

平成15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	8
3-3	研究開発の年度別計画	11
3-4	研究開発体制	12
4	研究開発の概要（平成15年度まで）	15
4-1	研究開発実施計画	15
4-1-1	研究開発の計画内容	15
4-1-2	研究開発課題実施計画	22
4-2	研究開発の実施内容	23
5	研究開発実施状況（平成15年度）	34
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	34
5-1-1	はじめに	34
5-1-2	実施内容	34
5-1-3	おわりに	129
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	130
5-2-1	はじめに	130
5-2-2	実施内容	131
5-2-3	おわりに	194
5-3	総括	197

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(Asymmetric Digital Subscriber Line)、FTTH(Fiber To The Home) 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(Video On Demand) 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、IPD 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)に関しては未だ発展途上の技術が多く含まれ、様々なアプリケーションを対象にいくつかの研究が続けられている。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の表示技術、防衛や医療等におけるシミュレーション技術、科学技術計算の可視化技術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に必要な高精細・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取得できたとしても、それをシナリオに合うコンテンツとして編集を行い、IPD表示環境に投影する形式に変換することができる汎用的な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いので、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コンテンツを制作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。しかしこれについても、大規模な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が、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広帯域を必要とする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能とするための、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

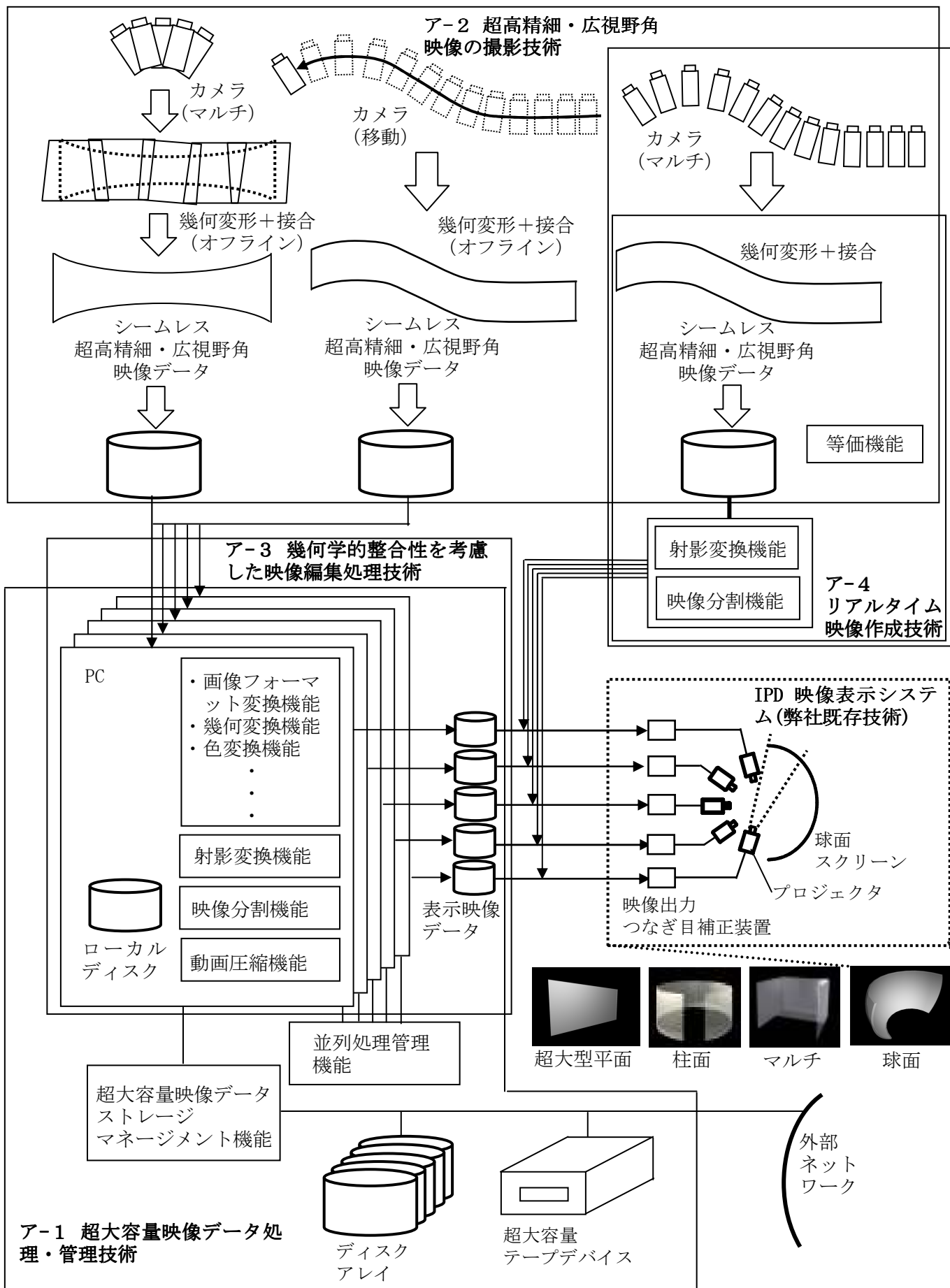
教育、医療、セキュリティ、サービス、文化・芸術、など様々な分野でのアプリケーションが期待できる、高臨場感ディスプレイ(IPD: Immersive Projection Display)用デジタル映像コンテンツに関し、その制作、実施、及び流通を効率的・高機能に行う技術の確立のため、高付加価値・大容量データの処理技術、及びこれらのデータをネットワークで結ばれた遠隔地間で効果的に処理し、円滑に実施・流通させる超広帯域・超高品質ネットワーク制御に関する技術の研究開発を行う。

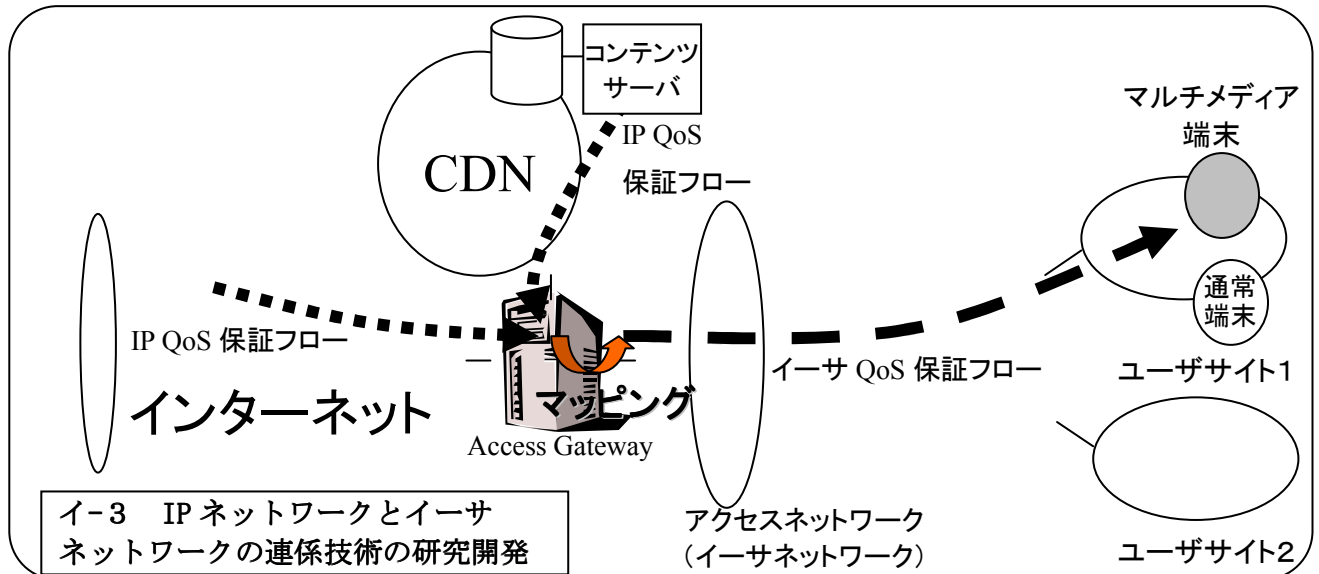
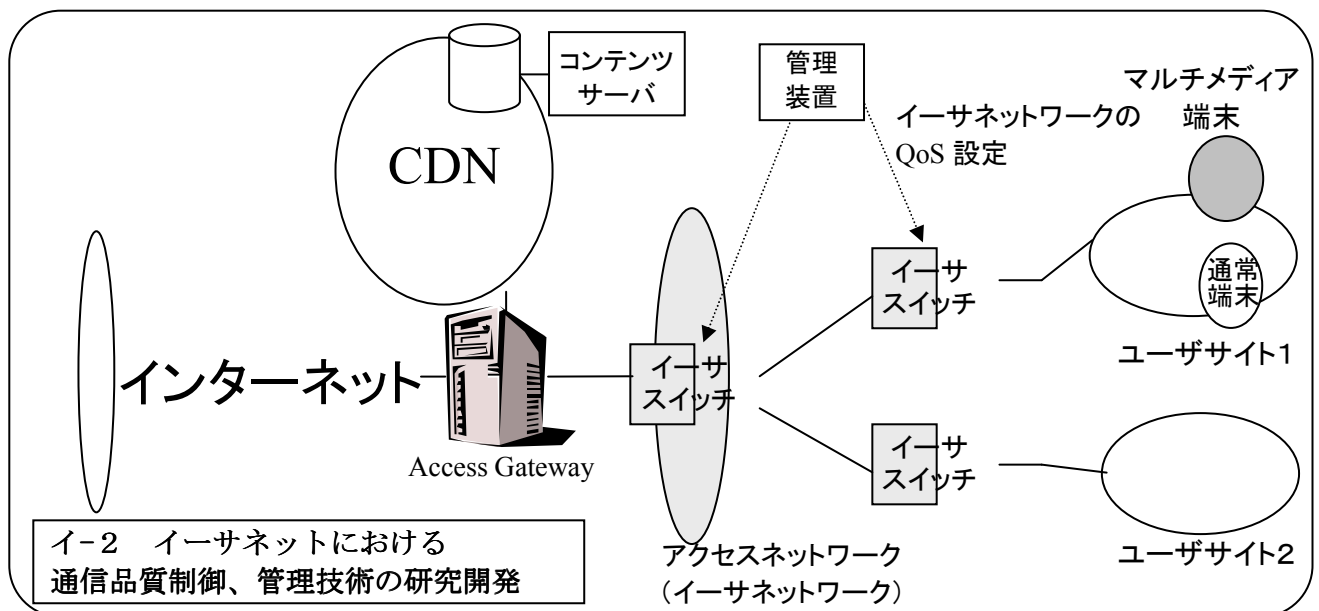
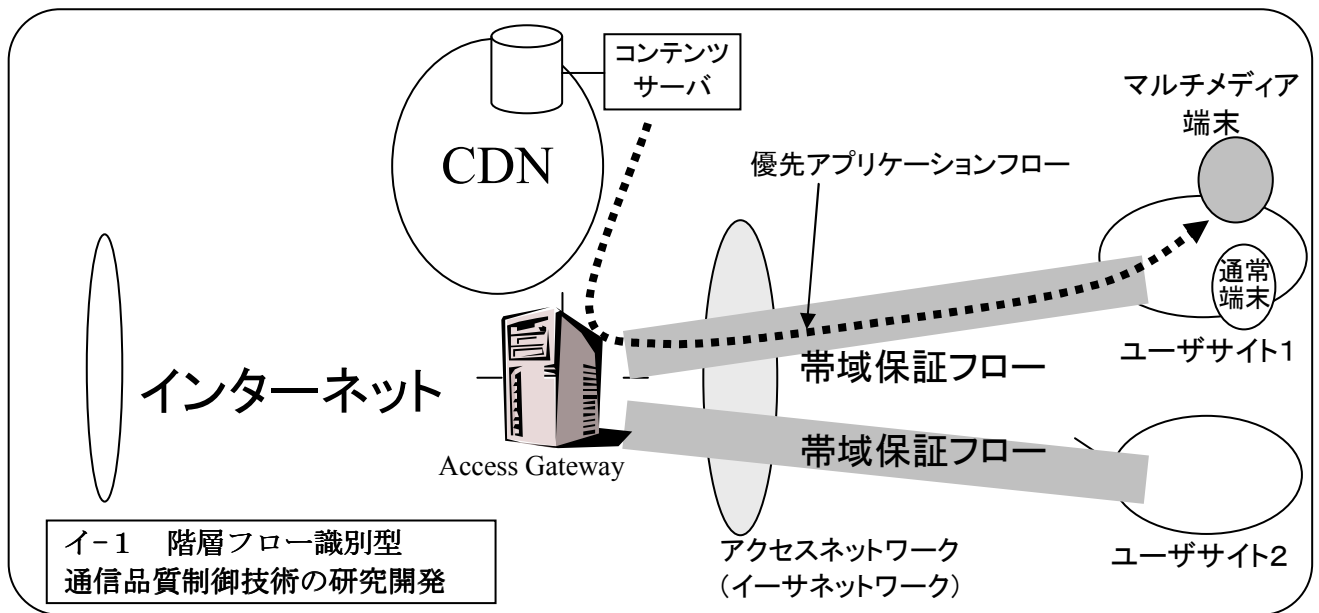
IPD 用映像制作では、IPD が持つ大画面、非平面、超高精細・広視野角などといった特徴のため、テレビや従来の劇場映画等を対象とした映像制作過程では有り得なかった新たな技術課題が生じており、撮影から編集、配信、表示に至るまでのトータルなソリューションが求められる。超高精細映像の撮影機能、映像の幾何学的整合性を考慮した編集機能、映画の10倍以上にも及ぶ超大容量データの処理方法等、諸課題を克服してはじめて臨場感あふれる新しい価値を持った映像空間の提供が可能になる。また、このような高付加価値で且つ大容量のデータを、ネットワークで結ばれた遠隔地間で効率的に処理し、スムーズに流通させるためには、広帯域な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

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。

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

(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形式の変換処理、プロジェクト分割処理、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けるPCの台数は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、下記の技術を確立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。

(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

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立する。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立する。

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併せて検討を加え、設計に取り入れるものとする。

以上の技術を確立し、実機によるプロトシステムを開発し、評価を行う。

ア-2 超高精細・広視野角映像の撮影技術

(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。

(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認する。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。

ア-3 幾何学的整合性を考慮した映像編集処理技術

(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。

(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係させ、評価を行う。

(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)、(2)で開発した技術・プロトシステムやCGを用い、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。

ア-4 リアルタイム映像作成技術

(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。

(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。

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

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

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

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

3-3 研究開発の年度別計画

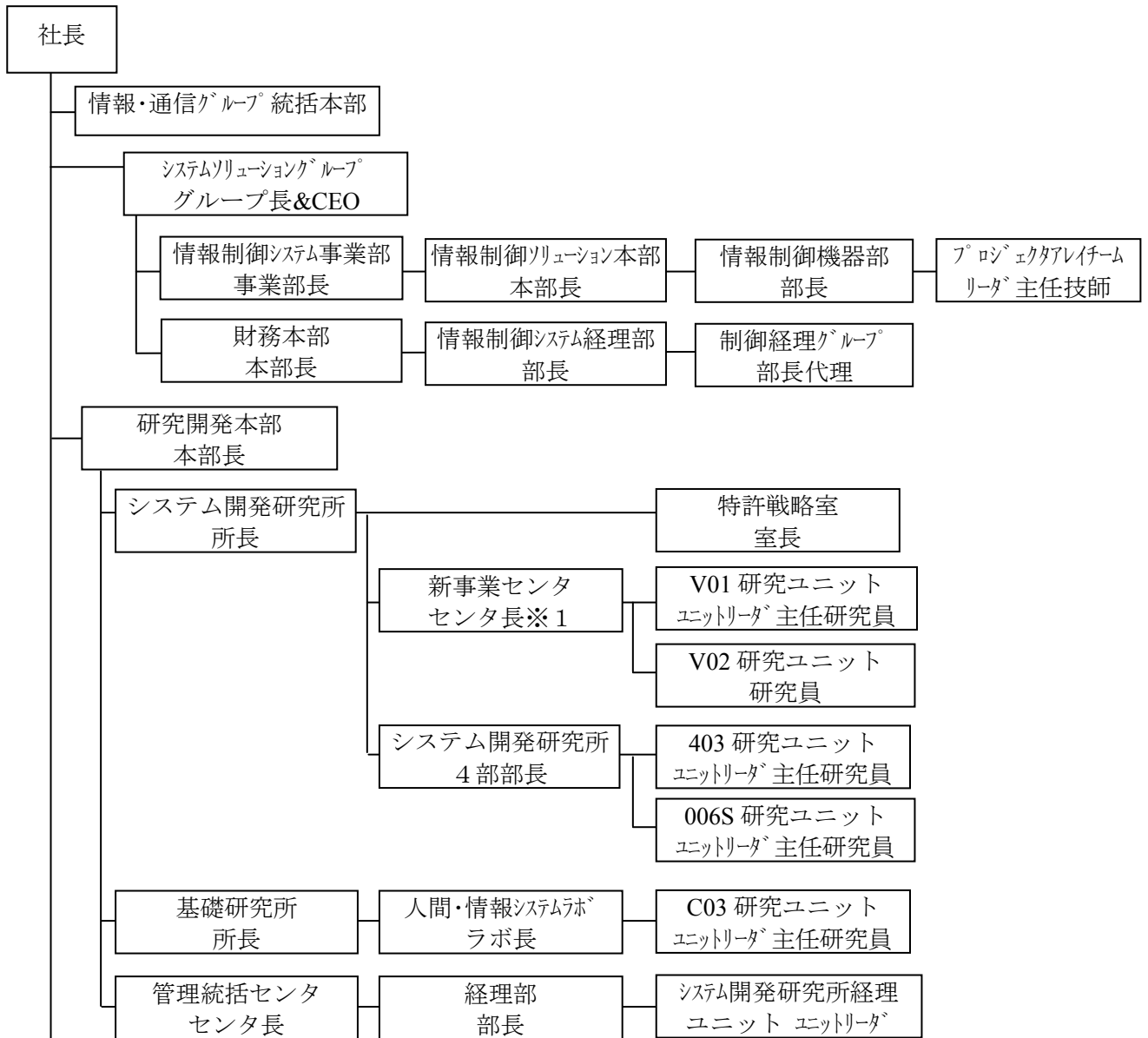
(金額は非公表)

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			→		
イ-2 Ethernet における QoS 制御、管理技術			→		
イ-3 IP ルータと Ether ネットワークの連携技術			→		(株) 日立コミュニケーションテクノロジー
間接経費					
合計					

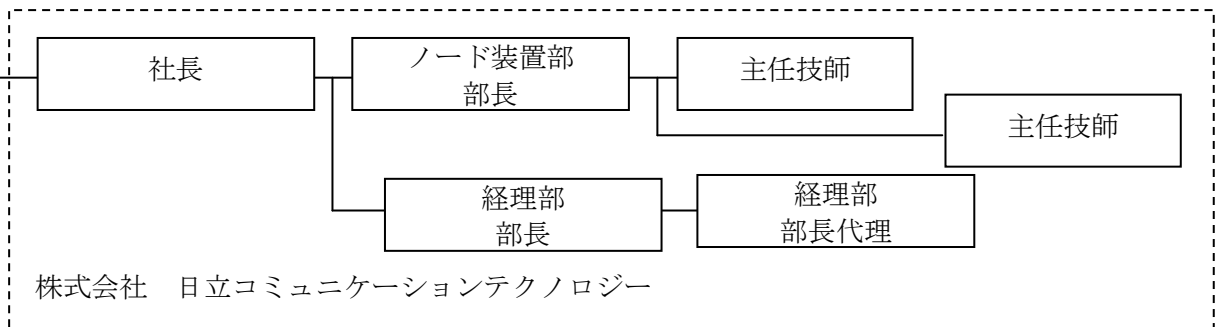
- 注) 1 経費は研究開発項目毎に消費税を含めた額で計上。また、間接経費は直接経費の30%を上限として計上(消費税を含む)。
 2 備考欄に再委託先機関名を記載

