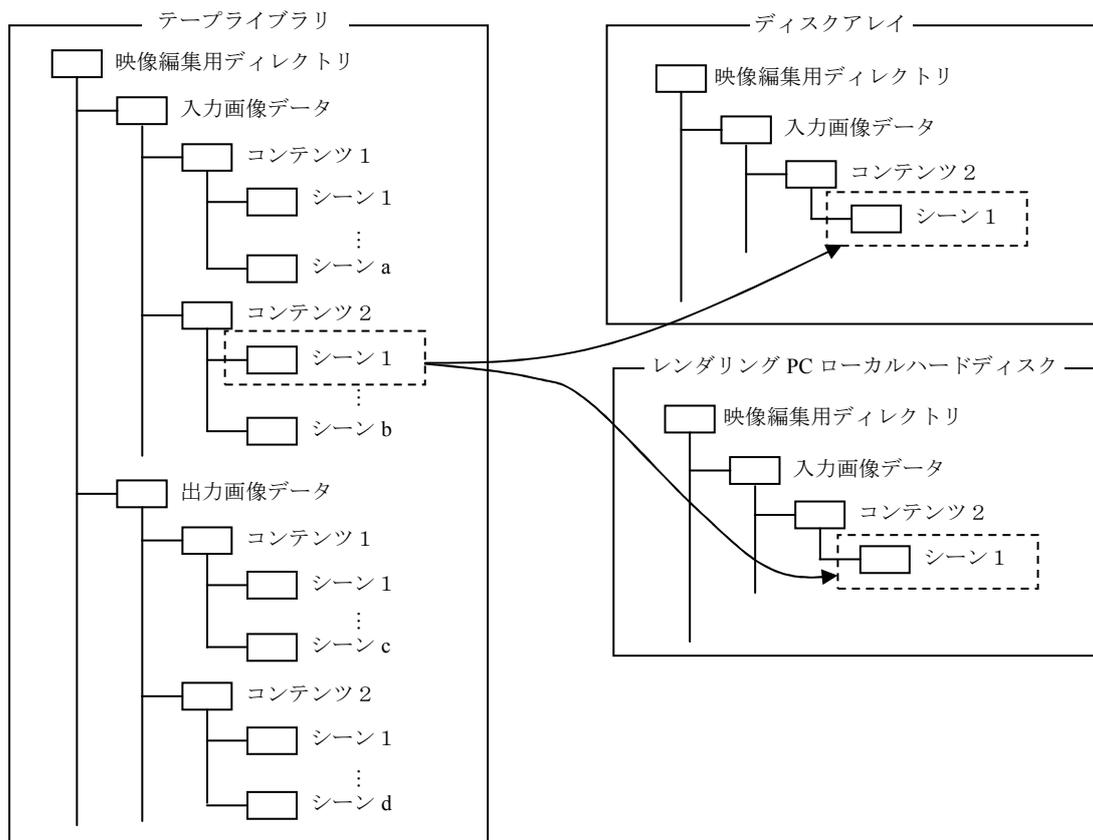


図 43 各記憶装置間のデータ移動・コピー

これは、例えば図 43に示すように、テープライブラリ上で、映像編集用ディレクトリ¥入力画像データ¥コンテンツ2¥シーン1以下にある画像データをディスクアレイやレンダリング PC のローカルハードディスクなどの記憶装置に移動またはコピーしても、それは同じ映像編集用ディレクトリ¥入力画像データ¥コンテンツ2¥シーン1以下に保存されるということである。(ただし、映像編集用ディレクトリにあたる物理的な位置は、各記憶装置で異なる。)

また、各記憶装置間でデータの移動や削除といった操作が行われるが、これは、

基本的にシーンごとに行うものとする。

5.2 ディスクアレイにおけるデータの管理

3章、4章で述べたように、本システムでは、全ての入力画像データ、出力画像データがディスクアレイを経由して、テープドライブとレンダリングPCのローカルハードディスクとの間でやり取りされる。そのため、複数のジョブを確実に処理するためには、ディスクアレイの管理が非常に大切である。