3-4 研究開発体制

(1) 研究開発管理体制

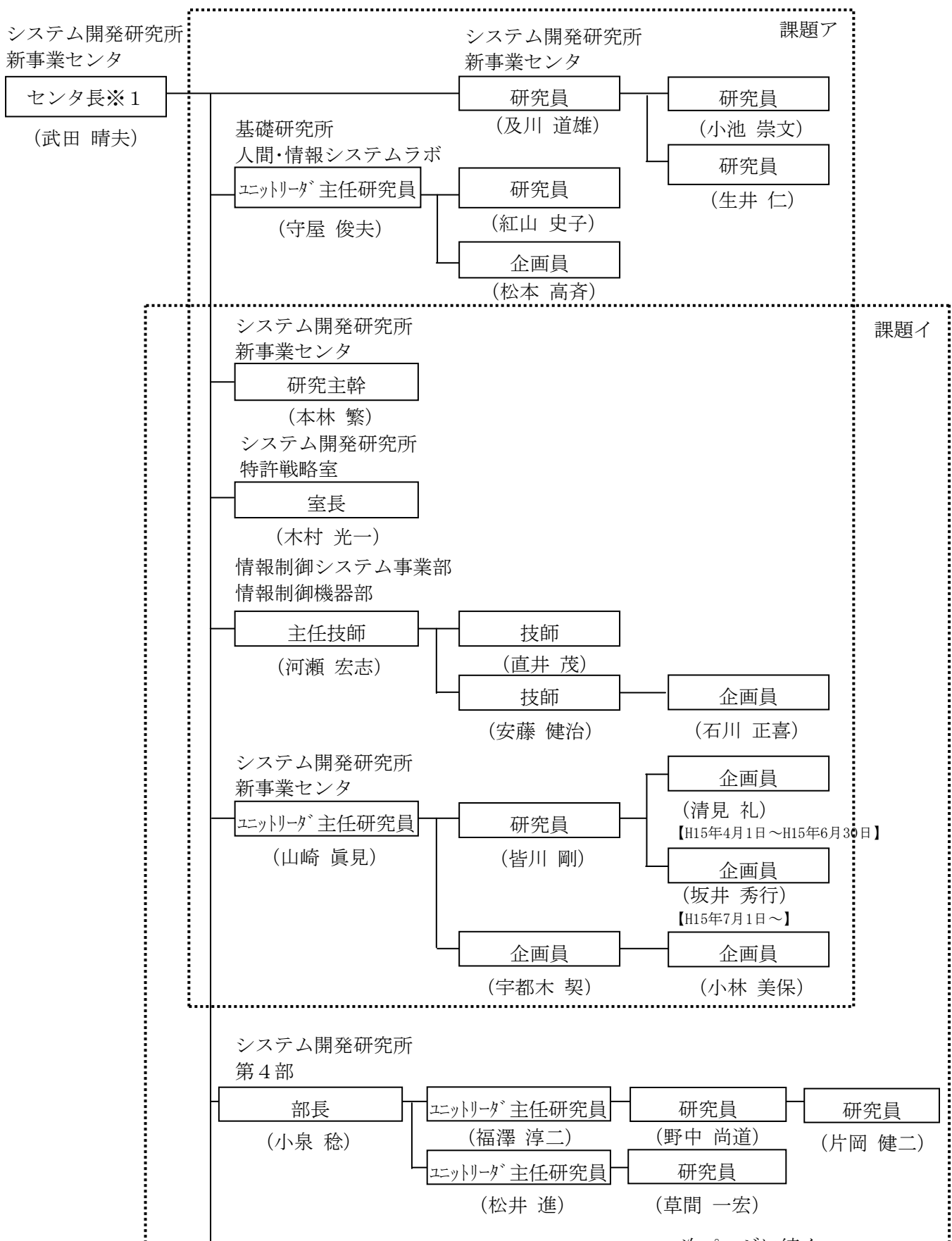


※1：本研究開発の全体取り纏め責任者



株式会社 日立コミュニケーションテクノロジー

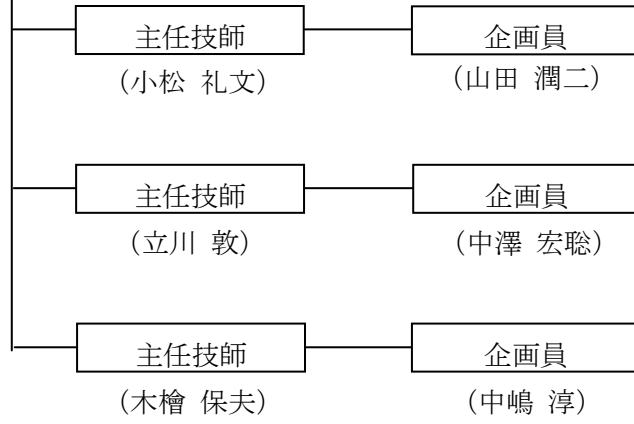
(2) 研究開発実施体制



一次ページに続くー

—前ページより続く—

【再委託先】(株)日立コミュニケーションテクノロジー
ノード装置部



以上課題イ

4. 研究開発の概要（平成15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

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。

平成15年度は、平成16年3月末における最終目標に向けて、各課題項目を実装するプロトシステムの開発、検証を主に行う。

各項目の詳細は、以下のとおりである。

ア-1 超大容量映像データ処理・管理技術

(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・制作に必要となる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

平成15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映像編集システムプロトタイプを開発する。その実現においては、平成14年度に検討・設計を行った下記技術を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回避する最適なスケジュール計画
- 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表示技術

(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となる、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。

平成15年度は、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仕組みを確立する。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を確立する。さらに、同アーキテクチャにおいて、映像データが失われるなどの事故が生じた場合、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、(1)記載の映像編集システムプロトタイプに組み入れる。

ア-2 超高精細・広視野角映像の撮影技術

(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。

平成15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射影変換等の画像処理を行うことでお互いをシームレスに繋ぎ合わせ、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技術を確立する。具体的には、平成14年度に設計した仕様に基づきカメラシステムハードウェアのプロトタイプの開発を行い、そのシステムで撮影される映像データを繋ぎ目のない画像に変換するための、画像処理技術を確立する。

(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。

平成15年度は、移動や回転を行うカメラによって撮影された連続する複数の映像を、射影変換等の画像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、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、平成14年度に検討を行った基本アルゴリズムに基づき、上記の撮影を行うための自律撮影ロボット制御技術の開発、およびそれによって撮影された映像の画像処理を行う実験システムの開発を行い、実証実験を行う。

ア-3 幾何学的整合性を考慮した映像編集処理技術

(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 課題解決に必要となる既存技術の調査

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。

平成15年度は、映像制作・編集過程におけるIPD固有の処理技術を確立するため、平成14年度の研究にて開発した、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技術について、課題ア-1記載の映像編集システムプロトタイプに実装する。

(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。

平成15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ついて、平成14年度に開発し

た手法を課題ア-1記載の映像編集システムプロトタイプに実装する。

(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-1～3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。

平成15年度は、課題ア-1～3の検証を目的とし、IPD表示環境に表示する実験コンテンツを、CGや課題ア-1記載の映像編集システムプロトタイプを利用して制作する。

ア-4 リアルタイム映像作成技術

(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。

平成15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、各カメラの位置や方向、レンズ歪みなどを考慮にいたった画像変換パラメータを算出するためのキャリブレーション技術、ならびに、映像データをIPD映像表示システムにおけるプロジェクト固有のパラメータに基づく映像に変換する映像変換技術を確立し、下記プロトタイプに実装する。

(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。

平成15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に、複数台のカメラにより撮影した映像を、遠隔にある複数プロジェクトを用いシームレスかつリアルタイムに表示可能とする実験アプリケーションの開発を行い、デモンストレーションを実施する。なお、実験アプリケーションシステムの通信路には課題イで開発した機能を取り入れ、通信品質を制御するものとする。これにより、課題アとイを融合したシステムの実用化に際しての課題抽出も行う。

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。

平成15年度は、平成16年3月末における最終目標に向けて、各課題項目を実装するプロトシステムの開発、検証を行う。

イ-1 階層フロー識別型通信品質制御技術の研究開発

(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) 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フィックと、該映像データ放送の複数観客からの反応情報返送トラフィックが混在する場合の、各トラフィ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

平成15年度は、上記（1）、（2）の平成14年度までの開発項目に基づき、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

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

平成15年度は、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

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

平成15年度は、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

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、管理技術の研究開発 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

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。

平成15年度は、平成16年3月末における最終目標に向けて、各課題項目を実装するプロトシステムの開発、検証を主に行った。

各項目の詳細は、以下のとおりである。

ア-1 超大容量映像データ処理・管理技術

(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

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。

平成 15 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映像編集システムプロトタイプを開発した。その実現においては、平成 14 年度に検討・設計を行った下記技術を取り入れた。

- ① 処理の効率的なスケジューリング管理
- 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回避する最適なスケジューリング計画
- 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 表示技術

(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。

平成15年度は、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仕組みを確立した。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 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を確立した。さらに、同アーキテクチャにおいて、映像データが失われるなどの事故が生じた場合、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、設計に取り入れた。

以上の技術を、(1) 記載の映像編集システムプロトタイプに組み入れた。

ア-2 超高精細・広視野角映像の撮影技術

(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。

平成15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射影変換等の画像処理を行うことでお互いをシームレスに繋ぎ合わせ、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技術を確立した。具体的には、平成14年度に設計した仕様に基づきカメラシステムハードウェアのプロトタイプの開発を行い、そのシステムで撮影される映像データを繋ぎ目のない画像に変換するための、画像処理技術を確立した。

(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)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

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。

平成 15 年度は、移動や回転を行うカメラによって撮影された連続する複数の映像を、射影変換等の画像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、超高精細な静止画像を取得する技術を確立した。具体的には、平成 14 年度に検討を行った基本アルゴリズムに基づき、上記の撮影を行うための自律撮影ロボット制御技術の開発、およびそれによって撮影された映像の画像処理を行う実験システムの開発を行い、実証実験を行った。

ア-3 幾何学的整合性を考慮した映像編集処理技術

(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。

平成15年度は、映像制作・編集過程におけるIPD固有の処理技術を確立するため、平成14年度の研究にて開発した、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技術について、課題ア-1記載の映像編集システムプロトタイプに実装した。

(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。

平成15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ついて、平成14年度に開発した手法を課題ア-1記載の映像編集システムプロトタイプに実装した。

(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-1～3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。

平成15年度は、課題ア-1～3の検証を目的とし、IPD表示環境に表示する実験コンテンツを、CGや課題ア-1記載の映像編集システムプロトタイプを利用して制作した。

ア-4 リアルタイム映像作成技術

(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。

平成15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、各カメラの位置や方向、レンズ歪みなどを考慮にいたった画像変換パラメータを算出するためのキャリブレーション技術、ならびに、映像データをIPD映像表示システムにおけるプロジェクタ固有のパラメータに基づく映像に変換する映像変換技術を確立し、下記プロトタイプに実装した。

(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)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。

平成15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に、複数台のカメラにより撮影した映像を、遠隔にある複数プロジェクタを用いシームレスかつリアルタイムに表示可能とする実験アプリケーションの開発を行い、デモンストレーションを実施した。なお、実験アプリケーションシステムの通信路には課題イで開発した機能を取り入れ、通信品質の制御を実施し、課題アとイを融合したシステムの実用化に際しての課題抽出を行った。

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。

平成15年度は、平成16年3月末における最終目標に向けて、各課題項目を実装するプロトシステムの開発、検証を行った。各項目の詳細は、以下のとおりである。

イ-1 階層フロー識別型通信品質制御技術の研究開発

(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) 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 (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。

平成15年度は、上記(1)、(2)の平成14年度までの開発項目に基づき、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した。

- 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

平成 15 年度は、平成 14 年度に開発した L2Shaper プロトタイプの改良を行い、上記①から④を実現した。①では、プロトタイプ上にレイヤ 2、レイヤ 3、レイヤ 4 の 3 階層のフロー識別機能を実装し、動作の検証を行った。②では、プロトタイプ上に優先制御および帯域制御機能を実装し、動作の検証を行った。③では、プロトタイプ上で 1000 フロー以上の通信品質制御の設定が可能であることを確認した。④では、プロトタイプ上で品質制御に関する性能評価を行い、100 Mbps 以上のフローを扱えることを確認した。

また、平成 14 年度に方式検討を行ったアプリケーション対応通信品質制御技術の更なる検討を行い、検証用実験コンテンツを開発して、アプリケーション対応通信品質制御技術の効果を検証した。

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。

平成15年度は、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した。

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

平成15年度は、平成14年度に開発したL2管理サーバプロトタイプの改良を行い、上記①から③を実現した。①では、平成14年度に開発したL2Shaperプロトタイプと、Extreme社製イーサスイッチとを対象とした、機種依存ドライバを改良し、動作の検証を行った。②では、①のプロトタイプソフトウェアを、実際にL2Shaper及びExtreme社製イーサスイッチと組み合わせて動作させ、1フローの設定が5秒以内に完了することを確認した。③では、プロトタイプ上で1000フローの通信品質制御の設定が可能であることを確認した。

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、ユーザー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。

平成15年度は、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において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した。

- 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

①では、アクセスゲートウェイのプロトタイプ上に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機能を実装し、両者の連動が実現されていることを確認した。②では、アクセスゲートウェイのプロトタイプ上で、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を検証した。

5. 研究開発実施状況（平成15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 はじめに

IPD (Immersive Projection Display: 高臨場感ディスプレイ) を対象にしたデジタル映像制作では、IPD が持つ大画面、非平面、超高精細、広視野角といった特殊性のために、従来のテレビや映画、すなわち限られた解像度の平面且つ矩形映像を対象としたコンテンツの制作過程では有りえなかった新たな技術課題が生じている。課題アでは、これらのなかで主要かつ基盤的な技術要素として、超大容量映像データ処理・管理技術、超高精細・広視野角映像の撮影技術、幾何学的整合性を考慮した映像編集処理技術、リアルタイム映像作成技術の4つを挙げ、研究を進めた。

5-1-2 実施内容

課題ア-1 超大容量映像データ処理・管理技術

IPD 用映像制作過程においては、投影する映像が高精細・広視野角となるため、扱う映像データとその処理量は膨大なものとなる。

本課題では、映像素材の加工時間（編集処理時間）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研究を行った。また、これらの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、超大容量映像データのマネージメント技術について研究を行った。

（1）画像処理並列化技術

1 コンテンツ編集作業

IPD 用コンテンツ編集作業は、切り貼り編集に代表されるノンリニア編集作業、ならびに IPD 映像表示環境に映像を出力するために行う映像変換（フォーマット変換）作業の2つの作業から構成される。本項ではこの中で主に前者を扱う。ノンリニア編集作業は、ビデオや映画の制作でも行なわれるが、IPD を対象とした映像編集では、投影するスクリーン形状が特殊となるため、通常の透視投影映像だけでなく、より広角の映像が表現できる柱面投影映像や魚眼投影映像などに対応できる必要がある。

図1にコンテンツ編集作業の流れを示す。

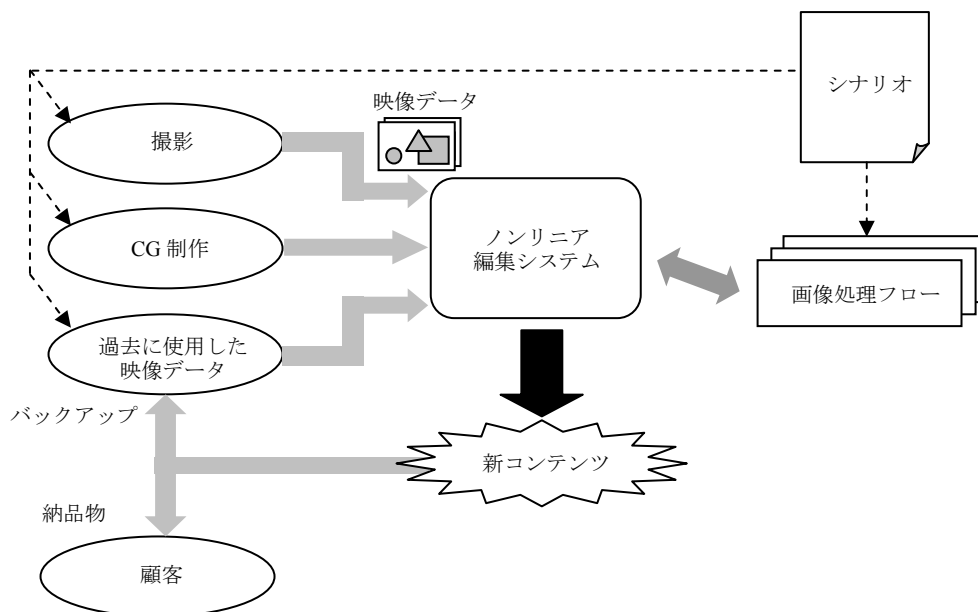


図1 コンテンツ編集作業の流れ

この図に示されるように、準備された映像データを組み合わせ、指定されたシナリオどおりのコンテンツに仕上げるために、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。

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ある。その内容は、入力映像データの指定、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。

この画像処理フローに従い、出力画像を作成することをレンダリングと呼ぶ。よって、ノンリニア編集システムにおける映像データ加工過程は、この画像処理フローを作成編集する過程と、画像処理フローに従いレンダリングを行う過程の2つに分けることができる。

なお、画像処理フローの内容は、一般にファイルに保存することができるので、これを画像処理フロー設定ファイルと呼ぶことにする。

オペレータが画像処理フローを作成編集する際には出力画像を作成し、その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必要がある。このように確認を目的としたレンダリングを、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 システム構成

本システムの構成を図2に示す。合計21台のパーソナルコンピュータ(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)が主に画像処理を行う。制御PCは1.4GHzのCPU2個、256MByteのメモリ、36.8GByteの内蔵ハードディスクを有する。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24MByte/secのテープドライブを搭載し、さらに1巻あたり200.0GByte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、2.0 GHz の CPU1 個、160.0 GByte の内蔵ハードディスクを有する。20 台のレンダリング PC のうち 6 台には 2.0 GByte、残りの 14 台には 512 MByte のメモリがそれぞれ搭載されている。

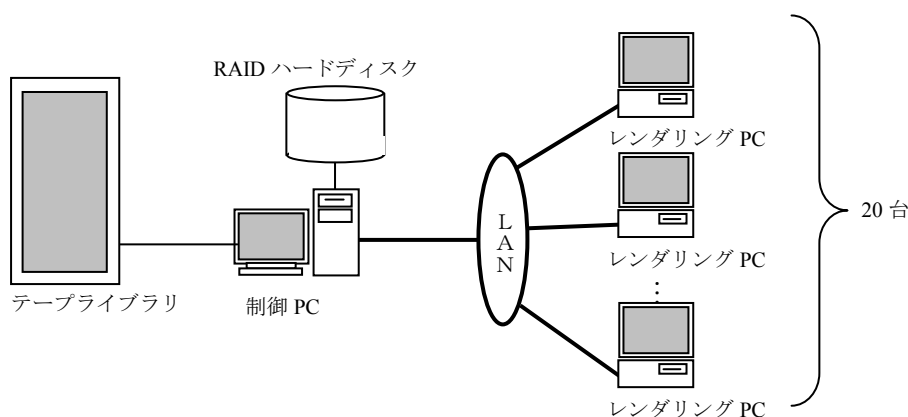


図2 システム構成

ただし、レンダリング PC の台数は固定的なものではなく、追加および削除することが可能である。

3 PC クラスタ

計算機クラスタとは、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種類である。ネットワーク接続された 1 台 1 台の計算機のことをノードと呼ぶ。そして、このノードに PC を使用したものを PC クラスタという。

4 本システムにおけるコンテンツ編集作業の流れ

本システムは、2 つのプログラムから構成される。1 つは画像処理フロー編集プログラムであり、もう 1 つは画像処理フロー実行プログラムである。

画像処理フロー編集プログラムは、画像処理フローの編集を支援するために、画像処理フローをグラフィカルに表示し、画像処理の追加や削除、パラメータの設定などといった操作をしたり、編集中の画像処理フローの簡易レンダリングを実行したりして出力画像を確認することができるプログラムである。画像処理フロー編集プログラムを用いて作成された画像処理フローは、ファイル（画像処理フロー設定ファイル）に出力される。

画像処理フローの本レンダリングは、画像処理フロー実行プログラムで行われる。画像処理フロー設定ファイルを画像処理フロー実行プログラムが読み込み、レンダリングを実行することで入力画像データから出力画像データおよび出力動画像データが作成される。

画像処理フロー編集プログラムは、制御 PC 単体で実行されるアプリケーションであるが、画像処理フロー実行プログラムは、制御 PC と複数台のレンダリング PC が協調して動作するプログラムである。

5 処理の効率的な配分管理

画像処理フロー実行プログラムはレンダリング処理を行う場合、それを大きく 2 つの工程に分けて実行する。以下この 2 つの工程を、前工程および後工程と呼ぶことにする。

前工程は、後工程を実行するための準備作業を行う工程である。主な作業は、テープライブラリおよびテープドライブを制御して、画像処理フローの実行に必要な入力画像デー

データをテープから RAID ハードディスクに読み込むことである。ただし、上記作業を行うにあたり、RAID ハードディスク上に必要な空き領域が無い場合、RAID ハードディスク上に置く必要のないデータをテープに移動したり、あるいは削除したりする。

後工程では、前工程で準備された入力画像データを用いて出力画像および出力動画を生成する。ここで出力画像とは、フレーム毎に 1 枚 1 枚の静止画データとして分けられているものを言う。以下出力画像の生成処理のみをレンダリング処理と呼び、出力動画の生成処理は動画生成処理と呼んで区別することにする。後工程では本システム内の複数台の PC を用いて、これら 2 つの処理を効率的に行う。

そのための方法として、以下に示す 2 つを前回の報告書に示した。

[配分方法 1] 全てのレンダリング PC にレンダリング処理と動画生成処理の両方を行わせる。このとき制御 PC には、レンダリング処理も動画生成処理も行わせない。

[配分方法 2] いくつかのレンダリング PC にはレンダリング処理を行わせ、残りのレンダリング PC には動画生成処理を行わせる。このとき制御 PC には動画生成処理を行わせる。また、レンダリング処理を行わせる PC と動画生成処理を行わせる PC の台数配分は、1 つの画像処理フローの実行中は固定するものとする。

[比較検討]

前回の報告書において、配分方法 1 と配分方法 2 の処理時間を比較するための簡単なシミュレーションを行った結果、ロードバランスを適切に保てるのであれば、配分方法 2 を用いることがよいことを述べた。ここで、ロードバランスを適切に保つとは、レンダリング処理と動画生成処理を行う PC の台数配分を適切に選択することを意味する。

そのため本システムでは、画像処理フローに基づきレンダリングを行う前に、その画像処理フローのフレームの進行に伴うレンダリング処理時間の推移（実測値）と動画生成処理の予測時間、総処理フレーム数、使用可能なレンダリング PC 台数などの情報を用いて、所要時間を求めるための、シミュレーションを行うこととした。

このシミュレーション結果が信頼できる場合には配分方法 2 を採用し、信頼できない場合には配分方法 1 を採用しようと考えた。

以下、表 1 に示す 3 つの画像処理フローを用いて、シミュレーション結果の信頼性について評価する。

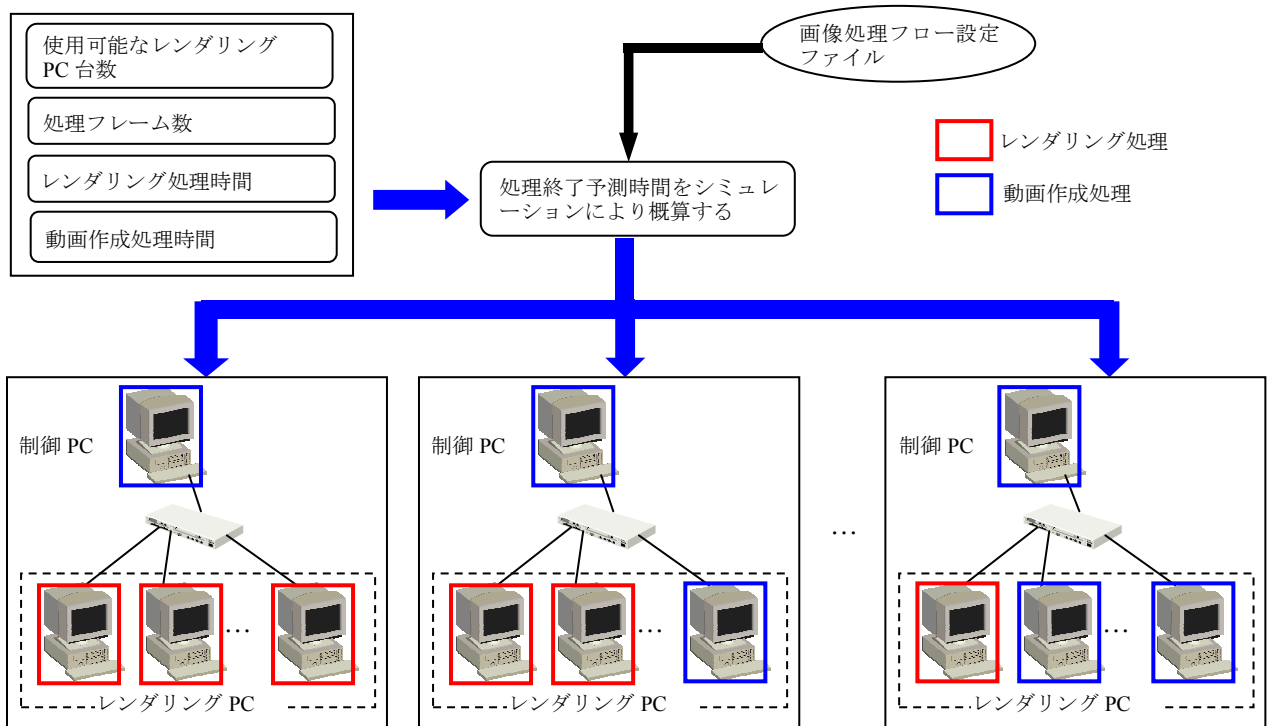


図3 後工程の処理概要（配分方法2）

表1 各画像処理フローのレンダリング処理時間

No.	画像処理フロー名	総フレーム数 [枚]	画像データ送信 処理時間 [sec]	レンダリング 処理時間 [sec]	動画作成 処理時間 [sec]	画像データ受信 処理時間 [sec]
1	画像処理 フローA	100	1.1562	11.9	1.2	0.0656
2	画像処理 フローB	100	1.0406	36.7	7.9	0.372
3	画像処理 フローC	100	1.1470	184.6	7.9	0.3686