本システムでは、これまで人手で行ってきた、出力画像データの作成に必要なディスク容量の予測およびハードディスク空き容量の確認といった作業を、ジョブの投入時にシステムに行わせる。さらに、このときハードディスク空き容量が不足する場合には、削除またはテープに移動させる入力画像データおよび出力画像データをオペレータに選択させる。なお、このときの選択もシーン単位で行う。

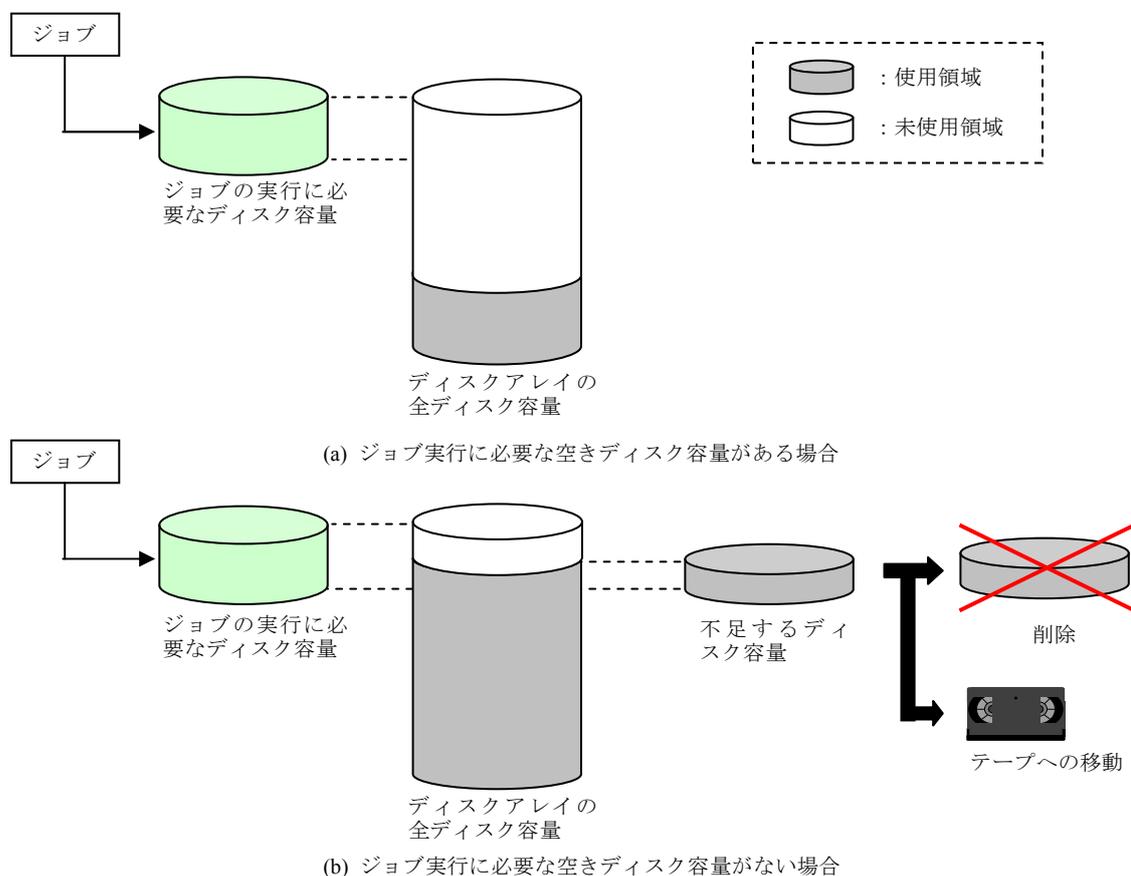


図 44 ディスクアレイにおけるディスク容量の確保

システムは、ジョブの実行に際して、空き容量の不足を検出した場合、オペレータに選択された入力画像データおよび出力画像データを削除またはテープに移動して、ディスクアレイの空き容量を確保する（図 44）。

なお、空き容量の確保では、テープの移動よりも削除を優先して行うこととする。これは、処理時間を考えた場合、テープにデータを移動するよりも削除のほうが早いからである。

また、基本的に、5.4 で後述するテープ管理テーブルに存在するデータは削除とし、存在しないデータは、テープへの移動を行うことによってディスクアレイ上の空き容量を確保する。これは、テープ管理テーブルに存在するデータは、テープに記録されたデータのコピーであるためである。

5.3 レンダリングPCのローカルハードディスクにおけるデータの管理

レンダリング PC のローカルハードディスクには、その PC で以前に処理された入力画像データが管理される。この入力画像データも、シーン単位に管理する。前述したように、あるコンテンツの作成を行っている期間は、同じ入力画像データを使用する確率が高い。そのため、各シーンで、そのシーンに含まれる画像データに対する最終アクセス日時を監視することで、そのシーンに対する最終アクセス日時を管理する。そして、空き容量の確保を指示された場合、ディスク内の空き領域の容量が指示された値以上になるまで、最終アクセス日時の古いシーンの削除を実行する。

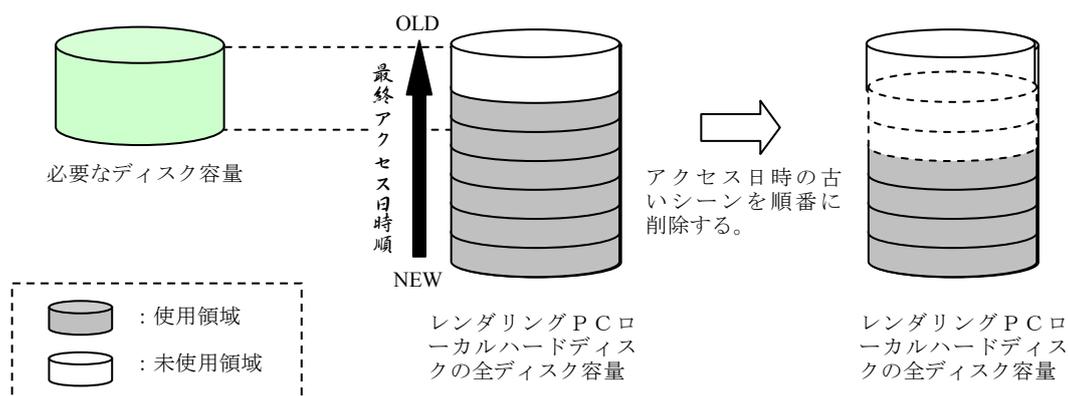


図 45 レンダリングPCのローカルハードディスクにおけるディスク容量の確保

5.4 大容量テープドライブ（テープライブラリ含む）におけるデータの管理

本システムで使用するテープライブラリには、バーコードリーダーが搭載されている。そのため、テープにシリアルな番号（テープ番号）を割当てて、それをバーコードとして貼り付けることで、テープ1つ1つの識別が可能である。そのため、テープ番号を指定することによるテープからのデータの読み込みや、テープへの書き込みといった動作を実現することは、それほど困難ではない。

しかし、本システムでは、テープ上にあるデータも図 42に示すような論理パス名で指定できるようにする。これによって、各記憶装置でのデータの存在確認が単純に行えたと共に、オペレータにとっても、画像処理フロー設定ファイルの記述において、意味のないテープ番号でテープおよびデータを指定する必要がなくなる。

また、テープ番号によってテープを指定させる方法でデータを読み込むことを考え

ると、論理的なデータの固まりが複数のテープに記録されている場合、オペレータは、そのことを意識して、設定を行う必要がある。さらに、テープヘータを書き込む場合でも、テープの容量と、書き込みたいデータ量を考慮して、テープの使用本数を決めるなどの配慮が必要となる。