図4は、表1に示す3つの画像処理フローを配分方法2で実行し、レンダリング処理と動画作成処理を行うPCの台数配分による所要時間の変化を調べた図である。また、図中で、黄色の点で示されているのは、画像処理フローの実行前に行うシミュレーションで選択された台数配分である。図4から分かるように、配分方法2では、レンダリング処理を行うPCと動画作成処理を行うPCの台数配分によって、処理のロードバランスが変わり、所要時間が変化する。ここで、シミュレーションを元にして選択した台数配分は、いずれの場合にも所要時間最小となるものであり、本システムで行うシミュレーションの信頼性を示す結果となっている。

また、それぞれの画像処理フローを配分方法1で実行した場合の所要時間は、それぞれ124[sec]、317[sec]、1428[sec]であり、シミュレーションで選択した台数配分で配分方法2を実行した場合と比べれば劣るが、著しい差はない結果であった。そのため、レンダリング処理時間の実測値が示されていない画像処理フローが与えられた場合など、シミュレーシ

ョンが正常に行えない場合には、根拠のない台数配分で配分方法 2 を実行するよりも、配分方法 1 で実行したほうが所要時間を少なくできる可能性が高い。

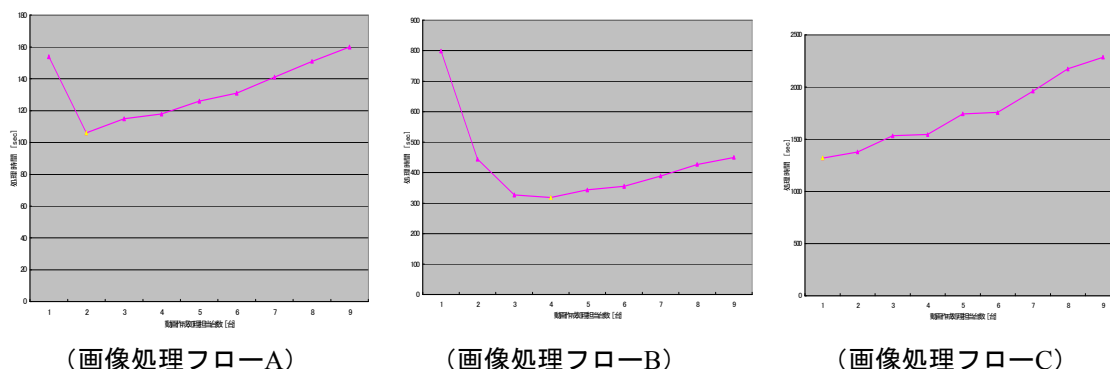


図 4 PC の台数配分による所要時間の変化

そのため、画像処理フロー実行アプリケーションにおいては、配分方法 2 の最適な台数配分を求めるシミュレーションが実行できる場合には、その台数配分で配分方法 2 を実行するが、何らかの原因でシミュレーションが実行できない場合には、配分方法 1 を実行することにする。

6 ボトルネックを解消するディスクアクセス管理

レンダリング PC が複数台あることを考えれば、画像処理フローの実行中は、ディスクアレイのアクセスが頻繁に発生することは容易に想像できる。

図 5 は、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時間を調べた結果である。

各レンダリング PC は、入力画像データが無ければ、レンダリング処理を実行することができない。そのため、入力画像データの転送時間の増加は、レンダリング処理の開始待ち時間を増加させることになる。そのため、このような待ち時間の増加をできるだけ押さえ、より多くのレンダリング PC に入力画像データを供給しなければならない。

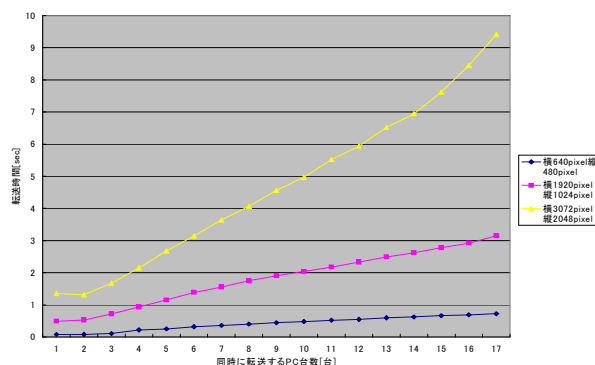


図 5 ディスクアレイからレンダリング PC に対する画像データ送信速度

そこで、同時にディスクアレイにアクセスできる画像データ送受信および動画圧縮データ送受信処理の数（同時アクセス許可数）に制限を設けることにした。

表 1 に示す 3 つの画像処理フローをレンダリング PC の台数を変えて実行し、その時の所要時間が 1 台のレンダリング PC を使用した時の何倍になっているかを調べた(高速化率)。計測は、同時アクセス許可数 1, 3, 6, 制限無しの 4 つについて行った。その結果を図 6 に示す。

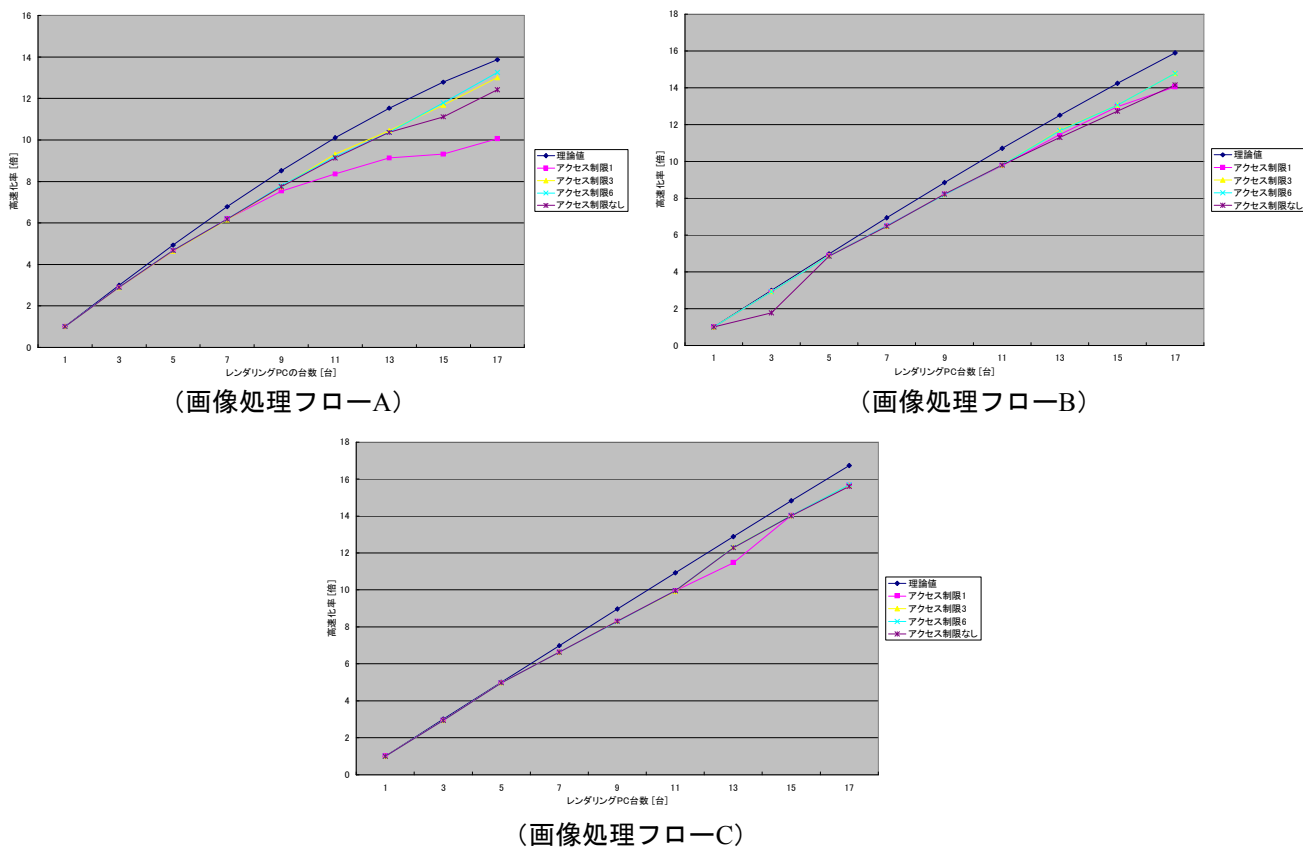


図 6 アクセス許可台数の高速化率への影響

3 つの画像処理フロー中で、最もレンダリング処理時間が短く、ある一定期間のディスクアレイに対するアクセスが多い画像処理フローA では、同時アクセス許可数の違いによる高速化率の差が大きくなっている。そして、その差は、画像処理フローB、画像処理フローCとレンダリング処理時間が長くなり、ある一定期間のディスクアレイへのアクセスが減るにつれて小さくなっていることが分る。

また、同時アクセス許可数を 6 としたときがどの画像処理フローを実行したときにも高い高速化率を示しており、後工程のみの所要時間を考えた場合には、同時アクセス許可数を 6 に設定するのが良いことが分かる。しかし、制御 PC は、レンダリング PC を用いてレンダリング処理と動画作成処理を実行する後工程の制御の他に、テープライブラリやテープドライブを制御して次に実行される画像処理フローの入力画像データの準備などを行わなければならない。そのため、後工程の制御だけに CPU を使用することはできない。上記の結果をみれば、後工程の所要時間が最小となるのは、同時アクセス許可数が 6 の場合であるが、同時アクセス許可数を 3 とした場合と 6 とした場合の差は僅かである。同時アクセス許可数に大きな値を設定したほうが、制御 PC の CPU 負荷が増えるため、同時アクセス許可数は 3 に設定することにした。

次に、ディスクアレイの代わりに一般的な IDE のハードディスクを取り付け、同様の計測を行った結果を図 7 に示す。図 7 は、画像処理フロー A を実行した場合の結果である。

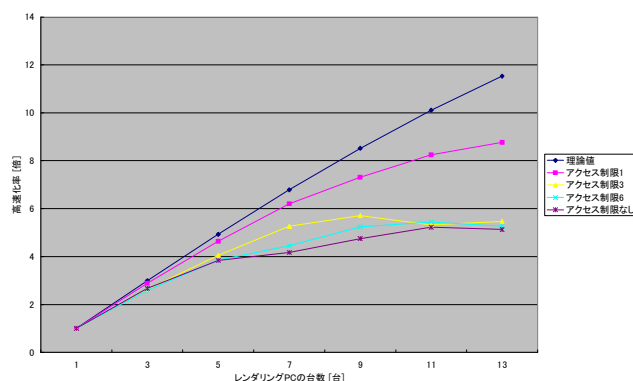


図 7 アクセス許可台数の高速化率への影響（画像処理フロー A、IDE HDD）

ディスクアレイを使用した場合には、同時アクセス許可数を増やすことによって、高速化率が上昇しているが、図 7 では、同時アクセス許可数を 1 としたときが最も高速化率が高く、同時アクセス許可数を増やすことで、逆に高速化率は低くなってしまっている。さらに、ディスクアレイを使用した場合には、13 台のレンダリング PC を用いて 1 台のレンダリング PC を使用した場合の約 10.3 倍まで高速に処理ができるようになったのに対して、通常の IDE ハードディスクを使用した場合には約 8.7 倍で留まっている。

このように、画像処理フロー実行アプリケーションでは、制御 PC に接続されたハードディスクに対するアクセスが多く、このハードディスクの性能が全体の性能に大きく左右する。そのため、このハードディスクの選定には、非常に注意が必要である。

7 効率的なジョブ管理方法

上述したように、本システムでは、画像処理フローを前工程、後工程の 2 つの工程に分けて実行する。そして、2 つの異なる画像処理フローを考えた場合、一方の画像処理フローの前工程ともう一方の画像処理フローの後工程との関連性は無い。そのため、ある画像処理フローの後工程が実行されているときに、他の画像処理フローの前工程を実行することができる。

これにより、後工程が前工程の終了に待たされる時間を短縮することが可能である。しかし、この場合、与えられた複数の画像処理フローをどの順番で実行するかによって、総所要時間が変わってくる。本システムでは、スケジューリングの目標が総所要時間最小で、工程数 2 のフローショップの場合に、簡単に最適解を求めることのできるジョンソン法[圓川 1996]を用いて、実行順序を決定する。ジョンソン法とは、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、前工程は処理時間の短い順に、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、まだ割り付けが終わっていない全てのジョブの中から、最も処理時間の短いものを探し、それが前工程のものであれば、そのジョブをスケジュールの前から割り付け、後工程のものであれば、そのジョブを後ろから割り付ける方法である（図 8）。

以下、表2に示す3つの画像処理フローを用いて、ジョンソン法によるスケジューリングの効果を評価する。

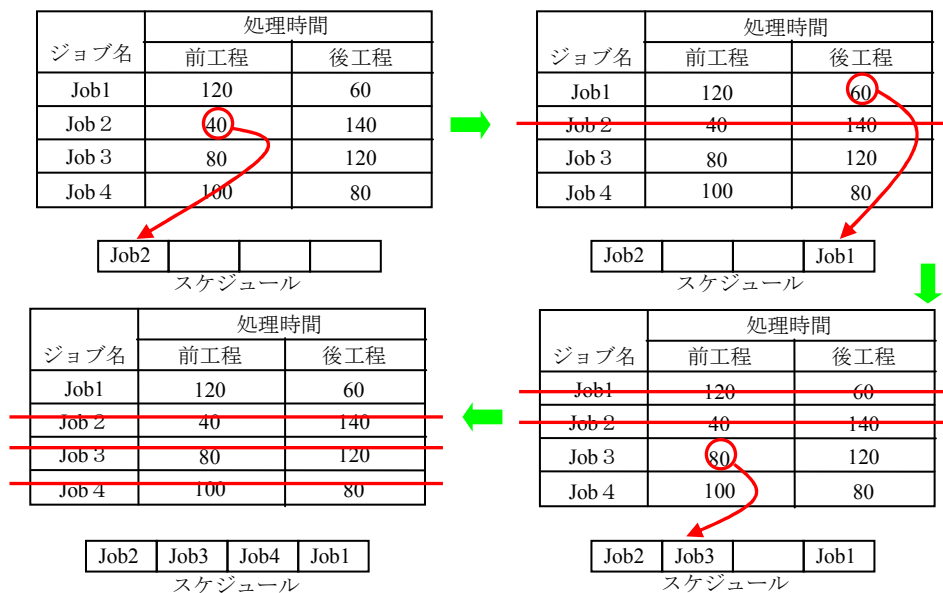
表2に示す3つの画像処理フローを実行する場合、実行順序の組み合わせとして6通りが考えられる。表3は、それぞれの順序で3つの画像処理フローを実行したときの総所要時間を計測した結果である。ここで、ジョンソン法によって選択された実行順序は、No. 6であった。

表3から分るように、ジョンソン法によって選択された順序で実行した場合が最も総所要時間が短くなっている。そして、総所要時間が最も長いNo. 1との差は、約11分である。

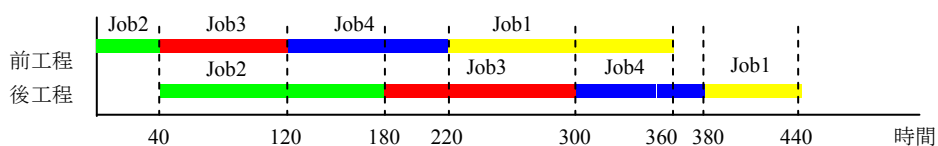
また、ある画像処理フローの後工程実行中に他の画像処理フローの前工程を実行する動作を行わずに、3つの画像処理フローを1つずつ順番に実行した場合の総所要時間は、表2に示す全ての処理時間を合計することで算出できる。その場合の総所要時間は3991[sec](66分31秒)である。この方法と表3のNo. 6の総所要時間とを比較した場合、約20分もの差が生じている。

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8 ジョンソン法

表 2 各画像処理フローの前工程・後工程処理時間

No.	画像処理フロー名	フレーム数 [枚]	前工程処理時間[sec]	後工程処理時間[sec]
1	画像処理フローD	210	853	172
2	画像処理フローE	20	807	429
3	画像処理フローF	60	637	1093

表 3 実行順序による処理時間の違い

No.	画像処理フロー Dの実行順番	画像処理フロー Eの実行順番	画像処理フロー Fの実行順番	総所要時間	
1	①	②	③	3468[sec]	57分48秒
2	①	③	②	3061[sec]	51分1秒
3	②	①	③	3386[sec]	56分26秒
4	③	①	②	3051[sec]	50分51秒
5	②	③	①	2808[sec]	46分48秒
6	③	②	①	2775[sec]	46分15秒

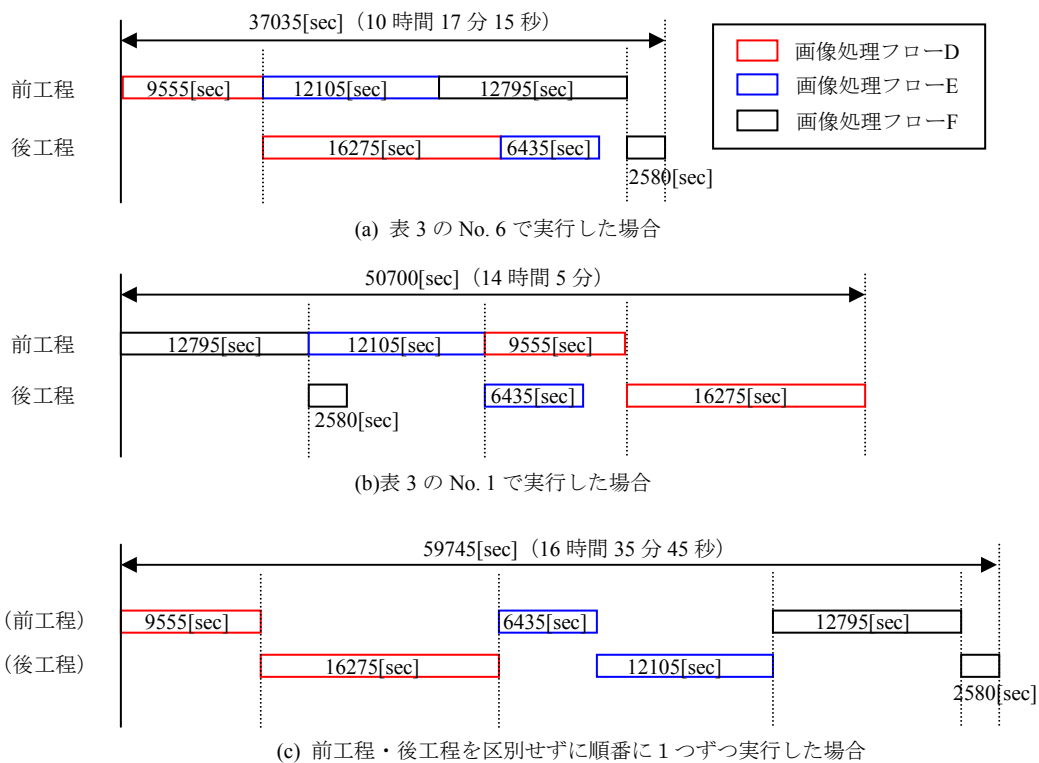


図 9 表の処理時間が15倍になった場合の総所要時間の比較

11分、20分という総所要時間の違いは、それ程大きく思えないかもしれないが、今回の計測で使用した画像処理フローの処理フレーム数は、表 2 から分るように、最も多いもので210フレームである。通常、我々が作成するコンテンツは、3600～5400フレーム程度である。ここで、前工程の処理時間を考えると、テープの読込み開始位置を定める移動時間とデータの読込み時間の2つからなる。また、後工程は、レンダリング処理時間と動画作

成処理時間の 2 つからなる。このうち、処理時間が処理フレーム数に依存しないのは、前工程におけるテープ移動時間のみであり、他は処理フレーム数に比例して処理時間も増える。そのため、前工程、後工程の処理時間は、処理フレーム数にほぼ比例して増加すると考えられる。そして、例えば、表 2 に示した画像処理フローの 15 倍のフレーム数を処理した場合に、前工程・後工程の処理時間が共に 15 倍になったと考え、表 3 の No. 1、No. 6 および前工程・後工程を区別せずに 3 つの画像処理フローを順番に 1 つずつ実行した場合の実行時間は図 9 のようになる。この図から、(a)と(b)の差は、約 3 時間 47 分、(a)と(c)の差は、約 6 時間 18 分となる。

以上から、1 つの画像処理フローの実行を前工程と後工程に分け、複数の画像処理フローを実行する場合に、ある画像処理フローの前工程実行中に、他の画像処理フローの後工程を実行し、それらの最適な実行順序をジョンソン法によって自動的に求める機能の効果は非常に大きいといえる。