しかし、図 42に示すような論理パス名でデータの読み込み、書き込みを行えるようにすることで、物理的にどのテープから読み込むか、どのテープに書き込むかといったことは、システムが判断することになる。これによって、オペレータは、読み込みを指示したデータが複数本のテープに記録されているかどうかや、記録したいデータが1本のテープに入るかどうかなどといったことに気を使う必要がなくなる。

また、テープライブラリの導入によって、テープの入替えが可能となるが、それはあくまでもライブラリ内にある有限個のテープに対してである。そのため、これを有効に活用するためには、必要なデータを記録したテープが確実にテープライブラリ内に存在するようにしなければならない。そこで、本システムでは、これまでに作成した全てのテープに対して、その中に書き込まれたデータの内容を管理し、これから実行しようとするジョブに必要なテープがテープライブラリ内に存在するかどうかを判断する。そして、もし、必要なテープがテープライブラリ内に存在しないならば、オペレータにテープの番号を表示して、必要なテープをテープライブラリに入れるように指示することにする。

上記の動作を実現するために、本システムでは、以下に示す3つの管理テーブルを使用する。

テープ管理テーブル（表 10）は、論理パス名と、それに対応するデータが記録されている物理的なテープおよび、その記録位置を管理するテーブルである。このテーブルには、本システムで作成した全てのテープの記録内容が管理される。

表 10 テープ管理テーブル

No.	項目	概要
1	論理パス名	テープに記録されているデータの論理パス名
2	テープ番号	1つ1つのテープを識別するためのシリアル番号
3	テープポジション	テープに記録されているデータのテープ上の位置(書き込み、読み込み開始位置)
4	データサイズ	テープに記録されているデータのデータ量

テープライブラリ管理テーブル（表 11）は、テープライブラリ内に格納されているテープとその種別を管理するテーブルである。

表 11 テープライブラリ管理テーブル

No.	項目	概要
1	テープライブラリ位置番号	テープライブラリ内でテープの格納位置を特定するためのシリアル番号
2	テープシリアル番号	1つ1つのテープを識別するためのシリアル番号(表 10 No.2と同じ)
3	テープ種別	テープの種類

次テープ管理テーブル(表 12)は、論理的なデータの固まりを複数のテープを使用して保存する場合に使用するもので、あるテープの続きのデータがどのテープに記録されているかを管理するテーブルである。

表 12 次テープ管理テーブル

No.	項目	概要
1	テープ番号	1つ1つのテープを識別するためのシリアル番号(表 10 No.2と同じ)
2	次テープ番号	上記テープ番号のテープに記録されたデータの続きが記録されたテープ番号

以下、テープへのデータ書き込み、テープからのデータ読み込み、テープライブラリ管理テーブルの更新、オペレータに対するテープの挿入指示について、簡単な処理の流れを述べる。

5.4.1 テープへのデータ書き込み

テープにデータを格納する場合、まず、与えられた論理パスを解析し、既に存在する論理パスへの追加なのか、論理パスを新規で作成するのかをテープ管理テーブルを用いて判断する。既に存在する論理パスへの追加である場合、テープ管理テーブルから該当するテープのテープ番号が取得できる。さらにテープライブラリ管理テーブルを用いてテープ番号からテープライブラリ位置番号を取得する。論理パスを新規で作成する場合には、テープ管理テーブルからテープ番号を取得することができないため、テープライブラリ管理テーブルから空テープのテープライブラリ位置番号を取得する。このテープライブラリ位置番号を用いて、書き込みを行うテープをテープドライブに挿入する。

テープドライブに挿入されたテープに対してデータを書き込む場合、書き込んだデータの論理パス名、テープ番号、テープポジション、およびデータサイズをテープ管理テーブルに追加する(図 46)。

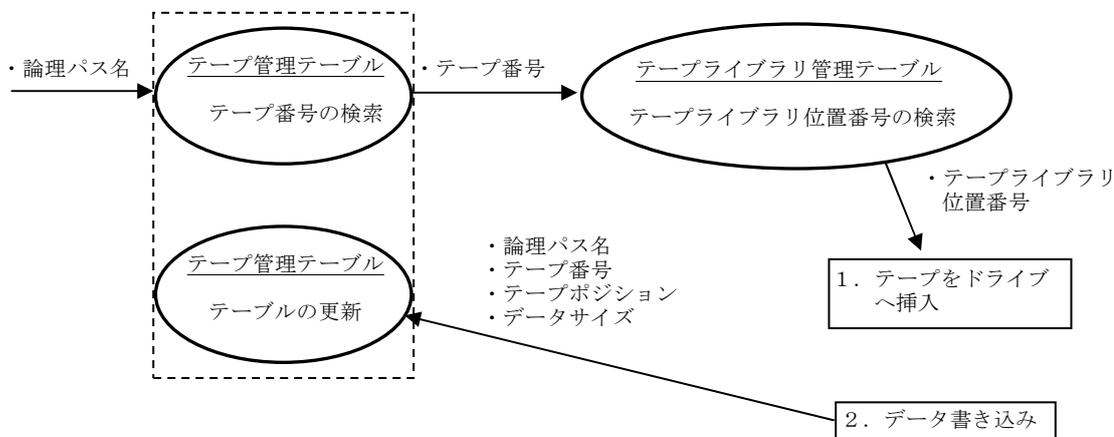


図 46 データ書き込み時の処理の流れ

また、データ書き込み時にテープエンドを検出した場合には、続きのデータを書き込むための空テープをテープライブラリ管理テーブルから検索する。この検索により、空テープのテープ番号と、そのテープが格納されているテープライブラリ内の位置番号が取得できる。取得したテープ番号を、次テープ管理テーブルに書き込むと共に、テープライブラリ内の位置番号を用いて、その空テープをテープドライブに挿入し、書き込みを続行する（図 47）。

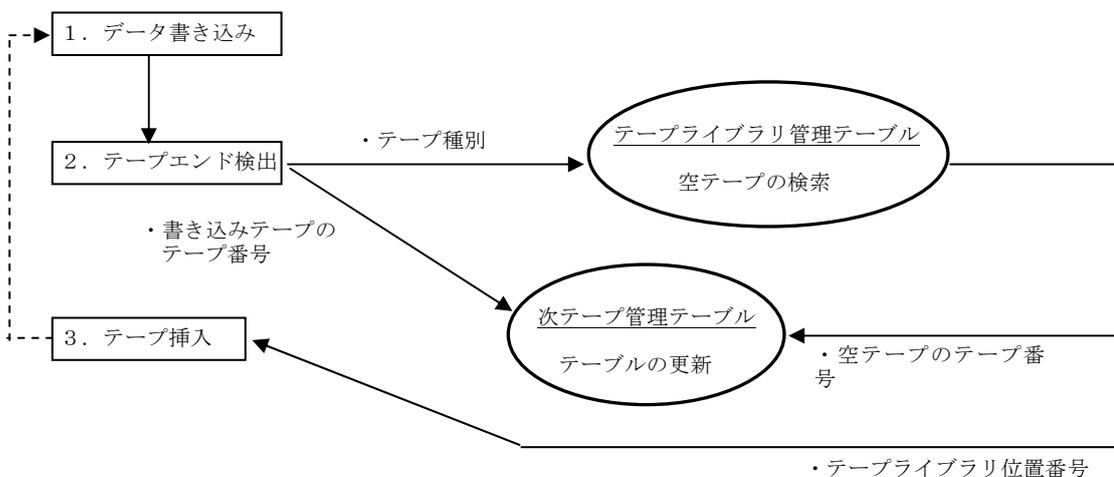


図 47 データ書き込み中にテープエンドを検出した場合の処理の流れ

5.4.2 テープからのデータ読み込み

テープからデータを読み込む場合、与えられた論理パス名をテープ管理テーブルによりテープ番号およびテープポジション(読み込み開始位置)に変換する。変換したテープ番号からテープライブラリ管理テーブルを用いて、テープライブラリ位置番号を取得し、そのテープをテープドライブに挿入する。また、変換したテープポジションを用いて、データ読み込み位置までテープを移動し、データを読み込む(図 48)。