なお、映像編集システムの実装においては、各 PC の処理負荷の状態がリアルタイムで表示されるジョブ負荷モニタ用 GUI を開発した (図 10)。これにより、どの PC のどの処理が、全体処理においてボトルネックになっているかをビジュアルに確認することができる。



図 10 ジョブ負荷モニタ用 GUI

8 従来システムとの比較

我々がこれまで使用してきた映像編集システムを用いて、代表的な IPD 用コンテンツを編集した場合の編集時間を表 4 に示す。ただし、表 4 に示した編集時間は、編集が終了した画像処理フローの実行時間のことであり、シナリオや設計書の作成や画像処理フローの編集時間は含まない。

また、これら 3 コンテンツの素材画像データの総データ量は 86 GByte であるが、従来システムが持つディスクアレイのディスク容量は約 60 GByte である。そのため、実際には、作成した出力画像をテープに移動して空きディスク容量を確保するなどの作業が必要となる。従って、従来システムで、実際に上記 3 コンテンツを作成する場合には、表 4 に示し

た時間より多くの編集時間を必要とする。しかし、本システムが持つディスクアレイのディスク容量が 1.3 TByte と非常に多いため、この時間を加えた場合には単純な比較ができなくなってしまう。そのため、表 4 の編集時間には空きディスク容量を確保するための画像データのテープへの移動時間は入っていない。

表 4 従来システムを用いたコンテンツ編集期間

コンテンツ名	編集時間		合計時間
	前工程	後工程	
コンテンツ FH	6 時間	19 時間 12 分	25 時間 12 分
コンテンツ OA	2 時間	30 時間 42 分	38 時間 42 分
コンテンツ AM	2 時間	18 時間 30 分	26 時間 30 分
コンテンツ総編集時間			90 時間 24 分

次に、同じコンテンツを本システムで編集した場合の編集時間を表 5 示す。表 5 の編集時間についても、表 4 の場合と同様に、編集が終了した画像処理フローの実行時間を示している。なお、表 5 の計測は、レンダリング PC を 20 台使用し、それぞれのコンテンツを個別に実行した場合の個々の編集時間を示している。

表 5 本システムを用いたコンテンツ編集期間

コンテンツ名		編集期間		合計時間
		前工程	後工程	
コンテンツ FH	シーン 1	8 分 22 秒	3 分 39 秒	12 分 01 秒
	シーン 2	20 分 49 秒	46 分 14 秒	1 時間 7 分 03 秒
	シーン 3	20 分 52 秒	46 分 15 秒	1 時間 7 分 07 秒
	シーン 4	10 分 40 秒	10 分 7 秒	20 分 47 秒
コンテンツ OA		26 分 46 秒	4 時間 14 分 16 秒	4 時間 41 分 02 秒
コンテンツ AM		28 分 49 秒	2 時間 42 分 51 秒	3 時間 11 分 40 秒
コンテンツ総編集時間				10 時間 39 分 40 秒

また、コンテンツ FH、コンテンツ OA、コンテンツ AM の画像処理フローを同じ優先順位で実行し、ジョンソン法によって実行順序を決めた場合の編集時間は、9 時間 9 分 29 秒であった。

以上の結果から、各コンテンツを個別に実行した場合、従来の約 1/8.5 の編集時間でコンテンツの編集作業が可能であり、3 コンテンツの画像処理フローを同じ優先順位として実行した場合には、従来の約 1/10 の編集時間で可能となった。

9 リアルタイム映像編集処理機能

ここまで、映像編集システムのオフライン処理について述べた。複数の PC を用いて画像処理の並列化を行うことにより、従来の映像制作に比べ大幅にコンテンツ制作時間を短縮することができた。しかし実際の映像制作過程では、映像表示環境の特性に合わせた色補正などの最終的な微調整のために、表示環境での確認と映像編集システムでのオフライン画像処理を交互に繰り返さなくてはならず、処理時間や作業工数が大幅に増えてしまうという問題があった。

そこで、この最終調整処理を容易に行うために、プレイヤー側のハードウェアでリアルタ

イムに画像処理を行う仕組みについて検討・実装を行った。

9.1 開発システムのアーキテクチャ

図 11 発したアーキテクチャを示す。本システムは、統括サーバと描画サーバにより構成される。

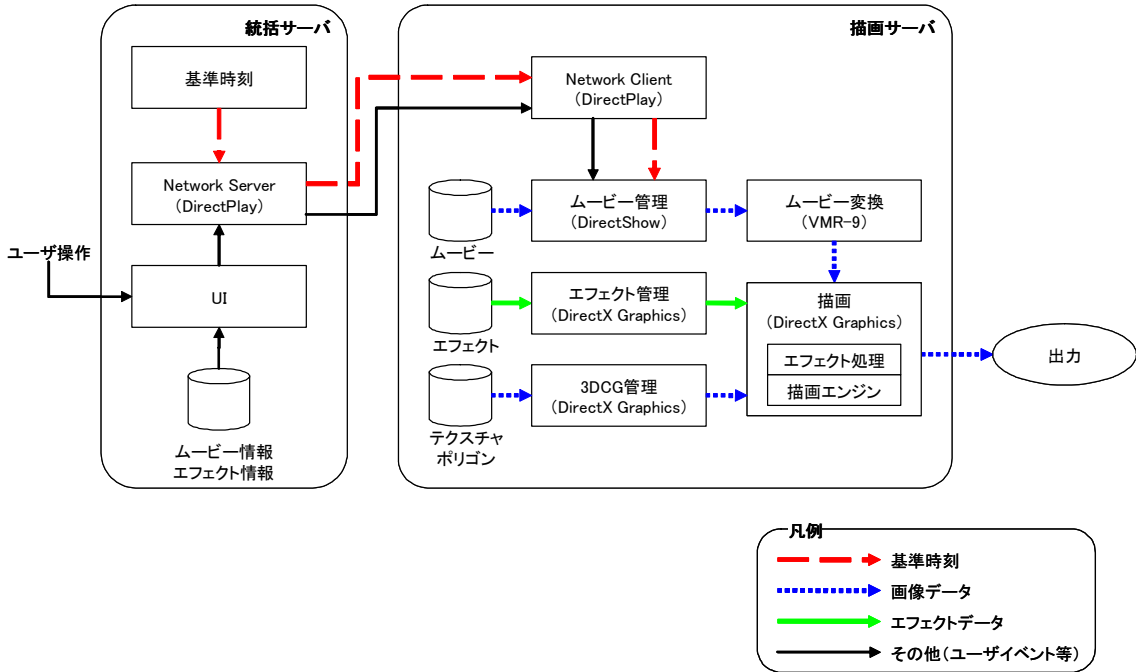


図 11 リアルタイム映像編集機能のシステムアーキテクチャ

9.2 フィルタ処理機構

フィルタ処理に関しては、主に、多重フィルタ、時間軸フィルタ、外部データ入力、シェーダ言語の拡張、UI 設計、の 5 つ機能の新規開発を行った。

これにより、フェードイン・フェードアウト、色相変換、フーリエ変換、モノクロ化、セピア化、モザイク、輪郭抽出、ポスタリゼーション、ガウスなどの様々な映像処理をリアルタイムに実施することを可能にした。

9.3 効果

本開発機能を、実際のコンテンツ制作者に使用してもらい、細かな調整が容易に出来ることを確認した。

IPD 用映像コンテンツ制作では、ディスプレイが持つ超高精細・広視野角といった特殊性のため、細微な画像処理であってもこれをオフライン処理で行おうとすると、その処理時間が膨大になってしまうといった問題がある。本開発では、色補正などの最終的な微調整を、プレイヤー側のハードウェアでリアルタイムに行う仕組みを実装した。これにより、従来、表示環境での確認と映像編集システムでのオフライン画像処理を交互に繰り返しながら、膨大な時間をかけて行っていた最終調整を、表示環境側だけで即座に実施可能とし、コンテンツ制作時の効率を大幅に向上させることができる。コンテンツ受注時における工数削減に、大きな効果があるものと期待される。

(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

1 概要

IPD 用コンテンツは、高精細・広視野角を特徴とするため、扱う入力画像も高解像度・大容量なものとなる。例えば、横 6000 ピクセル、縦 3000 ピクセルの画像 1 フレームあたりのデータ量は、約 72 MByte（非圧縮、1 ピクセル 4 Byte 表現の場合）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 30 frame/sec と仮定すると、出力画像だけで約 386 GByte となる。

通常、コンテンツ制作では、1 フレームの出力画像を作成するのに複数の入力画像が必要であること。またノンリニア編集システムを用いた映像編集では、入力画像とそれを加工して得られる出力画像を保存する空き容量が必要となることを考えると、3 分程度のコンテンツ制作に 1 TByte 以上のデータが必要となることも少なくない。

このため本編集システムでは、このような超大容量のデータを如何に効率的に扱うか、すなわちデータマネージメントが非常に重要となる。

2 各記憶装置の特徴とデータの流れ

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、各大容量記憶装置の特徴を示す。

大容量テープドライブは長寿命であり、精密機器であるハードディスクなどに比べて振動や衝撃に強い。そのため、データの長期保存に適している。しかし、ハードディスクに比べて低速であり、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に著しく効率が悪くなる。

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 にすることで、ディスクアレイを構成するハードディスクのうちの 1 台が故障してもデータを復旧することができる。

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

本システムでは、これらの特徴を持つそれぞれの記憶装置を図 12 すように組み合わせることとした。

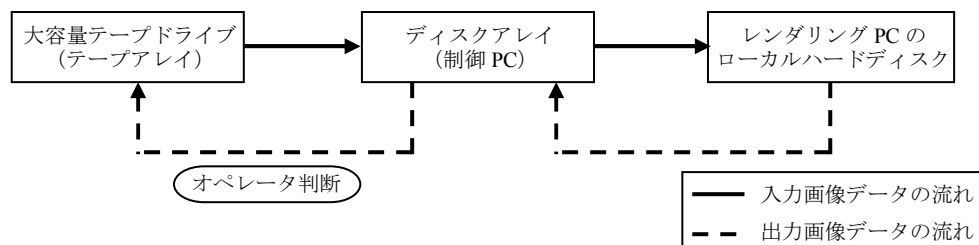


図 12 本システムにおける画像データの流れ

テープから供給される入力画像データは、ディスクアレイ経由で各レンダリング PC に与えられる。レンダリング PC における画像処理によって作成された出力画像は、ディスクアレイに移動する。これは画像処理フロー実行プログラムによって自動で行われる処理とする。ただし、ディスクアレイ上に移動された出力画像データをテープに移動する場合には、オペレータ判断を必要とする。これは、作成された画像データの中に、オペレータの操作ミスや思い違いによって、オペレータの意図した結果と異なるものが含まれるため、必要な画像データだけをテープに保存させるには、オペレータによる確認作業が必須であると考えられるためである。

ここで、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 に供給する際、ディスクアレイを経由させることで、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、以前に処理した入力画像がディスクアレイ上に残っていることが考えられるため、テープからの読み込み開始前に、ディスクアレイにおけるデータの存在確認を行うことで、テープからの読み込み頻度を減らすことができる。さらに、入力画像データのテープからディスクアレイへの読み込みと、ディスクアレイからレンダリング PC への供給は、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、複数の画像処理フローを順番に実行する場合、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ことができる。

また、入力画像データをディスクアレイからレンダリング PC に供給する場合、レンダリング PC のローカルハードディスクに書き込んだ後、メモリに読み込んで画像処理を行う。

これによって、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ローカルハードディスク内に保管しておくことができる。通常、あるコンテンツの制作期間中は、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で、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効果が期待できる。

3 データの管理方法

前述したように、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だけデータの移動またはコピーを行う。この存在確認を効率良く行うためには、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、コンテンツ編集は、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でデータを管理する。

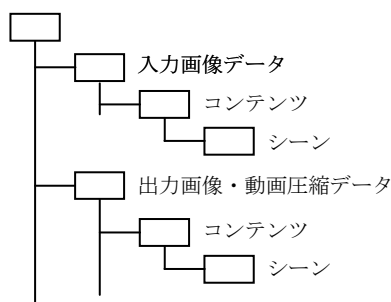


図 13 データを管理するディレクトリ階層

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はディレクトリによるデータ管理が一般的である。しかし、本システムで使用するテープアレイには、そのような管理方法は用意されていない。そのため、テープアレイではテープ管理データベースとテープライブラリ管理テーブルを用いて、ディレクトリによるデータ管理を実現する。

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。よってテープに番号（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。テープ管理データベースおよびテープライブラリ管理テーブルでは、テープを特定するためにこのテープ番号を使用する。従って、ディレクトリ構成によるデータ管理の対象となるテープは、テープ番号のバーコードが貼り付けられたテープに限定する。

テープ管理データベースは、論理パス名で指定されたデータがどのテープのどの位置に記録されているかを調べるために用いられるデータベースである。本システムで作成される全てのテープ（テープ番号のバーコード付き）の内容が登録される。そのため、このデータベースの登録の有無を確認することによって、そのデータがテープに書き込まれているかどうか分かる。テープ管理データベースの管理内容を表 6 示す。また、テープ管理データベースの更新は、データをテープに書き込む際に行う。

表 6 テープ管理データベースの管理内容

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番号
3	テープポジション	データが記録されているテープ上の位置 (書込、読込開始位置)
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は、テープライブラリ内にある 35 本のテープとそのテープの種類を調べるために用いられるテーブルである。テープの種類には、空テープと非空テープがある。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になったテープである。また、非空テープとは、必要なデータが記録されたテープである。テープライブラリ管理テーブルの管理内容を表 7 示す。テープライブラリ管理テーブルの更新は、テープライブラリの起動時とドアの開閉時に発行さ

れるイベントの検出で行う。

表 7 テープライブラリ管理データベースの管理内容

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番号 (表 6 No. 2 と同じ)
2	テープライブラリ位置番号	テープライブラリ内のテープ格納位置 (上段から順に 1~35 までの数字が割当てられている)
3	テープ種別	テープの種類

また、本システムでは、記録したいデータが1本のテープに入りきらない場合、テープドライブに他の空テープを入れ、データの書き込みを続ける。そして、そのようなテープからのデータ読み込み中にテープの終端を検出すると、テープドライブに入っているテープと続きのデータが保存されたテープを自動的に入れ替えて読み込みを続行する。このような動作を確実に行わせるためには、必要な画像データを保管したテープがテープライブラリ内に入っていないなければならない。そのため、必要な画像データが分かった段階で、その画像データがどのテープに入っているかを検出し、オペレータにそれらのテープをテープライブラリに入れさせる必要がある。

必要な画像データがどのテープに保存されているかは、テープ管理データベースの管理情報から特定可能であるが、テープ管理データベースの管理情報を用いた場合、必要な画像データ数が多くなることによってテープの特定に時間を要してしまう。そのため、効率良く必要なテープを特定するために、次テープ管理データベースを用いている。

次テープ管理データベースは、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ために用いられる。次テープ管理データベースの内容を表 8 示す。次テープ管理データベースの更新は、書き込み中のテープの空き容量不足を検出したときに行う。

表 8 次テープ管理データベースの管理内容

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番号 (表 6 の 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のデータが記録されたテープの番号

4 コンテンツ編集の流れと画像データの状態遷移

ここでは、コンテンツ編集作業の流れを追いながら、主にディスクアレイ上にあるデータの状態遷移を示す。ディスクアレイ上の画像データは、シーン単位に扱われる。そしてシーンには、処理可能状態、処理不可能状態があり、処理可能状態の中には、削除可能状態、削除不可能状態がある。

処理可能状態とは、そのシーンがシステムの処理対象になっている状態であり、この状態になれば(処理不可能状態)、レンダリング処理や動画作成処理を行うことができない。また、システムは画像処理フローの実行において、ディスクアレイの空き容量が不足した場合、不要なシーンの削除を自動で行う。このような削除の対象になっている状態が削除

可能状態である。

ただし、前提条件として、あるシーンに対して何らかの手を加えたシーンは、別のシーン名が付けられるものとし、同名のシーンには異なる内容の画像データが含まれることはないものとする。

以下、入力画像データの準備、画像処理フローの投入、画像処理フローの実行、納入物の作成とバックアップテープの作成の 4 場面に分け、各状態の意味と画像データの状態遷移を示す。

4.1 入力画像データの準備

通常映像データは、プロダクションや映像制作会社に依頼して作成されることが多い。そして、作成された映像データは、テープや DVD などのメディアで提供される。よってこのような映像データは、まずオペレータが手動でディスクアレイ上に置くことによって始めて、利用可能な状態になる。このとき、映像データは、図 13 に示すように、コンテンツディレクトリの下にあるシーンディレクトリに保管する。このように、映像データがコンテンツ別、シーン別のディレクトリに保管されている状態を処理可能状態と呼ぶ。

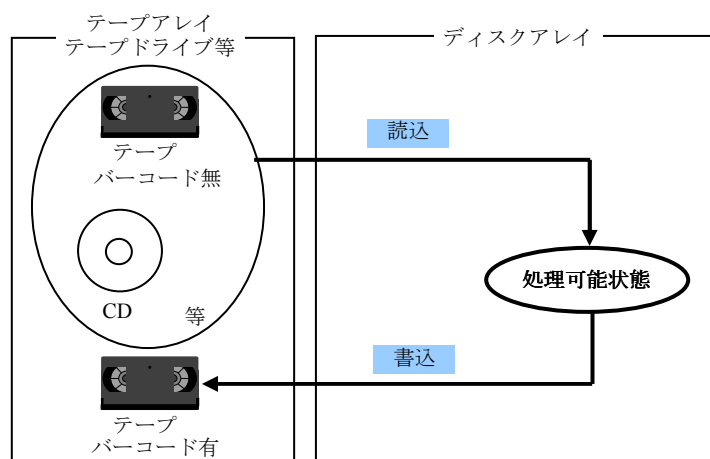


図 14 入力画像データの準備

オペレータは、処理可能状態にあるシーンを画像処理フロー編集プログラムに読み込み、そのシーンの加工手順を画像処理フローで表現していく。編集された画像処理フローは、ファイル（画像処理フロー設定ファイル）として出力される。

このとき、処理可能状態にある映像データを使用した画像処理フローはレンダリングの実行が可能となる。しかし、処理可能状態にある映像データをテープ番号のバーコードが付いたテープに保存しておくことで、画像処理フローの実行時に該当映像データがディスクアレイ上に無い場合でも、これを自動的にテープから読み込んで実行できるようになる。

4.2 画像処理フローの投入

画像処理フロー編集プログラムを用いて作成された画像処理フロー設定ファイルは、画像処理フロー実行プログラムで本レンダリングされる。

本レンダリングを行おうとする画像処理フロー設定ファイルを、画像処理フロー実行プログラムに投入すると、それを実行するために必要な入力画像データおよび実行すること

によって作成される出力画像データのデータ量が算出され、これとディスクアレイの空き容量とが比較される。そして、ディスクアレイの空き容量が不足する場合には、実行可能状態にあるシーンの中でディスクアレイ上に置いておく必要のないものを選択するようにオペレータに求める。

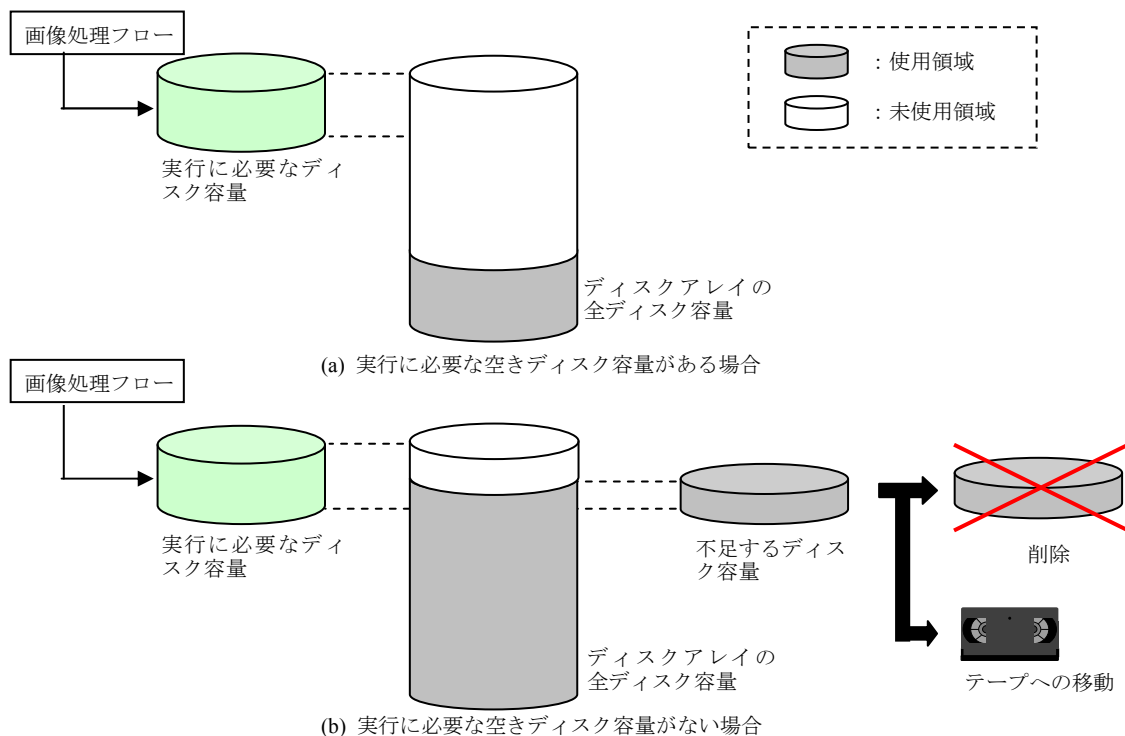


図 15 ディスクアレイにおけるディスク容量の確保

この要求に応じて選択されたシーンは削除可能状態に移行する。すなわち、処理可能状態にあるシーンは、その状態では削除不可能状態にあり、オペレータに削除可能として選択された時点ではじめて、削除可能状態になるのである。

ここで、画像処理フローの実行に必要な入力画像のデータ量は、テープ管理データベースに問合せで算出される。ただし、テープ管理データベースに該当シーンの登録が無い場合には、ディスクアレイ上で処理可能状態にあるシーンを検索する。そして、ディスクアレイ上にも存在しない場合には、その画像処理フローの投入は拒否される。

また、該当シーンがテープ管理データベースに登録されていた場合、そのテープがテープアレイ内にあるかどうかをテープライブラリ管理データベースに問合せる。そして、テープアレイ内に該当テープが存在しない場合には、テープ入替指示ダイアログボックスを表示して、テープアレイに入れるテープのテープ番号と、そのテープを入替可能なテープアレイの場所 (BIN 番号) を示す。テープライブラリは、テープを自動的に入替えることのできる機器である。しかし、それはあくまでも、テープライブラリに入れられる有限個のテープに対してである。そのため、画像処理フロー実行アプリケーションでは、投入された画像処理フローの実行に必要なテープが必ずテープライブラリにあることを、テープ入

替指示ダイアログによるガイド機能によって保障する。

4.3 画像処理フローの実行

投入された画像処理フロー設定ファイルは、前に説明したようにスケジューリングされ、順番に実行される。

ここで、画像処理フロー設定ファイルの実行は、前工程と後工程の2つの工程に分けて行われる。前工程とは、ディスクアレイ上に後工程を実行するために必要な入力画像を読み込み、出力画像を保存するための空き容量を確保する作業である。すなわち、削除可能状態にあるシーンをディスクアレイから削除して空き領域を確保する。そして、その作業終了後、テープから入力画像を読み込む。ただし、ここでの削除には2種類の方法が存在する。1つは、テープにシーンを書き込んでから削除するものと、テープへの書き込みを行わずに削除するものである。前者は、そのシーンに関する情報がテープ管理データベースに存在しない場合であり、後者は存在する場合に行われる。

テープ管理データベースは、シーンをテープに書き込んだときに更新される。従って、テープ管理データベースに、そのシーンに関するデータが存在するという事は、そのシーンが既にテープに書き込まれていることになるのである。まだテープに書き込んでいないシーンでもテープに書き込まずに削除して良い場合が考えられるが、シーンの削除を自動で行うため、データの安全性を考慮して、そのような動作は、自動削除の対象外とした。

これらの作業は、従来我々がIPD用のコンテンツを作成する時に、手動で行っていた作業である。すなわち本システムでは、画像処理フローの投入時にディスクアレイ上に置いておく必要のないシーンを選択するのみで、ディスクアレイ上のデータの整理と必要な入力画像データの準備が自動で行える。これにより、オペレータは実行する画像処理フローの選択と、必要であればディスクアレイに置く必要のないシーンの選択を行い、その画像処理フローの実行に必要なテープをテープ入替指示ダイアログボックスの指示に従ってテープアレイに入れるだけで良いのである。このため、オペレータの拘束時間は大幅に軽減される。

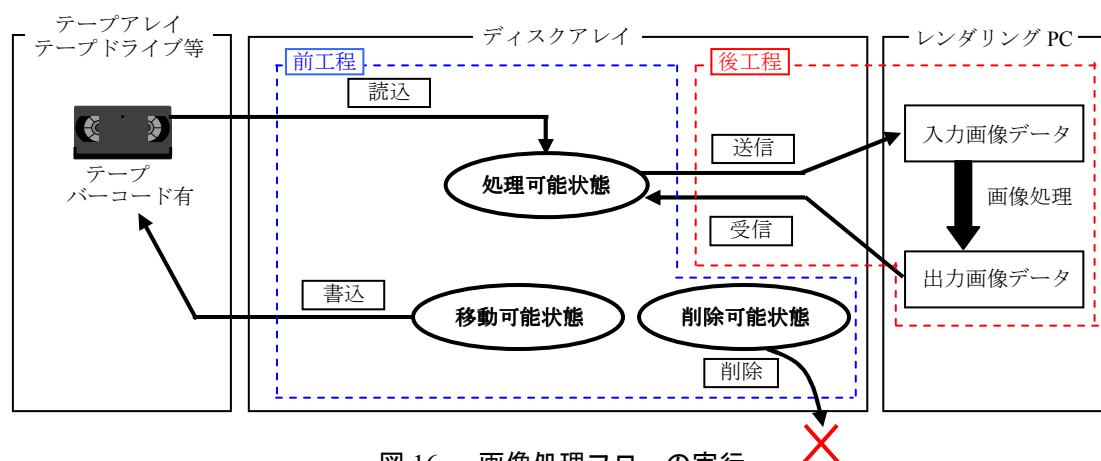


図 16 画像処理フローの実行

次に、後工程では、前工程で準備され処理可能状態にある入力画像を用いて出力画像を作成する (図 16)。

4.4 納入物の作成とバックアップテープの作成

コンテンツ編集が終了すると、オペレータは顧客への納入物と制作作業のバックアップテープを作成する。

納入物は、入力画像データ、出力画像データ、出力動画データの中から必要なものを、テープ番号のバーコードのないテープや DVD などのメディアに保存する。また、編集作業のバックアップテープは、テープ番号のバーコードが付いたテープに、入力画像データ、画像処理フロー設定ファイル、出力画像データ、出力動画データなどを保存する。これらの作業は、全てオペレータの手動操作で行われる。

バックアップテープの主な目的は、同じコンテンツを作り直すことができるようにすることである。そのような観点から考えると、入力画像と画像処理フロー設定ファイルがあれば、出力画像は作成できるため、出力画像のバックアップは取らない場合もある。

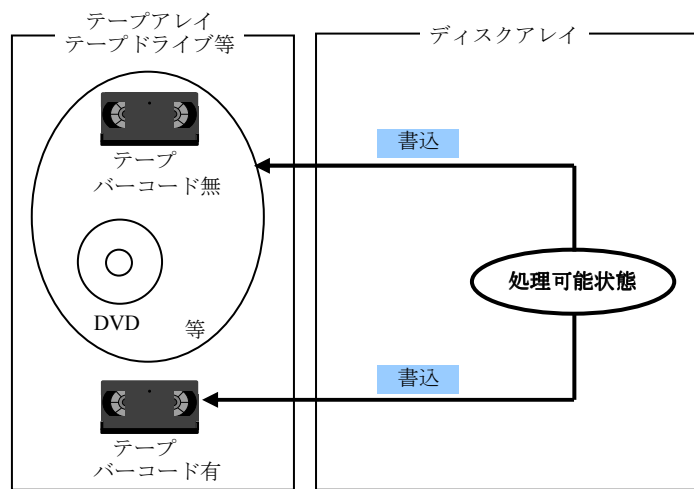


図 17 納入物の作成とバックアップテープの作成

5 バックアップ方法の検討

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。ここでは、これらが故障した場合のデータの安全性について検討する。

まず、ディスクアレイとレンダリング PC のローカルハードディスクについて考える。2 で述べたように、本システムでは、テープに記録された映像データをディスクアレイ経由で、レンダリング PC のローカルハードディスクに読み込み、そこで画像処理を行う。ディスクアレイは、RAID5 で使用しているため、それを構成するハードディスクのうちの 1 台が故障した場合には、データを復旧することが可能である。通常 2 台のハードディスクが同時に壊れる確率は非常に低いため、ディスクアレイ上にあるデータの安全性は確保されていると考える。これに対して、レンダリング PC のローカルハードディスクが故障した場合には、データを復旧することはできない。

そのため本システムでは、レンダリング PC のローカルハードディスクに大切なデータを置かないように制御する。作成された出力画像および出力動画は速やかにディスクア

レイに移動する。また、レンダリング PC のローカルハードディスクには、処理に使用した入力画像が保存されるが、この入力画像データは、ディスクアレイ上にあったデータのコピーであり、レンダリング PC のローカルハードディスクの故障によって消失してしまっても問題にならない。

次にテープが壊れ、その記録内容が読めなくなった場合を考える。現在、本システムでは、このようなトラブルへの対処方法として、同じテープ番号を持った複製テープを作成しておくことが有効である（方法 1）。しかし、この方法では、バックアップテープと同じ本数のテープが必要となる。

この問題に対して、図 18 に示す方法（方法 2）の採用が考えられる。図 18 は、2 本のテープを使用してデータのバックアップを行っているコンテンツの例である。この例において、テープ 1、テープ 2 のどちらか 1 本の読込みができなくなったときのために、テープ 1 とテープ 2 の同じ位置、同じデータ量の値を取得し、その加算値を同じ位置、同じデータ量の領域に書き込んだテープ 3 を作成する。このようなテープを作成しておくことで、例えばテープ 2 が読込めなくなった場合に、テープ 3 とテープ 1 から取得した値の引き算を実行してテープ 2 のデータが復元できる。ただし、この方法でデータが復元できるのは、コンテンツのバックアップテープ中の 1 本が故障した場合のみである。

以下、上記 2 つの方法の比較を行う。まず、記録されたデータの安全性を確保するために必要となる冗長なテープ（冗長テープ）の本数を比較すると、方法 1 では、コンテンツのバックアップテープと同じ本数のテープが必要となるが、方法 2 では、コンテンツのバックアップテープの本数に関わらず 1 本で良い。

次に、冗長テープの作成にかかる時間を概算してみる。1 本のテープが記録できる最大のデータ量は 200 GByte であり、テープデバイスの最大転送速度は 24 MByte/sec である。テープの読込・書込が最大転送速度で実行できたとすれば、1 本のテープの読込・書込時間は、約 2 時間 20 分となる。方法 1 では、冗長なテープの作成に、コンテンツのバックアップテープを読込み、同じ本数だけ書き込む必要がある。そのため、図 18 の例では、読込み時間に 4 時間 40 分、書込み時間に 4 時間 40 分が必要となり、合計 9 時間 20 分必要である。また、方法 2 では、冗長なテープの作成に、コンテンツのバックアップテープを読込み、1 本のテープに書き込む必要がある。そのため、図 18 の例では、読込み時間に 4 時間 40 分、書込み時間に 2 時間 20 分が必要となり、合計 7 時間必要となる。

次に、実際にテープの故障が発生した場合に、データの復旧に要する時間を比較する。方法 1 は、冗長テープがそのまま利用可能であり、復旧には時間を要さない。方法 2 は、正常なバックアップテープと冗長テープのデータを読込んだ後、テープ 1 本の書込みが必要である。このため、図 18 の例では、読込み時間として 4 時間 40 分、故障したテープの作成に 2 時間 20 分が必要となり、合計 7 時間必要となる。

ただし、バックアップのデータの安全を確保する目的には以下の 2 つが考えられる。1 つは、メディアに単純な故障が起きたときに、記録したデータが無くならないようにすること。もう 1 つは、災害が発生したときに、記録したデータが無くならないようにすることである。メディアの単純な故障であれば、複数のメディアが同時に使用できなくなる確率は低いといえるが、地震や火事などの災害が発生した場合には、ほとんどのメディアが

使用できなくなると考えられる。

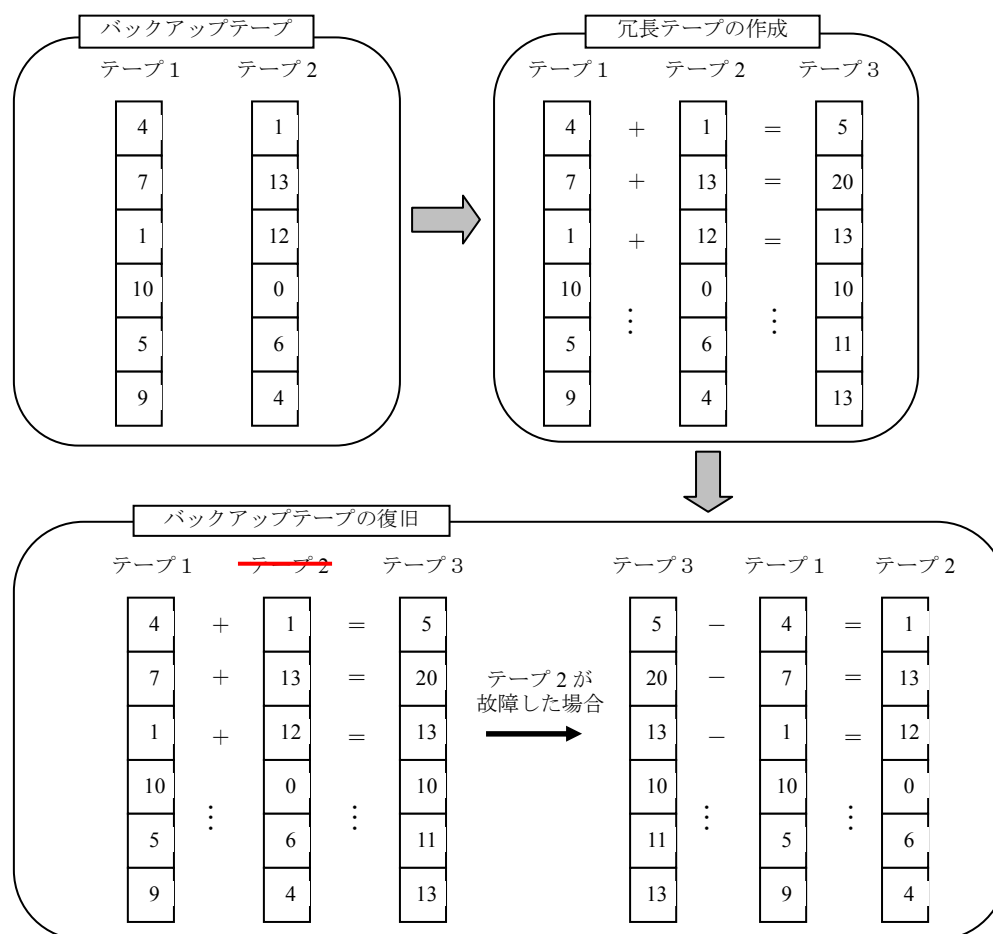


図 18 テープの冗長性を高める方法（方法 2）

地震や火事などの災害に対する対応を考えた場合、前者の方法であれば、冗長なテープを他の場所に保管しておくことによって対応が可能である。しかし、後者の方法では、あるコンテンツのバックアップテープが同時に 2 本故障してしまうとデータの復旧ができなくなってしまう。そのため、地震や火事などの災害で、バックアップテープが複数故障してしまう状況での対応は難しい。以上から、本システムにおいては、地震や火災などの対応を重視し、方法 1 を採用することにした。

表 9 方法 1 と 2 の比較

No.	項目	方法 1	方法 2
1	必要となる冗長テープの本数	コンテンツのバックアップテープと同数	1 本
2	冗長テープの作成時間 (図の場合)	9 時間 20 分	7 時間
3	故障したテープの復旧に要する時間 (図の場合)	不要	7 時間
4	地震や火災などへの対応	容易	困難

課題ア-2 超高精細・広視野角映像の撮影技術

本研究では、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あわせることで、1枚の超高精細・広視野角の動画の取得を可能にするカメラシステムに関する研究開発を行った。

具体的には、光学中心の一致を保証しないマルチカメラを対象として、撮影した映像を画像処理によって見た目に違和感無く接合する方法の検討、ならびにその原理実験を行った。一方、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について実験システムを試作し基本技術の検討・実験を行った。

さらに、移動カメラにより広域・高解像度映像を自動撮影する方法について研究を行った。移動カメラを未知環境にて自律的に移動させるために必要な、①環境の幾何情報を取得する方法、②移動カメラの自己位置を推定する方法、③幾何情報の地図としての記録を行う方法の研究を行った。

(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

高臨場感映像システムは、表示する映像が高精細で且つ広視野角である、という特徴を持つ。このような特徴を持つ映像の撮影には、一般のテレビや映画の撮影を前提としたカメラでは不十分であり、ここにおいて新たな撮影手段の提供が必要となる。その一つの解として、複数のカメラを異なる方向に向けて一度に撮影を行い、それらの映像から何変換などを施すことによりシームレスな1つの画像を作成するマルチカメラシステムがある。カメラを増やすことで全体の解像度を1台のカメラの場合に比べ何倍にも高めることができ、また各カメラを異なる方向に向けることで、全体として非常に広い視野角の映像が取得できるようになる。

このようなマルチカメラシステムを実現させる上で特に考えなければならないのが、単一視点制約とよばれる条件である[八木 2001][横矢 2000]。これは、複数カメラの撮影映像が、2次元画像処理によってシームレスに繋がるためには、撮影時に各カメラの光学中心(viewpoint)が完全に一致していなければならない、というものである。

しかしながら、実際のカメラの物理的・形状的な制約により、この条件を満たすようにカメラを配置することは、直接的には一般に不可能である。そこでこの問題を解決する方法として、ミラーを使う方法がある[Nalwa 1996][Kawanishi 1998][Majumder 1999]。しかし、その理想的な撮影を実現するためには、各カメラの光学中心を実機上で合わせる作業、すなわち、どのようにして各カメラの光学中心を正確に合わせるかというキャリブレーション手段が重要となる。

そこで本研究では、このミラー式マルチカメラにおける光学中心位置を一致させる目的に特化した、実際的且つ高精度なキャリブレーション手段の提供を目的とし、そこで必要となる技術開発を行った。

1 カメラモデルと一般のキャリブレーション方法

1.1 カメラモデル

カメラモデルには、コンピュータビジョンの分野などで広く用いられるモデル[Tsai 1987][Zhang 1998][出口 2002]を仮定する。内部パラメータとしては、焦点距離 $f^{(k)}$ 、画

素の縦横の長さをあらわす係数 $k_u^{(k)}$, $k_v^{(k)}$ 、画像座標系の u 軸と v 軸との角度 $\theta^{(k)}$ 、画像中心 $(u_0^{(k)}, v_0^{(k)})$ 、湾曲歪みパラメータ $\lambda_1^{(k)}$, $\lambda_2^{(k)}$ が定義され、外部パラメータとしては、カメラの光学中心位置を示す 3 次元ベクトル $\mathbf{b}^{(k)}$ 、カメラ方向を示す 3×3 の回転行列 $\mathbf{R}^{(k)}$ が定義される。なおここで k はカメラ番号である。

3次元空間内の i 番目の特徴点をワールド座標系における 3次元ベクトルで \mathbf{p}_i とあらわし、その点の第 k カメラ画像上の位置を $\mathbf{x}_i^{(k)} = (x_i^{(k)} \ y_i^{(k)} \ 1)^T$ とおく。これらの関係式

$$\mathbf{x}_i^{(k)} = C^{(k)}(N[\mathbf{A}^{(k)} \mathbf{R}^{(k)}(\mathbf{p}_i - \mathbf{b}^{(k)})]) \quad (1)$$

であらわされる。ここで $\mathbf{A}^{(k)} = \begin{pmatrix} f^{(k)} k_u^{(k)} & -f^{(k)} k_u^{(k)} / \tan \theta^{(k)} & u_0^{(k)} \\ 0 & f^{(k)} k_v^{(k)} / \sin \theta^{(k)} & v_0^{(k)} \\ 0 & 0 & 1 \end{pmatrix}$, $N[\]$ はベクトルの第 3

要素を 1 にする正規化操作である。 $C^{(k)}(\)$ は湾曲歪み変換をおこなう $\lambda_1^{(k)}$, $\lambda_2^{(k)}$ を係数とする関数で、歪みが無い理想的なカメラを仮定した場合の撮影位置を 3次元同次座標表現である。

1.2 一般のキャリブレーション方法

式(1) から未知数以上の拘束式が得られるように複数点の観測を行い、それらを連立させて解を求めるのが、カメラキャリブレーションの基本的な考え方である。

具体的手法としてはさまざまなものがあるが、1つの分類として、(1) 内部・外部パラメータを同時に求めるものと、(2) 内部パラメータは静的なもののみを求め、あらかじめ求めておき、外部パラメータのみを求めるものがある。また別の分類として、(a) 被写体の 3次元位置を既知とするものと、(b) これを未知とし複数画像から解くものがある。

(1)と(a)を組み合わせた条件は、カメラの内部仕様を測定する目的などにおいて用いられ、その代表的な手法が文献 [Tsai 1987] や文献 [Zhang 1998] などによるものである。(1)-(b) はいわゆるセルフキャリブレーションと呼ばれるもので、さまざまな条件設定が考えられ、例えば文献 [Hartley 1992] ではユークリッド座標値として求めるための条件などが論じられている。(2) の分類はカメラの位置・姿勢推定問題として扱われ、代表的なものとしては、(2)-(a) では透視 3点問題 [Wolfe 1991]、(2)-(b) では E 行列を用いる方法 [Tsai 1984] などが古くから研究されている。

最適化手法の選択や、数値計算時の初期値の問題など多くの条件が関係するため一概には言うことはできないが、一般に未知パラメータの自由度が多いほど、解を求める際の誤差に対するロバスト性が悪くなる。特にセルフキャリブレーションなどで非常に多くの未知パラメータを同時に求めなければならない問題設定においては、数式上は解が求まるものの、現実的な誤差の存在下においては実用的な推定は極めて困難である、との指摘が多くされている。

ところで、本研究が目的とするのは、複数カメラの位置 $\mathbf{b}^{(k)}$ ($k = 1, 2, \dots$) が一致しているか否かを判別することである。よって、それ以外のパラメータは必要無い。しかし一

般のキャリブレーション手法では、すべての未知パラメータを（陰であっても）正確に求めない限り、 $\mathbf{b}^{(k)}$ の精度を上げることはできず、多くのパラメータの誤差の影響を考慮する必要がある。また、たとえ上記(2)の方法によりあらかじめ内部パラメータを専用の装置で求めたとしても、そこで求まる値にある程度の誤差の混入は避けられないし、また実際にカメラを設置する際に、ピント調整などの影響で内部パラメータが変化してしまう可能性もある。

そこで本研究では、必要の無いパラメータに依存しない複数カメラの位置一致判別手段を考えていくことにする。

2 マルチカメラシステムにおけるキャリブレーション

2.1 単一視点制約

マルチカメラシステムでは、撮影した複数の画像に2次元変形処理をそれぞれ加えることで、シームレスに繋がった広視野角映像を作成する。ここで2次元変形処理というのは、この変形のしかたが、被写体の奥行き方向の位置によらない、ということの意味している。

さて、2次元変形処理で映像接続が可能となるためには、各カメラが単一視点制約を満たす、すなわち各カメラの光学中心位置が一致する必要がある。この意味を式を用いて説明する。

いま、任意の被写体の位置を、 $(X_i \ Y_i \ Z_i)^t$ のかわりに

$$\mathbf{p}_i = \mathbf{b}^{(1)} + \rho_i \mathbf{m}_i \quad (2)$$

とあらわす。 $\mathbf{b}^{(1)}$ は第1カメラの光学中心位置、 ρ_i は奥行き距離を示すスカラー、 \mathbf{m}_i は方向を示す3次元単位ベクトルである。もし第 k カメラの光学中心位置が $\mathbf{b}^{(k)} = \mathbf{b}^{(1)}$ と一致すれば、式(2)を式(1)に代入することで

$$\mathbf{x}_i^{(k)} = C^{(k)} \left(N [A^{(k)} R^{(k)} \mathbf{m}_i] \right) \quad (3)$$