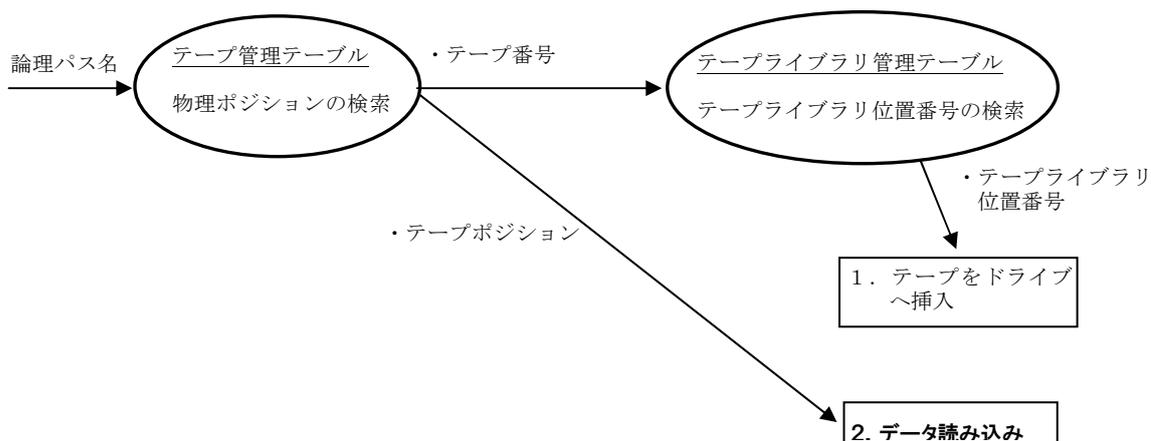


図 48 データ読み込み時の処理の流れ

データ読み込み時にテープエンドを検出した場合、次テープ管理テーブルから、続きのデータが記録されているテープの番号を求め、テープライブラリ管理テーブルを用いて、テープ番号からテープライブラリ位置番号を求める。このテープライブラリ位置番号を用いて、該当テープをテープドライブに挿入し、読み込みを続行する(図 49)。

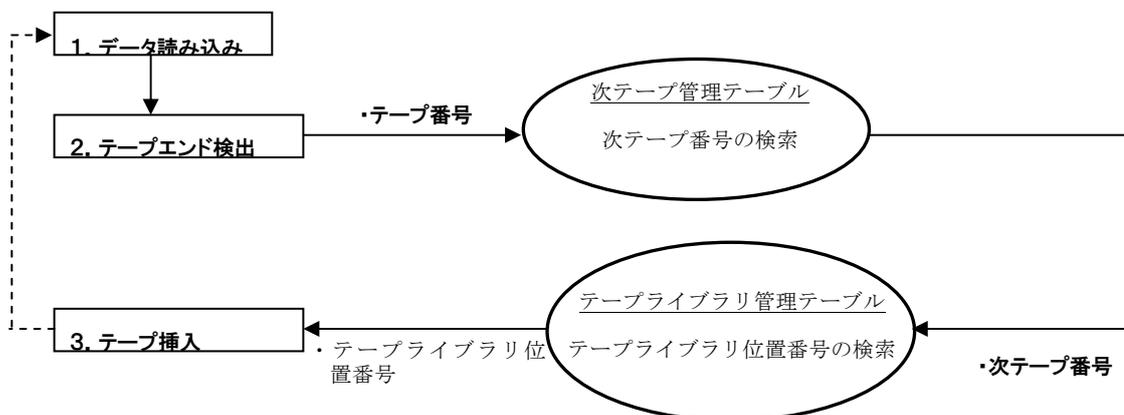


図 49 データ読み込み中にテープエンドを検出した場合の処理の流れ

5.4.3 テープライブラリ管理テーブルの更新

テープ管理テーブル、次テープ管理テーブルの更新は、5.4.1 で示したように、テープへのデータ書き込み時に行われる。しかし、テープライブラリ管理テーブルの更新は、テープライブラリから発行されるイベントを検出して行う。