が得られ、被写体の奥行き ρ_i とは無関係な式となる。これに対し $\mathbf{b}^{(k)} \neq \mathbf{b}^{(1)}$ とすると、 ρ_i を消去することができず、画像が奥行き ρ_i に依存するようになってしまう。

2.2 画像接続のための変形処理

以上より、各カメラが同一の光学中心位置 $\mathbf{b}^{(1)}$ をもつとすると、その他のカメラパラメータが撮影画像に及ぼす影響を、画像の2次元変形だけであらわすことができることがわかった。

画像をシームレスに繋ぎ合わせるということは、すべてのカメラ画像が同一のカメラパラメータのものとなるように、この2次元変形の逆変換処理を行うことを意味する。よってこの逆変換処理は、式(1)におけるパラメータ $A^{(k)}$ 、 $R^{(k)}$ 、 $\lambda_1^{(k)}$ 、 $\lambda_2^{(k)}$ を用いて決定することができる。

ところが実際のシステムでは、必ずしもこれらのパラメータを個々に求める必要は無い。

なぜなら、隣り合う画像がシームレスに繋がる変換処理を得ることが目的であって、その変換処理がどのような要素から成り立っているか、という情報は必要無いからである。具体的には、格子模様などのテストパターンを撮影して、それがシームレスに繋がるような2次元変換処理を直接求めてやればよく、これを $A^{(k)}$, $R^{(k)}$ 等の要素に分解する必要は無い、ということになる。

2.3 推定するパラメータの限定

以上述べてきたことを整理すると、マルチカメラシステムを実現するためには、各カメラの光学中心位置 $b^{(k)}$ を一致させることと、シームレスに繋げるための2次元変換処理を求めることが必要だが、どちらの場合にも、 $b^{(k)}$ 以外のカメラパラメータは必ずしも必要無いことがわかった。

そこで本節では、 $b^{(k)}$ を一致させることに焦点をあて、他のパラメータを陰にも陽にも用いないキャリブレーション手法を考えることにする。これにより、簡易で且つ高精度の測定が期待されるからである。

2.4 キャリブレーションの条件

マルチカメラシステムの実際の装置構成の観点から、キャリブレーション実施の条件を考えると、以下のようなものが挙げられる。

- (1) ミラーの配置上、隣り合うカメラ間で同一の被写体（オーバーラップ領域）を撮影することが一切できない。
- (2) カメラシステムは一般に大がかりなものとなり、視野角も広いため、カメラやキャリブレーションパターンを厳密に移動させたり回転させる操作が無いことが望ましい。

このような条件で適用可能なキャリブレーション方法を考える必要がある。

3 提案するキャリブレーション方法

3.1 受動的読み取り方法と能動的読み取り方法

一般のカメラキャリブレーションでは、キャリブレーションパターンなどを撮影した画像から、そのパタンの画像上での位置を与えられたものとして読み取る。よって観測パラメータは受動的なものであると言える。

これに対して本方法では、画像上のある位置を指定し、ここに写る被写体の位置はどこかを調べるという、新たな測定方法を考える。本方法は前記の方法と対応させて能動的である、ということができる。

具体的な手段としては、スクリーンに特徴点をプロジェクタで投影し、これをカメラで撮影して指定された位置にそれが写るようにダイナミックにプロジェクタの映像を変えていく、あるいは、図1に示すように、3次元配置既知の細かい格子模様をカメラで撮影し、指定された画像上の位置に対応するその格子模様上の位置を目視により読み取る、などの方法が考えられる。

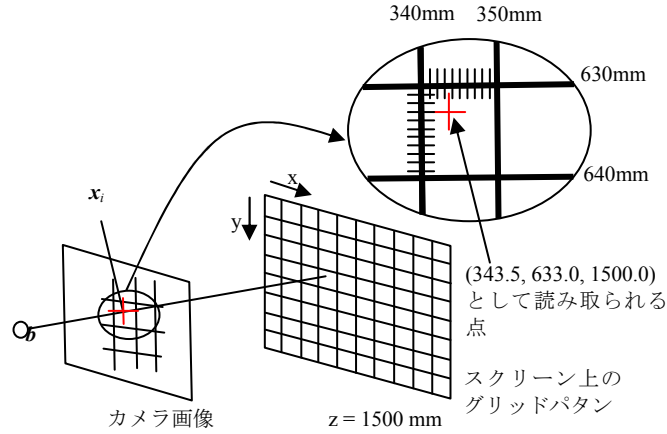


図1 提案方法における観測パラメータの読み取り手段例

3.2 提案方法の手順

- (1) ある3次元位置既知の特徴点 p_j の、第 k カメラでの画像上位置 $x_j^{(k)}$ を観測する。
- (2) $x_j^{(k)}$ と全く同じ場所に写る、 p_j とは異なる別の特徴点 q_j の3次元位置を、図1で示した方法により測定する。
- (3) このような点 p_j, q_j は式(1)より $C^{(k)}(N[A^{(k)}R^{(k)}(p_j - b^{(k)})]) = C^{(k)}(N[A^{(k)}R^{(k)}(q_j - b^{(k)})])$ を満たすが、 $C^{(k)}(\cdot)$ は1対1写像だから逆関数が存在して $A^{(k)}R^{(k)}(p_j - b^{(k)}) = \gamma_j^{(k)} A^{(k)}R^{(k)}(q_j - b^{(k)})$ の関係が得られる。ここで $\gamma_j^{(k)}$ は $\gamma_j^{(k)} \neq 0, 1$ なる任意のスカラーである。さらに $A^{(k)}R^{(k)}$ は正則だから逆行列が存在し

$$p_j - b^{(k)} = \gamma_j^{(k)} (q_j - b^{(k)}) \quad (4)$$

の関係が得られる。

- (4) 式(4)の幾何学的な意味は、カメラの光学中心 $b^{(k)}$ が、2点 p_j, q_j を結ぶ直線上にある、ということである。よって、2つ以上の式(4)が得られれば、すなわち2つ以上の特徴点の組 $\{p_j, q_j\}$ ($j = 1, \dots, J; J \geq 2$) が観測できれば、 $b^{(k)}$ を式(4)であらわされる直線の交点として一意に求めることができる。
- (5) 誤差の存在下においては、その交点は例えば

$$\sum_{j=1}^J D(\tilde{b}^{(k)}, p_j, q_j)^2 \leftarrow \min \quad (5)$$

であらわされる最適問題の解 $\tilde{b}^{(k)}$ として求めることができる。ここで $D(\tilde{b}^{(k)}, p_j, q_j)$ とは、2点 p_j, q_j を結ぶ直線と $\tilde{b}^{(k)}$ との距離である。なお式(5)の解は、代数的手法で求めることができる。

3.3 提案方法の特徴

ところで、式(4)あるいはその導出過程を注視すると、カメラの内部パラメータ $C^{(k)}(\cdot)$,

$A^{(k)}$ やカメラ方向 $R^{(k)}$ の値には全く影響されないことに気がつく。よって本方法は、カメラ位置のみを、それ以外のパラメータとは無関係に求めることができる。これにより、他パラメータの誤差の混入する可能性がなくなるため、高精度の測定が期待できる。

本方法の、幾何学的な意味を図2に示す。実際に得られる画像は、カメラ内部パラメータとカメラ回転 $R^{(k)}$ の影響で、理想的な透視投影画像に比べて、直線が歪んだり、場所が移動したり、あるいは拡大や縮小するなど、幾何学的な変形が行われて写る。しかしながら、もし2点 p_j, q_j と視点 $b^{(k)}$ が直線上にあれば、2点の画像はどのような変形が行われようとも、常に同じ点に写る。逆に、画像上で同じ点に写っていれば、 p_j, q_j と視点 $b^{(k)}$ は直線上にある。本方法はこの性質を利用したものである。

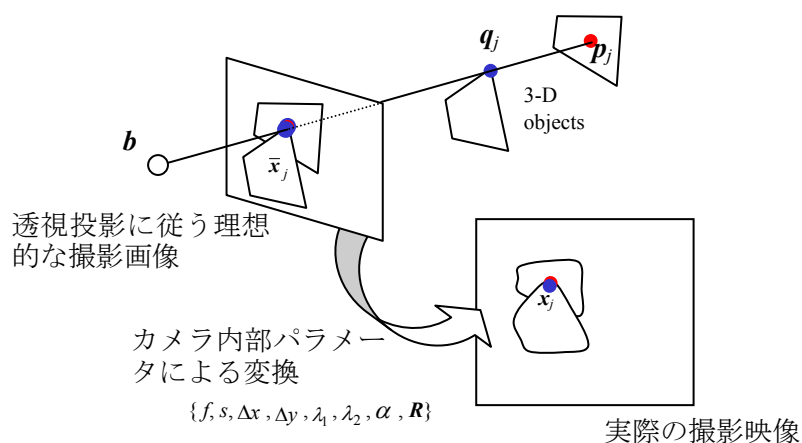


図2 提案方法の幾何学的な意味

4 応用方法

前章の方法では、図3(a)に示すように2つの観測値 p_j, q_j の相対位置関係がわかっていないなければならない。そのためには、カメラあるいはキャリブレーションパターンを正確な方向に、正確な距離だけ移動するなどの特別な機構を持った測定装置が必要で、前節で述べた条件(2)から望ましくない。

そこでここでは、このような機構なしにも、すなわち、図3(b)に示すように p_j と q_j がそれぞれ独自の座標系で測定され、お互いの関係がわからないときにも、キャリブレーション可能となる方法を示す。

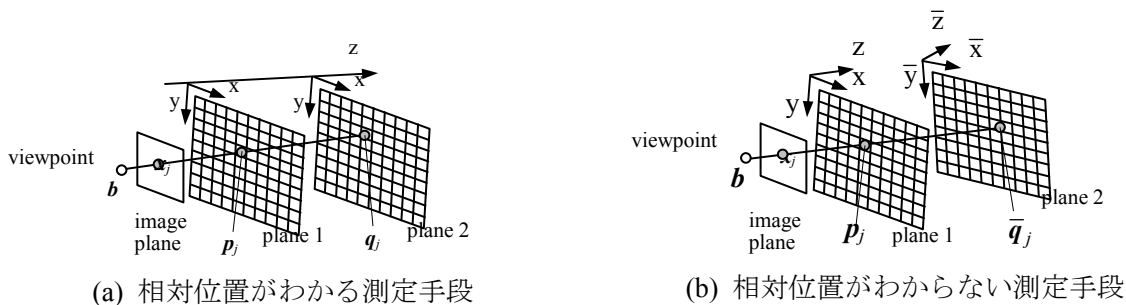


図3 2点の相対位置がわかる測定手段と、わからない測定手段

4.1 応用方法の前提条件とホモグラフィー

実際の測定の実施容易性から、3次元空間内の特徴点 p_j ($j = 1, 2, \dots$) は平面上にあり、 q_j ($j = 1, 2, \dots$) は別の平面上にあることを前提条件とする。

異なる座標系で表現されていることを明示するため、以下2つの観測点を p_j, \bar{q}_j と表記する。 \bar{q}_j を p_j と同じ座標系であらわしたものを q_j とおくと、両者の関係は

$$q_j = G\bar{q}_j - g \quad (6)$$

であらわされる。ここで G は座標系の回転を示す回転行列、 g は座標原点の移動を示す3次元ベクトルで、 G, g ともに未知である。

p_j, \bar{q}_j はそれぞれ平面上にあるから、平面の単位法線ベクトルを h_p, \bar{h}_q 、原点から平面までの距離を d_p, \bar{d}_q とおくと

$$(h_p, p_j) = d_p, \quad (7)$$

$$(\bar{h}_q, \bar{q}_j) = \bar{d}_q \quad (8)$$

の関係が成り立つ。 p_j, \bar{q}_j ($j = 1, 2, \dots$) が既知なので、 $\{h_p, \bar{h}_q, d_p, \bar{d}_q\}$ も既知とすることができる。

式(6)-(8) より、 p_j と \bar{q}_j の間に

$$p_j \approx B^{(k)} \bar{q}_j \quad (9)$$

の形の、ホモグラフィーと呼ばれる関係が成り立つことがわかる。ここで $B^{(k)}$ は

$$B^{(k)} \approx \left(I - \frac{(b^{(k)} h_p^t)}{d_p} \right)^{-1} G \left(I - \frac{G^{-1}((g - b^{(k)}) \bar{h}_q^t)}{\bar{d}_q} \right) \quad (10)$$

として形成されるものである。

4.2 $B^{(k)}$ の算出

式(10) であらわされる $B^{(k)}$ は、4つ以上の観測点の組み合わせ p_j, \bar{q}_j ($j = 1, 2, \dots, J$) で求められることが知られており、その最適値としては例えば

$$\sum_{j=1}^J [(\mathbf{B}^{(k)} \bar{\mathbf{q}}_j)^2 - (\mathbf{p}_j, \mathbf{B}^{(k)} \bar{\mathbf{q}}_j)^2] \leftarrow \min \quad (11)$$

として求められるものが知られている[Tsai 1984]。

4.3 $\mathbf{b}^{(k)}$ の算出 ($\mathbf{B}^{(k)}$ の分解)

2つのカメラで同一平面を撮影したときにも式(9)の関係が成り立ち、内部パラメータ既知のときに $\mathbf{B}^{(k)}$ をカメラ位置等の要素に分解する方法は数多く示されている。しかし今回のように $\mathbf{B}^{(k)}$ が式(10)で示されるように2つの平面から構成される場合には、同様の手法が利用できず、別の分解方法が必要となる。

そこでこの方法として、文献[守屋 2003]に示された方法を適用する。ここでは、レーザビームの複数平面上の投影位置から、ホモグラフィーの関係をあらわす行列 \mathbf{B} を求め、それを分解することでレーザビームの位置などを求めている。その問題設定は、今回の問題設定と幾何学的に等価であり、パラメータの表現方法に若干の違いがあるものの、行列 \mathbf{B} は式(10)と同じ形式で構成されている。よって、全く同様の手法を今回の $\mathbf{B}^{(k)}$ の分解に適用することができる。

結論をいうと、 $\mathbf{B}^{(k)}$ から式(10)における $\mathbf{b}^{(k)}$ 等の構成要素を一意に求めることはできない。これは、1自由度分、未知パラメータに比べて拘束式の数が足りないからである。わかるのは $\mathbf{b}^{(k)}$ が、ある円 $\Omega^{(k)}$ に存在する、ということまでである。詳しい説明は文献[守屋 2003]にゆずるが、この円 $\Omega^{(k)}$ は $\mathbf{B}^{(k)}$ と既知パラメータから代数的に求めることができるものである。

このように、カメラの光学中心位置を一意に決めることはできないが、複数のカメラの光学中心位置が一致しているかどうかの判別には、上記の知見を利用することができる。これを次節で説明する。

4.4 複数カメラの光学中心位置一致の判別

第 k カメラの光学中心位置を $\mathbf{b}^{(k)}$ とし、第 $k+1$ カメラの光学中心位置を $\mathbf{b}^{(k+1)}$ とし、両者が等しいかどうかの判別を行う手順は以下のとおりである。

- (1) まず、第 k カメラで、2つの平面上の4組以上の特徴点位置 $\mathbf{p}_j, \bar{\mathbf{q}}_j$ ($j = 1, 2, \dots, J$) の読み取りを図1の方法で行う。これを用いて、両者のホモグラフィー変換を行う行列 $\mathbf{B}^{(k)}$ を4.2の方法で求める。
- (2) 次に、第 $k+1$ カメラで、上と同じ2つの平面上の4組以上の特徴点位置 $\mathbf{p}_j, \bar{\mathbf{q}}_j$ ($j = J+1, J+2, \dots$) の読み取りを行う。これを用いて、両者のホモグラフィー変換を行う行列 $\mathbf{B}^{(k+1)}$ を求める。
- (3) もし、 $\mathbf{b}^{(k)} = \mathbf{b}^{(k+1)}$ であれば、同一の式(10)が構成されるので、 $\mathbf{B}^{(k)} \approx \mathbf{B}^{(k+1)}$ となることは明らかである。よって $\mathbf{B}^{(k)} \approx \mathbf{B}^{(k+1)}$ であることは、 $\mathbf{b}^{(k)} = \mathbf{b}^{(k+1)}$ のための必要条件である。
- (4) 一方、 $\mathbf{B}^{(k)} \approx \mathbf{B}^{(k+1)}$ なる $\mathbf{B}^{(k)}, \mathbf{B}^{(k+1)}$ が得られても、 $\mathbf{b}^{(k)}, \mathbf{b}^{(k+1)}$ がそれぞれ上記の円 $\Omega^{(k)} = \Omega^{(k+1)}$ のどこに存在するかは、前節の議論より決定することはできない。

い。ところで、平面 2 の平面 1 に対する相対的な傾きは、文献[守屋 2003]で述べられているように $\mathbf{b}^{(k)}$ あるいは $\mathbf{b}^{(k+1)}$ がこの円上を動くときに変化する。よって、もし $\mathbf{b}^{(k)} \neq \mathbf{b}^{(k+1)}$ と仮定すると、 $\mathbf{b}^{(k)}$ を用いて求めた平面 2 の傾きと、 $\mathbf{b}^{(k+1)}$ を用いて求めた平面 2 の傾きが異なるものになってしまう。これは、同じ 2 枚の平面を用いて第 k カメラの撮影と第 $k+1$ カメラの撮影を行う、という撮影方法に矛盾する。よって $\mathbf{b}^{(k)} = \mathbf{b}^{(k+1)}$ でなければならない。これより $\mathbf{B}^{(k)} \approx \mathbf{B}^{(k+1)}$ であることは、 $\mathbf{b}^{(k)} = \mathbf{b}^{(k+1)}$ であるための、十分条件になっていることもわかる。

以上をまとめると、 $\mathbf{B}^{(k)} \approx \mathbf{B}^{(k+1)}$ が言えるかどうかを調べるだけで、2 つのカメラの光学中心位置を、それぞれ値として特定することはできないものの、お互いが一致しているか、あるいは一致していないかの判別は可能である。次章では、このときにどの程度の誤差範囲でこの判別ができるのかを、実験とシミュレーションによって明らかにしていくことにする。

5 実験とシミュレーション

5.1 予備実験

まず、3 で述べた測定方法の有効性と、読み取り誤差の程度を見積もるための予備実験を、実際のカメラを用いて行った。

図 1 で説明した "カメラの撮影画像上の位置を指定し、そこに写る被写体の 3 次元位置を測定する" という作業を想定するものである。実験手順は以下のとおりである。

- (1) 任意の場所に複数の点を描いた平面 A を用意する。各点の平面上の位置をあらかじめメジャーで正確に測っておく。左上を原点とし、座標値を x_i, y_i として記録する。
- (2) 平面 A を、格子模様を描いた平面 B に重ねてカメラでの撮影を行う。
- (3) 画像上に写る点の中心位置を、サブピクセル単位で読み取り、記録する。
- (4) 平面 A を取り除く。
- (5) 記録したピクセル位置が、格子模様のどの位置を撮影しているか、撮影画像上で目視により読み取る。最小単位は 0.5 [mm] である。読み取り方は、図に示すとおり画像上で格子模様を等間隔に分割し(補助線を描く)、位置を推定する。この値を x'_i, y'_i として記録する。

以上の手順で得られた 3 次元座標値 $(x_i, y_i, 1500.0)$ が、2 でいうところの、"指定した画像上の位置に写る被写体の 3 次元位置" の真値で、 $(x'_i, y'_i, 1500.0)$ が、その読み取り値と見なすことができる。なおここでは平面を $z = 1500.0$ と設定したが、これは任意でかまわない。このときの、真値と読み取り値との偏差(誤差)を評価する。

結果を表 1 に示す。点数は 35 である。これによると、被写体が 1,000~1,500 [mm] 程度の距離にあれば、各座標値を分散 0.75×0.75 [mm²] 程度以下の精度で測定可能であることがわかる。

表 1 実験における真の値と読み取り値との偏差

	カメラと平面の距離が約 1000 mm のとき		カメラと平面の距離が約 1500 mm のとき	
	$ x - x' $	$ y - y' $	$ x - x' $	$ y - y' $
誤差平均	0.27 [mm]	0.20 [mm]	0.54 [mm]	0.39 [mm]
誤差 2 乗平均	0.44×0.44 [mm ²]	0.32×0.32 [mm ²]	0.70×0.70 [mm ²]	0.55×0.55 [mm ²]

5.2 シミュレーションによる基本手法の検証

実際のカメラの光学中心位置の真値を知ることは現実的には困難なので、シミュレーションにて本手法の検証を行う。

まず 3 で述べた、画像上の同じ場所に写る、2 つの異なる特徴点の 3 次元位置 p_j, q_j が、絶対座標として測定できる場合について検証を行う。

具体的な測定手段として想定するのは、以下のような方法である。なおカメラの仕様としては、実験システムと同じように、解像度を 640×480 [ピクセル]、視野角は水平方向で約 30° と設定した。

- (1) カメラから約 1,000 [mm] の位置に平面 1 をおき、その平面にはカメラの撮影範囲において J 個の特徴点がランダムに描かれているものとする。その位置はあらかじめ p_j として計測されているものとする。
- (2) その点の撮影画像上の位置 x_j が読み取る。
- (3) 上記平面 1 を格子模様を描いた平面 2 に置き換えると同時に、カメラを平面に対して垂直の方向に正確に 500.0 [mm] 遠ざける。
- (4) 画像上の位置 x_j に写る、平面 1 上の座標値を図 1 の方法で読み取る。この値の z 成分に 500.0 [mm] を加えたものが q_j となる。
- (5) 以上により求まる p_j, q_j ($j = 1, 2, \dots, J$) を用いて、3 で述べた方法でカメラの光学中心位置を求める。

以上の作業を計算機上でシミュレートする。

p_j は誤差無く計測できるものと仮定し、 q_j に対しては前節の結果を考慮し、分散 0.75×0.75 [mm²] の正規分布に従う誤差をランダムに加える。これによる解を求め、それと真の値との距離を評価することとした。

結果を表 2 に示す。これは 1000 回の試行を行い、そのときの平均値と分散を求めたものである。観測する特徴点数が必要最小限のとき ($J = 2$)、推定される光学中心位置は、真の値から約 1.9~9.8 [mm] ($\cong 5.86 \pm \sqrt{15.7}$) 程度ずれる、また観測点を増やせばこのずれは少なくなっていく、特徴点が 8 組あればこれが $0.9 \sim 4.0$ [mm] $\cong 2.42 \pm \sqrt{2.47}$ 程度になる。

表 2 光学中心位置の真値と推定値の距離 Δl の期待値と分散

特徴点の組数 J	2	4	8	16	32
Δl の平均 [mm]	5.86	3.47	2.42	1.67	1.18

Δl の分散 [mm ²]	15.7	5.22	2.47	1.09	0.60
-----------------------------------	------	------	------	------	------

5.3 応用方法のシミュレーションによる検証

次に図3で述べた条件、すなわち、画像上の同じ場所に写る2つの異なる特徴点の3次元位置 p_j, \bar{q}_j が、それぞれ独自の座標系で測定され、お互いの相対関係がわからない場合についての検証を行う。

これをモデル化するため、図3に示すように、平面2については、平面1に対して僅かに（各回転軸1°程度）傾け、またX, Y, Zそれぞれの方向にさらに僅かに（各、数cm程度）移動させることとした。なお、これらの量は推定時には未知として扱う。この平面によって読み取られる値を直接3次元値にしたものが \bar{q}_j である。

シミュレーションを行うにあたって、4で述べた $B^{(k)} \approx B^{(k+1)}$ の式について、誤差の存在下において何をもってこれを満たすというかが問題となるが、今回は、ある平面 Π_1 上にある I 個 ($I \geq 4$) の任意の点 c_i と、別の平面 Π_2 上にある I 個の点 c'_i の関係が $c'_i \approx B^{(k)} c_i$ を満たし、また平面 Π_2 上にある点 \tilde{c}'_i が $\tilde{c}'_i \approx B^{(k+1)} c_i$ を満たすとき、評価値 $E = \sum_{i=1}^I |\tilde{c}'_i - c'_i|^2$ が少なければ少ないほど $B^{(k)}$ と $B^{(k+1)}$ が同じ変換を示す、すなわち $B^{(k)} \approx B^{(k+1)}$ である、ということとする。

以上の考えに基づき、次の手順でシミュレーションを行った。

- (1) カメラ1の光学中心位置を適当に $b^{(1)}$ と設定し、そこから観測されるべき観測値を逆算する。これに前節と同じランダム誤差を加える。
- (2) 以上の値より $B^{(1)}$ を求める。
- (3) $b^{(1)}$ の周辺 2 cm 以内の 64000 ($40 \times 40 \times 40$) 個の異なる点 $b_n^{(2)}$ ($n = 1, 2, \dots, 64000$) にカメラ2を動かし、そこから観測されるべき観測値を逆算し、上と同様の誤差を加える。
- (4) それぞれの $b_n^{(2)}$ で得られた観測値を用い、上に述べた評価値 E を求める。これを最小にするものを探す。
- (5) このときに $B^{(1)} \approx B^{(2)}$ が最も成り立っているものと解釈し、このときの $b_n^{(2)}$ が、この推定において $b^{(1)}$ と同じと認識されたものとなる。

結果を表3に示す。ここではそれぞれ100回の試行を行った。カメラ1の光学中心位置と、それと同じ位置にあると推定されたカメラ2の、実際には誤差の影響で同じ位置となっていない、そのずれの大きさの平均値と分散を求めたものである。

4の方法と同じ観測点数で比べたとき、全般的に精度が落ちるものの、その程度はそれほど大きなものではないことがわかる。

なお、上で用いた $b^{(1)}$ や $b_n^{(2)}$ の値は、シミュレーションにおける観測値を逆算するために使っただけで、実際の運用時において、観測値からは一意には求めることができないことに注意する。

表3 光学中心位置の真値と推定値の距離 Δl の期待値と分散

特徴点の組数 J	8	16	32
Δl の平均 [mm]	2.83	2.45	1.98
Δl の分散 [mm ²]	1.75	1.64	1.29

5.4 一般的なキャリブレーション手法との比較

以上述べた方法の精度をどう評価するかは、その誤差が最終的なアプリケーションにおよぼす影響、すなわち、マルチカメラの撮影画像がどれだけシームレスに接続されるか、によって判断されなければならない。しかしそのためには、2で述べた繋ぎ合わせのための変形パラメータの算出手法や、隣り合うミラー間のエッジを除去する方法など、その他の多くの課題が解決された後に総合的に検証されなければならない、ここで独立してその評価を下すことはできない。

ここではひとつの参考として、代表的なカメラ位置推定方法である透視3点問題[Wolfe 1991]を用いて推定を行うことと比較する。透視3点問題では、焦点距離 f が既知でなければならない。よって今回、撮影画像の真の焦点距離 f が、計算時ににおいて $\varepsilon\%$ の誤差を含んだ値 f' として使われたときに、それによって生じるカメラ位置推定値の誤差を、前節と同じようなカメラ配置による、ある1つの例で計算した(表4)。なお、焦点距離以外の入力パラメータには全く誤差が含まれないものと仮定する。

これによると、事前のカメラキャリブレーションにおいて焦点距離 f が0.3%程度の誤差以下の精度で算出されているのはじめて、本手法と同程度の位置推定が可能となる。しかしこれは、カメラ内部パラメータのキャリブレーションに関する従来の報告から総合的に判断するに、一般の方法では現実的には達成が容易ではない精度であると思われる。よってこのように f だけに着目したとしても、それとは無関係に測定が行われる本手法の有効性が確認される。

表4 透視3点問題を用いたときの光学中心位置の真値と推定値の距離 Δl

f' の誤差 ε	0.1%	0.2%	0.3%	0.5%	1.0%
Δl [mm]	1.4	3.2	6.0	9.0	17.0

5.5 映像の繋ぎ合わせへの影響

表3によって示された、本測定の期待される精度が、最終的に作成される繋ぎ合わせ映像の接続性におよぼす影響を見積もる。

いま、横に並べた2台のカメラの撮影映像を接続することを考え、この2つのカメラの光学中心位置のずれがある方向に a mm あるものと仮定する。カメラの位置が a mm ずれてある被写体を撮影するという事は、カメラを不動にして、被写体が a mm カメラのずれと逆方向に移動することと等価になる。この被写体の3次元的な移動により、画像上の被写体の位置がどれだけ2次的に移動するかは、カメラから被写体までの距離に依存する。この関係をあらわしたのが図4である。この関係を、式であらわすと近似的に以下が成り立つ。

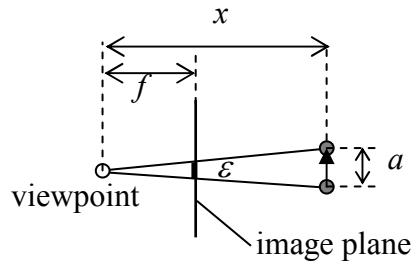


図4 被写体までの距離 z と被写体の動き a と画像上でのずれ ε の関係

$$\varepsilon \cong f \frac{a}{z} \quad (12)$$

ここで ε [pixel] は画像上でのずれ、 z [mm] は被写体までの距離、 f [pixel/mm] は焦点距離である。

ここで具体的な値を入れて計算してみる。カメラの解像度を 640×480 pixel、水平方向の視野角を 30° とすると、 $f = \frac{640}{2 \tan(30/2)} \frac{1}{z}$ だから $f \cong 1194$ である。例えば $2,000$ mm 前

方にある被写体に生じるずれ ε は、 $a = 2.0$ mm として $\varepsilon = 1.2$ pixel である。もしこれが $4,000$ mm 前方にあれば $\varepsilon = 0.6$ pixel となる。このずれ ε が、画像接続時の隣り合う映像間でのずれ量になる。これより、被写体が $2,000$ mm より近くにあるときは、画像接続時のずれが多少目立つが、それより離れるにつれ徐々に少なくなっていき、 $4,000$ mm 程度より離れるとほとんどずれは見えなくなる。よって、本手法での精度は、被写体が $2,000$ mm 程度より遠方にあるような撮影においては問題無く適用可能であり、そのような撮影は広視野角映像を前提とした撮影対象として最も一般的なものであると考える。図5に、試作した12台構成によるミラー式マルチカメラシステムと、その撮影映像を示す。



図5 12台カメラ構成によるミラー式マルチカメラシステムと撮影映像

6 マルチカメラシステムによる同期撮影

複数のカメラを用いた撮影システムにおいて、全てのカメラが同じ瞬間を常にとらえるようにするための時間的整合性の保証は非常に重要である。そこで、この時間的整合性の

保証に着目し、必要となる完全同期撮影方法の確立を目的とし、それを実現するカメラや周辺機器の選定、撮影方法について検討を行った。ここでは、32 台のカメラで一つの被写体を取り囲む撮影方法で、同期撮影実験を行った。

6.1 システム構成

大量カメラによる完全同期撮影を実現するためには、これに必要な機能を持つ特別な機器を選定する必要がある。ここで完全同期撮影とは、各カメラのシャッターを切る時刻が、完全に一致していることをいう。また本研究で対象とするのが動画撮影のために、このような撮影が秒 30 フレーム以上のレートで行われる必要がある。

多くのビデオカメラが市場に出回っているが、単体使用を目的としたものが一般的であり、ほとんどが複数台カメラの同期撮影機能をもっていない。また同期撮影機能をもっているものについても、ごく限られた少数台のカメラの同期を想定しているものであり、数十台規模の大量カメラの同期撮影となると、候補となるカメラは極端に減り、特に 30 台を超える完全同期を行った撮影実績があるカメラは、調査した範囲においては見当たらなかった。

また、大量カメラによる完全同期撮影を実施する際には、扱うデータ量が膨大になるために、システムが大規模になりがちであり、カメラの選定だけではなく、必要な周辺機器とその数量、接続方法についても、詳細な検討を行う必要がある。

6.2 32 台カメラ同期撮影のための機器構成

① カメラの選定

大量カメラによる完全同期撮影が可能なカメラの選定を行った。選定の基準は以下とした。

[必須項目] 外部同期可能、30fps 以上のフレームレート、VGA 以上の解像度、多数台カメラの同期撮影可能、Firewire, CameraLink などのデジタルインタフェース

[比較項目] 装置規模（小規模が○）、画質、同期撮影実績

本実験においては、上記項目を最も満たしている、Point Gray Research 社のデジタルカメラ、Dragonfly を使用することにした。

② 周辺機器の検討

32 台カメラによる同期撮影を行うにあたり、カメラを制御する PC、大量の画像データを保管可能であるハードディスク、使用カメラに応じた大量カメラ同期撮影に必要な機器の検討を行った。結果を表 5 に示す。

表 5 周辺機器一覧

機器名	個数	用途	備考
サーバ PC	1	<ul style="list-style-type: none"> 同期信号の発信 撮影画像の取り込み 撮影画像の確認 	メモリ 2 GB 以上
クライアント PC	4	<ul style="list-style-type: none"> 撮影画像の取り込み 	メモリ 2 GB 以上
大容量ハードディスク	5	<ul style="list-style-type: none"> データ保存 	サーバ PC : 200 GB クライアント PC : 120 GB

IEEE1394 ボード	8		カメラ 4 台に付き、1 枚必要。 30 fps を満たすためには、 2 枚/PC が最大。
Sync-Unit	8	<ul style="list-style-type: none"> カメラの撮影タイミングを示す同期信号を発信 Sync-Unit 間の同期をとる 	
Firewire HUB(4port)	16	<ul style="list-style-type: none"> 複数の IEEE1394 ケーブルを集線 	
Gigabit HUB	1	<ul style="list-style-type: none"> 各 PC 間のデータのやり取りを行う 	
CS マウントカメラレンズ (4mm&6mm)	32		撮影範囲に応じて、4 mm レンズと 6 mm レンズを使い分ける。4 mm レンズの方が広視野を撮影可能である。
LAN ケーブル	5		
IEEE1394 ケーブル			

③ カメラの設置位置

被写体をあらゆる角度から見ることを可能とするため、一つの被写体を取り囲むようにカメラを設置し、撮影を行うことにする。カメラを円状に設置し、さらに隣接するカメラ間の距離や円の中心とのなす角度を均等にした。

④ カメラと被写体の距離

被写体とカメラとの距離を検討する。被写体が全てのカメラの撮影範囲内に収まるよう、距離の調整を行う必要がある。装置が大掛かりになるため、撮影ごとに被写体とカメラとの距離を変更するのは、困難である。そこで、焦点距離の異なるレンズを付け替えることにより撮影範囲の調整に対応する。

⑤ クロマキースクリーンの設置

被写体領域の抽出が容易となるよう、被写体背後にクロマキースクリーンを設置する。本撮影においては、被写体の全周囲をカメラで取り囲んでいるため、全てのカメラにおいてクロマキースクリーンが被写体の背後に来るよう考慮しなくてはならない。また、スクリーンの皺による影は被写体領域抽出精度を落とすことになるため、スクリーンを立体裁断することにより皺を低減することが望ましい。

⑥ 照明の設置

どの方角から見ても被写体への照射条件を同一とするためのライティングは非常に重要である。被写体自身への影、スクリーンへの影が生じないように、照明を全方向からまんべんなく当てて調整を行う。これにより、全てのカメラにおける被写体の明るさに大きな差異が生じることはなく、また、撮影画像における被写体抽出が容易になる。

また、カメラの周波数と照明の周波数が異なると撮影画像にフリッカが生じるため、照明に電源を供給する電源ボックスは、電流を交流から直流に変換する機能を備えるものを使用することが望ましい。

⑦ 32 台同期撮影のための機器構成図

Dragonfly を用いた複数カメラの同期撮影におけるシステム構成を図 6 に示す。

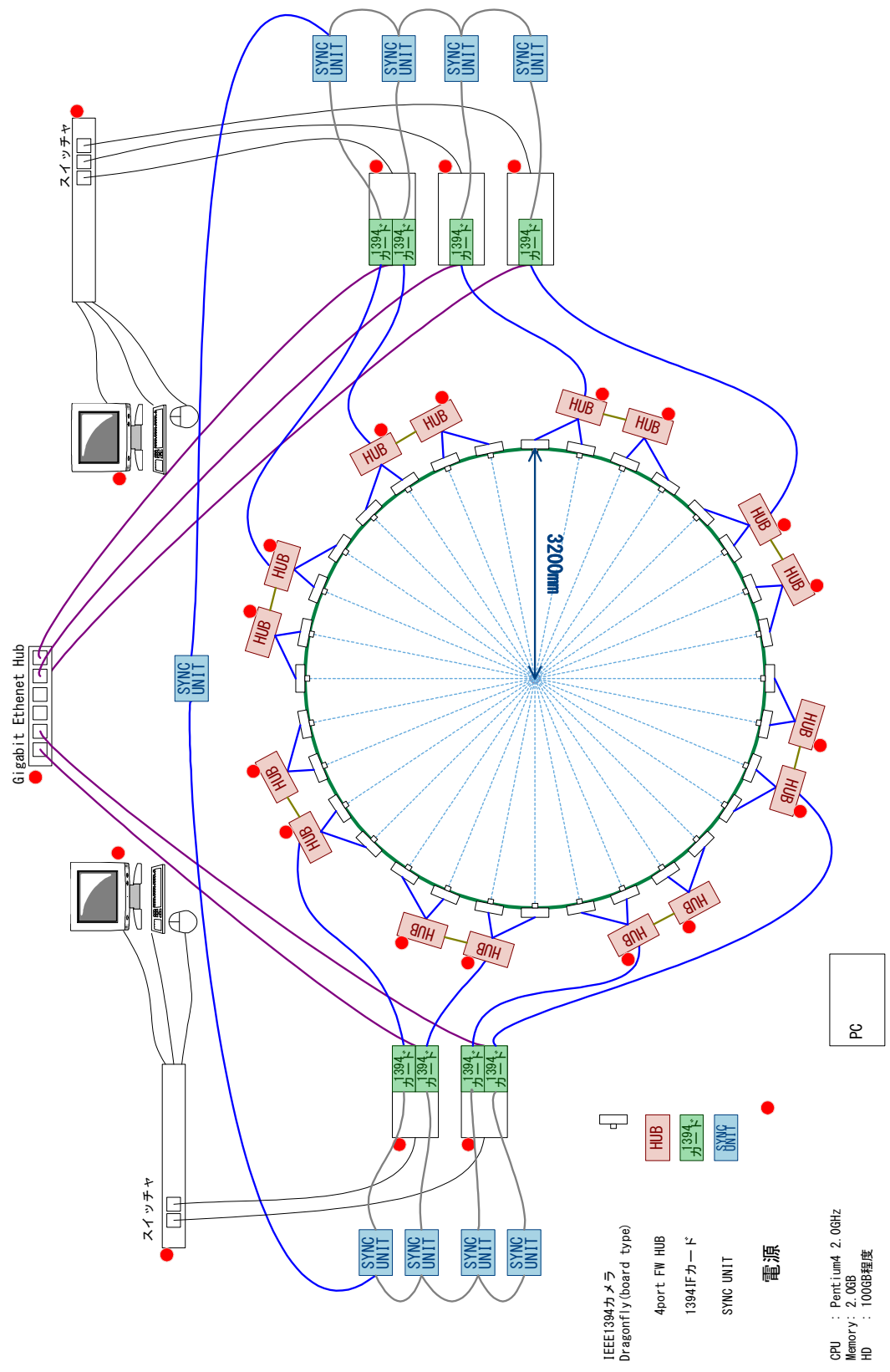


図 6 32 台同期撮影のための機器構成図

7.3 異視点画像を用いたアプリケーション

本実験において撮影した映像コンテンツの1つであるゴルフショットを用いたアプリケーションの一例を示す。



(1) 32 台カメラ映像同時表示

(2) 任意時刻・任意視点画像のインタラクティブ表示



(3) スイング軌跡の抽出・表示



(4) 中間画像作成

図7 異視点同期撮影画像を用いたアプリケーション例

図7(1)は、32台すべての撮影映像を1画面上に同時に表示した映像である。

図7(2)は、ジョイスティックやキーボードを入力装置とし、ユーザの入力に応じて視点位置移動表示したり、時間移動表示したり、視点時間移動表示したりするものである。撮影画像の同期が完全にとれていない場合、視点位置移動時に被写体にがたつきが生じるが、完全同期撮影においては、スムーズな映像の切り替えが可能である。

図7(3)は、ゴルフスイングにおける、カメラ方向、及び、時間方向のクラブヘッドの3次元的な軌跡を自動的に抽出し、表示するものである。

図7(4)は、撮影画像を基に作成した中間画像である。これにより、視点位置解像度及び時間解像度の向上が可能となる。

7 まとめ

ミラー式マルチカメラシステムにおける各カメラ位置の調整を目的とした、光学中心位置一致判別方法を提案した。本方法は、一般のカメラキャリブレーションに比べ、推定するパラメータの数が少なく、また推定に必要な既知パラメータの数も極めて少ないため、

安定に且つ高精度の推定が可能である。4で述べた方法はその基本的な考え方であり、実際の測定は、その実施容易性から5で述べた方法を用いるのが現実的と考える。

(1)では、位置一致判別手法の提供までを述べた。実際の位置合わせでは、こう見えるべきだ、という位置にカメラのXYZステージを移動する作業が必要となるが、これはビジュアルサーボの考え方で実施できる。その際に、2つの被写体平面の手前の平面を透明ガラスで作成することで両方を同時に撮影できるようにし、あらかじめ計算された位置に特徴点や特徴線を描いておくなどの工夫を行うことにより、特徴点の3次元座標値を逐次読み取る必要はなくなり、実際の作業量はそれほど大きくなっていくものと予想される。その詳細検討ならびに、撮影した映像の接続のための映像変換方法の具体方法の開発が今後の課題である。

また、マルチカメラに撮影において非常に重要となる同期保証について、32台もの大量のカメラで完全同期撮影を可能とする、機器の選定やシステム構成の検討、アプリケーション作成による効果の確認を行った。十分な同期精度を得られることを確認した。

(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

本研究では移動カメラによる広域・高解像度映像の自動撮影に関する基礎技術として、移動カメラの自己位置推定技術の開発に取り組んだ。

移動カメラを未知環境にて自律的に移動させるためには、環境の幾何情報を取得し、幾何情報をもとに移動カメラの自己位置を推定し、幾何情報を地図として記録することが必要となる。これを踏まえ、本研究では移動カメラの台車部分となる車輪型移動ロボットを用い、これに環境中における障害物までの距離・方位の計測が可能なレーザー距離センサを搭載することで、ロボットが環境中において自己位置を推定しながら移動した範囲の環境の地図を生成するロボットシステムに取り組んだ。これについて以下に述べる。

1 移動カメラのための自己位置推定技術と環境地図生成技術

本研究では広域を移動しながら自動撮影を行う移動カメラの実現を目的として、移動カメラが自己位置推定を行いながら、周辺の環境の地図を生成する技術の開発を実施した。より具体的には、車輪型移動ロボットに搭載されたレーザー距離センサにより水平面のスキャンを行い、1回のスキャン毎に得られる複数の各計測点に対して特徴を設け、複数回のスキャン間で特徴の照合を行うことで、自己に対する環境の相対的な位置・姿勢の変化量を求め、これを自己位置の推定に用い、この自己位置をもとに移動カメラ周辺の環境を地図として表すものである。以下にその詳細を述べる。

1.1 レーザー距離センサによる自己位置推定

レーザー距離センサによる自己位置推定手法について述べる。全体の処理の流れとしてはレーザー距離センサによりスキャンを行い、前回に行ったスキャンとの差分の情報よりスキャンによる計測点の点列間の相対的な回転・並進を求め、これをロボットの相対的な移動量として初期位置から逐次的に積算することにより、世界座標系におけるロボットの位置・姿勢を推定する。詳しい手順は以下ようになる。

1. 水平に φ の範囲をスキャンし、計測点 $S[i]$ を記録(ただし $S[i]=[\psi[i], d[i]^T]$, ψ : 角度, d : 距離)

2. $S[i]$ と $S[i-1]$ 間で各計測点間の距離 d の差分の二乗和を最小とする $\phi[i]$ を算出
3. $S[i]$ を θ 回転し、 $S[i]$ と $S[i-1]$ のそれぞれにおいて、2つの計測点からなる座標系を設定し、 $S[i]$ と $S[i-1]$ で設定された座標系が重なるような計測点の共通な座標系に配置での相対位置が一致する点に対応するときの $S[i]$ と $S[i-1]$ 間の並進成分 X_t, Y_t からなる投票空間に投票
4. 並進成分の投票空間においてピークを検出し、ピークに対応する並進成分 $X_t[i], Y_t[i]$ を記録
5. 1~4 で得られた $\phi[i], X_t[i], Y_t[i]$ を 2 回のスキャン間におけるロボットの相対移動量 $Pr[i]=[\phi[i], X_t[i], Y_t[i]]^T$ として、世界座標系におけるロボットの推定位置 $P[i]=P[i-1]+Pr[i]$ を算出