テープライブラリの起動時とドア開閉時に、テープライブラリはテープライブラリ位置番号とテープ番号を知らせるイベントを発行する。本システムでは、この情報を元にテープ種別を求めて、テープライブラリ管理テーブルの更新を行う。

システムで管理するテープ種別を図 50に示す。テープはまず、管理テープと非管理テープに分かれる。前述したように、本システムでは、テープにシリアルな番号を割当て、それをバーコードとして貼り付けることで1つ1つのテープを識別する。さらに、テープへのデータ書き込み時にテープ管理テーブルの更新を行い、その管理テーブルを用いて論理パス名でのテープへのアクセスを実現する。

しかし、映画会社や映像制作会社から供給されるテープや、顧客に納入するテープにはテープ番号がなく、テープ管理テーブルにも管理情報が登録されていない。このため、このようなテープ番号のないテープを非管理テープとして、論理パス名での読み込み、書き込みを行えないようにする。ただし、オペレータが物理的なテープやテープライブラリの格納位置を意識して、対話形式で読み込みや書き込みを行うことは可能とし、対話形式で行う手順をコマンドとして表現してバッチファイルを記述することで読み込み、書き込みを制御する手段を設ける。

また、テープ番号が付けられた管理テープには、空テープと非空テープの識別を行う。空テープとは、データを書き込んでいないテープまたは、過去に書き込んだが、記録したデータが不要となったテープである。非空テープとは、必要なデータが記録されたテープのことを意味する。

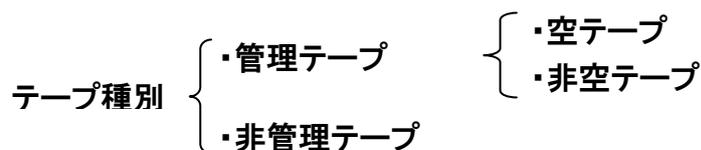


図 50 テープ種別

テープライブラリに非管理テープが入れられた場合、テープライブラリからは、テープライブラリ位置番号だけが知らされる。この場合には、テープ種別を非管理テープとしてテープライブラリ管理テーブルの更新を行う。

また、テープライブラリに管理テープが入れられた場合、テープライブラリからは、テープライブラリ位置番号とテープ番号が知らされる。この場合、そのテープ番号がテープ管理テーブル内にあるかどうかを調べ、ない場合には、テープ種別を空テープ

としてテープライブラリ管理テーブルを更新し、ある場合には、テープ種別を非空テープとしてテープライブラリ管理テーブルを更新する。

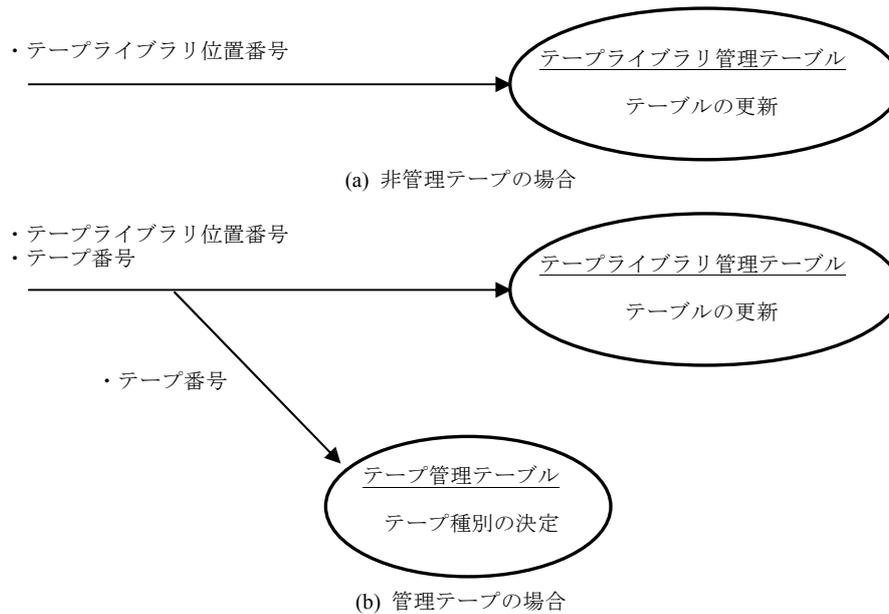


図 51 テープライブラリ管理テーブルの更新手順

5.4.4 オペレータに対するテープの挿入指示

本システムでは、これから実行しようとするジョブに必要なテープがテープライブラリ内に存在しない場合、テープ番号を表示して、オペレータにテープの挿入を指示する。これは、ジョブから入力画像データの論理パス名を取得し、その論理パス名をテープ管理テーブルによってテープ番号に変換する。そして、そのテープ番号を持つテープがテープライブラリ内に存在するかどうかをテープライブラリ管理テーブルで確認し、存在しない場合、そのテープ番号をモニタなどに表示する（図 52）。

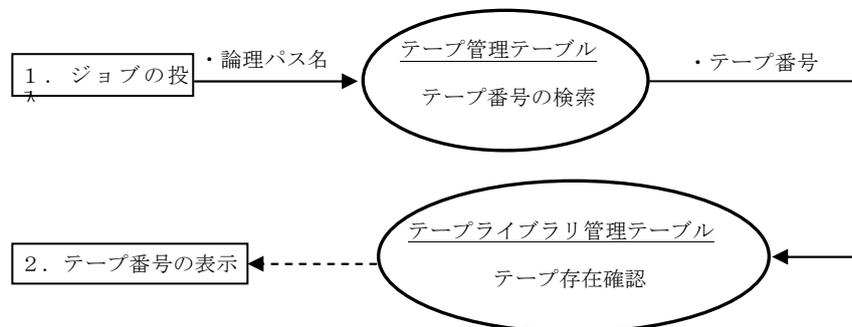


図 52 オペレータに対するテープの挿入指示の処理の流れ

なお、ジョブ投入時には、5.2 で述べたように、ディスクアレイの空き容量確保のために、テープに移動するデータをオペレータに指示させる場合がある。このため、テ

テープライブラリ内の空テープが不足した場合、必要な入力画像データが記録されたテープ以外に空テープの挿入を指示することがある。

また、システムは、投入されている全てのジョブに必要な、入力画像データ用テープの存在を確認するとともに、オペレータによってテープへの移動を許可されたデータの総データ量から必要な空テープ数を算出する。そして、テープライブラリ管理テーブルを用いて、投入されているジョブの実行に必要なないテープのテープライブラリ位置を求め、これを表示する。

このように、挿入を必要とするテープと、挿入可能なライブラリ上の位置を同時にオペレータに示すことで、テープライブラリでのテープの入替え作業がスムーズに行えると考えられる。

5-1-1-3 まとめと今後の課題

高解像度・大容量のコンテンツを効率的に作成するために、ネットワークに接続された複数のPCにレンダリング処理および動画圧縮処理を分散・並列して実行する方法について検討を行った。その結果、システム内のレンダリングPCにレンダリング処理と動画圧縮処理の両方を行わせる方法と、レンダリングPCと制御PCにレンダリング処理と動画圧縮処理のどちらかを行わせる方法を、レンダリング処理と動画圧縮処理の処理時間、使用するPC台数などを考慮したシミュレーションを行うことにより決定することにした。

また、ネットワーク接続された複数の計算機に対する映像データの供給や、映像素材や加工後の映像データが置かれる記憶媒体への画像データの入れ替え作業の効率化を図るために、実運用を考慮に入れた超大容量データマネージメント技術の検討を行った。それにより、システム内に存在する大容量記憶装置間のデータの流れと、各記憶装置におけるデータの管理方法を決定した。

今後、本検討内容に基づいて開発を行い、本システムの検証を行う。