以上により、ロボットの自己位置推定が行われる。

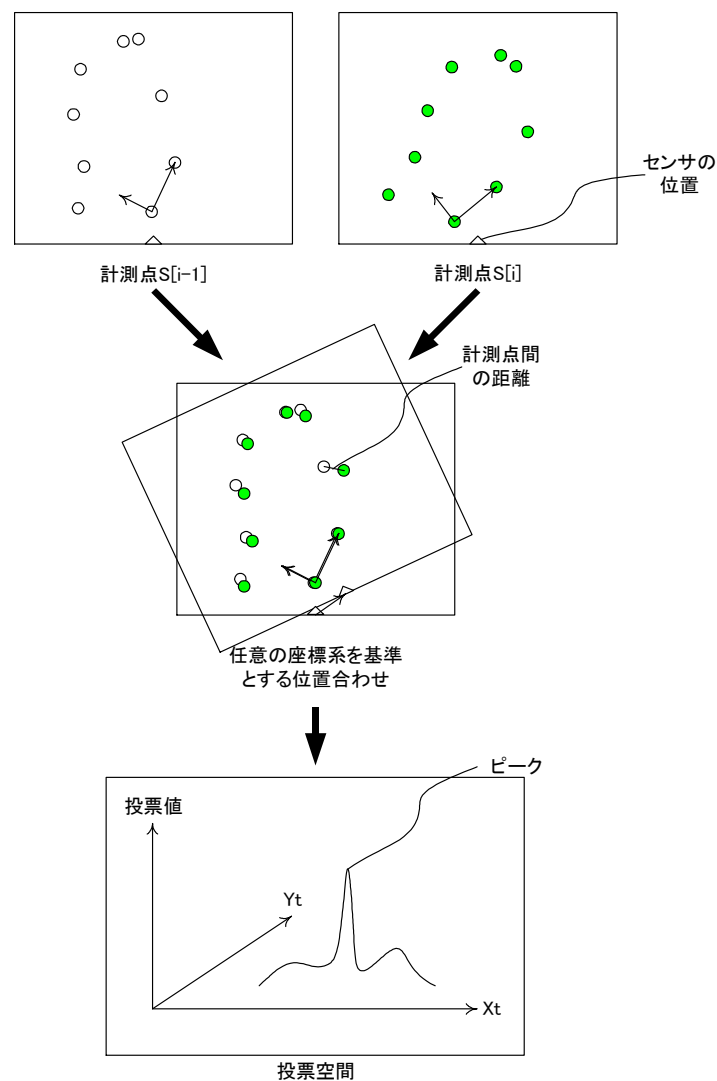


図8 投票による並進移動量の算出

手順3に関する補足を図8を用いて説明する。レーザー距離センサの最新の計測値 $S[i]$ 、1回前の計測値 $S[i-1]$ とすると、計測点は図8上段の図のように計測されたとする。これらの計測値より任意の2つの座標系を計測値 $S[i]$ と $S[i-1]$ のそれぞれで設定し、これらの座標

系が重なるように位置合わせを行い、 $S[i]$ と $S[i-1]$ の計測点を共通の座標系で表す（図 8 中段）。このとき、座標系の設定に用いた計測点以外の点のそれぞれについての距離を算出し、距離が閾値以下のときはセンサの相対位置を算出して対応する投票空間に投票する（図 8 下段）。これを $S[i]$ と $S[i-1]$ の全ての計測点の組み合わせによる座標系で繰り返す。これにより投票空間にピークが現れるため、このピークに相当する X_t と Y_t をスキャン間におけるロボットの並進移動量として記録する。

1.2 環境地図の生成

前節にて推定された自己位置をもとに環境地図を生成する。ここでは[根岸 2003]を参考としてグリッドの状態を設定する。ただし、[根岸 2003]ではグリッドに確率的に状態が更新されていくのに対して、ここでは簡単のため 3 値により状態を設定することで地図を表現する。具体的には以下の手順により生成される。

1. センサ座標系で得られた計測点 $S[i]$ をロボットの推定位置 $P[i]$ に沿って世界座標系上に変換し、計測点 $Sw[i]$ を算出
2. 世界座標系に対して設けられたグリッドにおいて、計測点 $S[i]$ に対応するグリッドを占有グリッドとし、占有グリッドの情報をもとに非占有グリッドと未知グリッドを設定することにより環境地図を画像として生成

2 について補足する。環境地図は世界座標系を基準としたグリッドの画像として生成される。環境地図の生成においては、スキャンにより得られたセンサ座標系での計測点を世界座標系でのロボットの位置・姿勢を基準とする座標変換を行うことで、スキャンデータを世界座標系で表現し、これをもとに環境における障害物の有無を示すグリッドの状態を色により表現することにより環境地図を画像として表現する。

グリッドは、1) 占有、2) 非占有、3) 未知、の 3 つの状態のいずれかが割り当てられる。状態の割り当てにおいては、まずレーザー距離センサにより障害物が確認された方位・距離に相当するグリッドが占有グリッドとして設定される。次に、センサと占有グリッドを結ぶ直線上にあり、占有グリッドよりセンサ側にあるグリッドには障害物が存在しないものとして非占有グリッドが設定される。またセンサと占有グリッドを結ぶ直線上にあり、占有グリッドからセンサに対して離れる方向にあるグリッドには障害物が存在するか否か未知であるものとして未知グリッドが設定され、以上により環境地図が生成される。

1.3 評価実験

提案手法の評価を目的として実験を行った。実験システムの構成と実験結果について以下に述べる。

1.3.1 システム構成

本実験システムは車輪型移動ロボット、レーザー距離センサ、制御用 PC から構成されている（図 9）。

レーザー距離センサは一回の計測により 0.5 度毎に 180 度の水平面の走査を行い、走査した範囲にある障害物に対する方位と距離の計測が可能である。なお、最長計測距離は 8 mである。



図9 レーザー距離センサ搭載移動ロボットの外観

1.3.2 実験

本実験では提案した手法を用いてオフィス環境の部分的な地図を生成するものとした。人間がジョイスティックを介してロボットを操作することにより目的地まで移動させ、並行してロボットは移動しながら自己位置推定を行い、環境地図を生成した。生成した環境地図を図10に示す。環境地図は約16 m×約12 mの範囲で生成した。同図では赤の点が占有グリッド、緑の点が非占有グリッド、水色の点が未知グリッドとして表されている。青い枠は物体が存在するおよその領域を示している。また×印はロボットの推定位置を示しており、ロボットの走行距離は約5 mである。

実際のオフィス環境に概ね合致した地図が生成されているが、壁面などの平面の観測時に占有グリッドが同一直線状に並ぶべきところが帯状に散らばってしまうなど、位置推定誤差に伴う地図の精度低下が見られた。この位置推定誤差はスキャンデータ間の照合における誤った照合に起因すると考えられる。

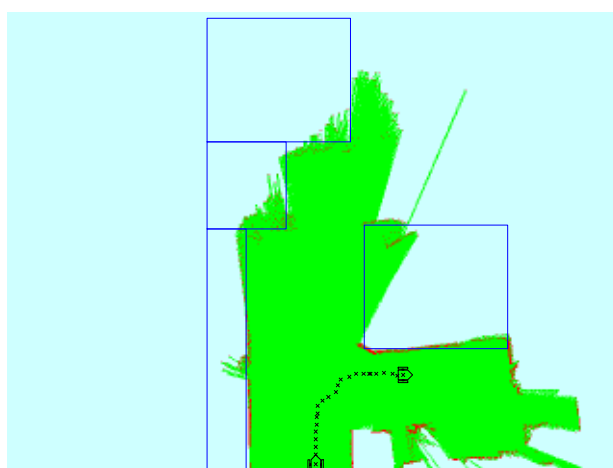


図10 ロボットの推定位置と生成された環境地図

2 移動カメラによる広域・高精細映像生成

2.1 概要

本研究では、自動走行ロボットにカメラを取り付け、広範囲にわたる映像を連続的に異地点から撮影し、これらを繋ぎ合わせることで広域を対象とした超高精細・静止画像の取得を可能にするための技術開発に取り組んだ。

2.2 システム構成と画像モザイクキング

カメラの移動制御のために用いるロボットは、前後方向の移動と、その場での回転ができる。撮影は 640×480 [pixel] の撮影解像度を持つ CCD カメラに、のズームレンズを取り付けて行う (図 11)。カメラはロボットの車輪の向きと垂直に取り付ける。



図 11 撮影システム

画像モザイクキングは、複数の映像をシームレスに接合して広視野の映像を作成する技術である。広視野映像を人間が見ると、個々の映像を見る場合に比べて被写体の位置関係を把握しやすく、高い臨場感を得る。そのためバーチャルリアリティや、監視などへの応用が期待されている。

本研究では、移動ロボットにカメラを搭載して屋内環境のモザイク映像を作成するシステムの開発を行った。ロボットに搭載されたカメラは環境に固定されたカメラより自由度が高く、また撮影の自動化が可能である。ただし一方でロボットが自由に撮影を行うと映像の自由度も高くなり、モザイクキングが難しくなる。モザイク映像の元となる映像間に強い幾何的な関係があれば、比較的簡潔なアルゴリズムで高速にモザイクキングできるため、ロボットはそうのように映像を撮影することが望ましい。そこで本研究では、棚や机に対して平行にロボットが移動して棚や机上の静物の映像を得るという課題について取り組んだ。各映像の撮影の際、棚板の縁に対して垂直にカメラが向いていればモザイクキングが簡単になるが、撮影用の視野の狭いカメラで多くのエッジの中から棚板の縁を特定する必要がある。そこでハフ変換を拡張し、複数のカメラ映像から棚や机の縁を認識する手法を提案、実装し、本棚の撮影に適用した。

2.3 ロボットの構成と問題設定

二輪の移動ロボットを用いて撮影を行う。このロボットはロボット座標系において、x 軸方向の前後への移動と、原点を中心とした xy 平面上での回転が可能である。カメラは y 軸と光軸が平行となるように設置されている。またカメラ座標系の y 軸とロボット座標系の z 軸は同一直線上に存在する。本稿ではカメラを予め適当な高さで固定し、一つの柵の一段分のみの撮影を扱う。このロボットが自律的に、柵の縁を基準に柵に沿って平行移動しながら撮影を行う。映像を撮影する度に画像平面を全て L と平行 ($\theta = 0$) にすることを目指す。それにより、映像の回転を考慮しないでモザイクキングできる。

2.4 柵板の縁の認識

机や柵の縁の特徴は、(1)比較的他のエッジよりも長い、(2)地面に対して水平、である。屋内環境においては、壁の模様や他の特別な場合を除いて他の物体とは区別しやすい。しかし撮影用のカメラの視野は限られるため、一つの映像でエッジの長短を区別できない。この問題に対し、次のようにハフ変換を拡張して、複数の映像から最長のエッジを検出するアルゴリズムを提案する。

まず、画像座標系上の直線 $y = ax + \beta$ を $\alpha \beta$ 平面中の点(α, β)に変換するハフ変換を考える。地面と水平な直線 L は映像上で水平になるとは限らないので、地面と水平でない他の雑音エッジ X と区別がつかない (図 12 (a))。しかし、画像座標系において L は地面と水平という条件から

$$y = x(h/l)\sin\theta + (fh/l)\cos\theta \quad (13)$$

という直線となる。ここで h, l, f はそれぞれ L からカメラまでの高さ、L とロボットの回転軸との距離、焦点距離である。したがって L のハフ空間への写像は楕円

$$\alpha^2 + (\beta/f)^2 = (h/l)^2 \quad (14)$$

上に存在する (図 12 (b))。そこで、ロボットをその場で回転させて得られた映像について次の処理をする。

- (1) パラメータ α, β を離散化して 2 次元配列を 2 つ ($A_1 A_2$ と呼ぶ) 用意し、各要素を 0 に初期化
- (2) 各映像について以下を繰り返す
 - ① A_1 の各要素を 0 に初期化
 - ② A_1 に各エッジ点をハフ空間中の直線へ写像した直線の通る要素の値をエッジ強度の分だけ増加 (投票)
 - ③ A_2 の各要素に A_1 の各要素値の n 乗を足す
- (3) パラメータ h/l ($= \gamma$ とする) を離散化して 1 次元配列 (B と呼ぶ) を用意し各要素

を 0 に初期化

(4) A_2 の要素ごとに γ を計算し、 B に要素値を投票

このようにすると B において、図 12 (c) のように、 L に対応する γ 値周辺における投票値が大きくなる。上記の処理の後に新たに映像が得られた後、上記の 1 次元配列の値で投票に重み付けしてハフ変換すると、ハフ空間中で点 pL の存在しうる領域以外の雑音エッジが抽出されにくくなる。

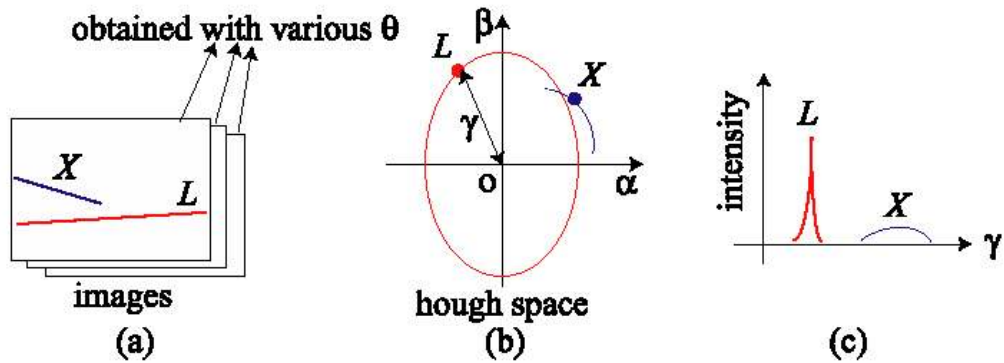


図 12 ハフ空間での特徴

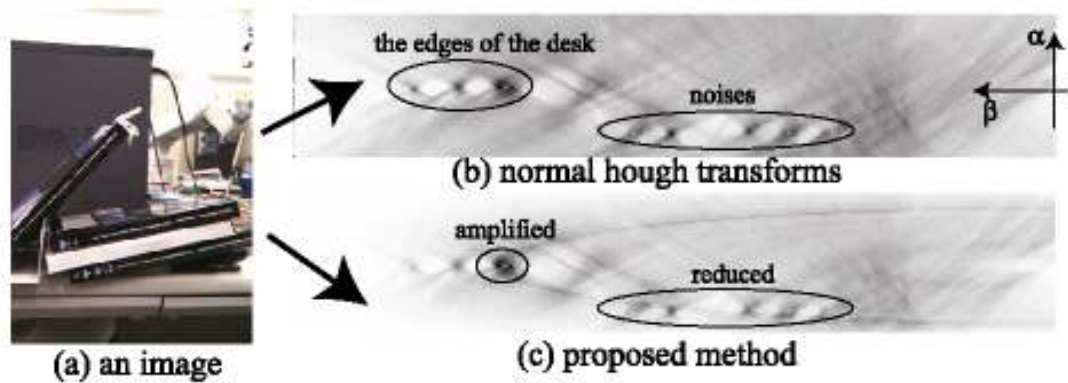


図 13 ハフ変換の結果

2.5 実験

2.5.1 机の縁の抽出例

映像のように机の上に本を置いてエッジ抽出を行った。ロボットに回転させて机を観測させた後、通常の手法と提案手法によりハフ変換を行った。その結果、通常の手法（図 13 (b)）よりも、提案手法（図 13 (c)）の方が、本によるエッジよりも机の縁が強く写像された。

2.5.2 モザイク映像作成

実際に本の映像を撮影した。机を本棚に見立てて本を 1.5 [m] 並べ、ロボットを適当な距離に置き、以下のように自律的に行動させた。

- (1) その場で回転
- (2) 抽出したエッジにカメラを向ける
- (3) 棚の端に移動
- (4) 100 [mm]ごとに前進し、エッジが映像中で水平になるようにカメラの位置を調整し映像を撮影
- (5) 棚の端に達したら撮影終了

得られた映像を図 14 に示す。通常ハフ変換で棚板のエッジ抽出を行って撮影を行う方法では、本と机の接する部分をエッジ抽出してロボットの向きが不適當になる現象が見られたが、提案したエッジ抽出方法では、重み付けによって一度抽出した棚板のエッジが抽出されやすいため、そのような現象は見られなかった。



図 14 自動撮影・繋ぎ合わせにより作成された映像

2.6 まとめ

棚や机のエッジを抽出するためのアルゴリズムを提案し、モザイク映像を自動撮影するロボットに実装し有効性を確認した。今後はこの抽出アルゴリズムを定量的に評価すると共に、ロボットが複数の棚や机のある環境で移動して映像を得るための行動の実装と、異なる高さの棚の撮影を行うためのハードウェア開発を目指す。

課題ア-3 幾何学的整合性を考慮した映像編集処理技術

マルチプロジェクタによる IPD 映像表示環境を対象にして、そのスクリーン形状やプロジェクタ構成の違いなどの特殊性に対応するため、映像制作や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し、管理する方法について検討を行い、試作した映像編集システムに実装した。また本映像編集システムを用い、実写コンテンツ及びCGコンテンツ制作実験を行った。

(1) 映像編集情報管理方式に関する研究

1 IPD 映像表示環境

IPD 映像表示環境（プロジェクタアレイシステム（図1））は、複数のプロジェクタをアレイ状に並べ、分割した映像をそれぞれのプロジェクタから投影し、これをシームレスにつなぎ合わせることで高精細・広視野角の映像を表示するものである。プロジェクタの配置を変えることにより様々なスクリーン形状に対応することが可能である。

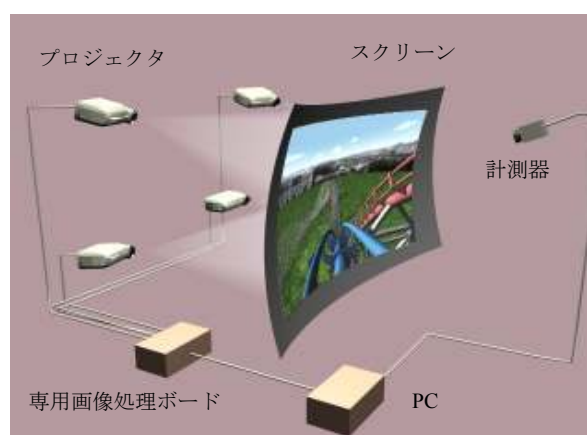


図1 プロジェクタアレイシステム

IPD 表示環境において、違和感の無いリアルな映像を表現するためには、図2に示すように、映像撮影時のカメラ位置と被写体の関係が観察者の理想視点位置とスクリーン上の表示位置との関係に一致するように撮影・投影し、現実世界と表示映像間の幾何学的な整合性を保障する必要がある。プロジェクタアレイシステムでは、各プロジェクタに理想視点から見た投影映像を供給するだけで上記の原理に基づく映像を表示することができる。すなわち、各プロジェクタの設置位置の誤差や、プロジェクタの固体差などの影響は、内部の専用画像処理ボードが自動的に行うので、供給する画像の作成にあたってはこれらの誤差を考慮する必要がない。

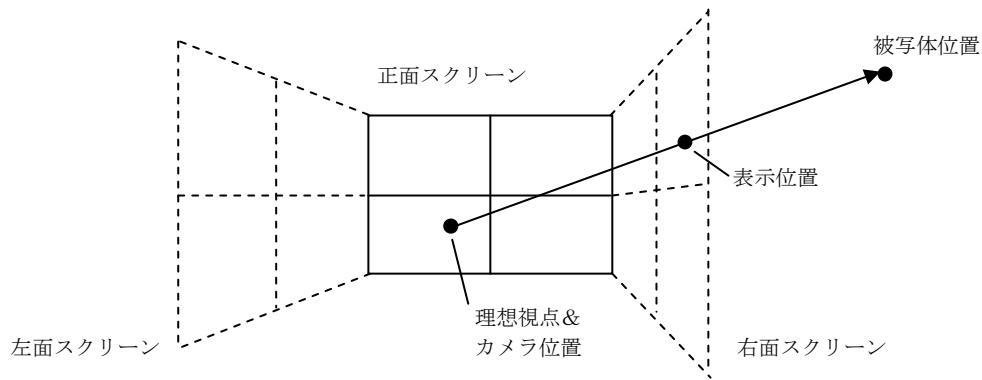


図2 表示環境と表示映像の理想的な幾何学的関係

2 画像処理フローの仕様

本システムにおいては、画像処理フローを用いて入力映像から出力映像へ至る画像処理の流れを定義する。そのため、IPD 映像表示環境の特殊性に対応するために必要となる処理も、画像処理フロー中で使用可能な画像処理として作成する。なお、この画像処理を行う一つの単位を、画像処理ノードと呼ぶことにする。

以下、画像処理フローの仕様を示す。

- ① 画像処理ノードは、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。入力ポートと出力ポートには、作成された順に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3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。以後、画像処理を図3のように示す。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。

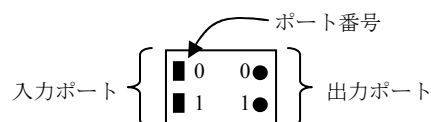


図3 画像処理の入力ポートと出力ポート

- ② ある画像処理ノード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4に示すように、AとBの2つの画像処理があった場合、画像処理ノードAの出力ポートと画像処理ノードBの入力ポートを接続することにより、画像処理ノードAの実行後にその出力画像が画像処理ノードBで処理されるようになる。

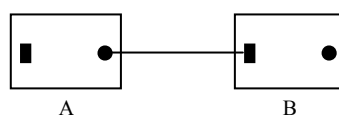


図4 画像処理ノードの接続

- ③ 画像処理フロー中で使用する画像処理は、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ノード（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ノード（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のノード（画像編集処理）である。
- 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、1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5）。

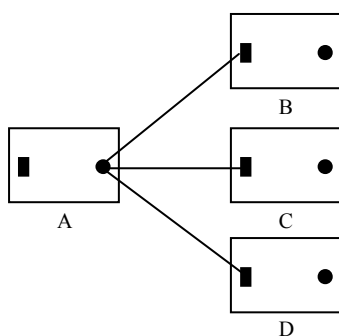


図5 画像処理ノードの接続例

- ⑥ 各画像処理ノードのポート数として、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、画像処理設計者によって決められた数のポート数を持つ。
- 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 画像処理ノードには、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は、整数型、倍精度型、色型、文字列型の4つである。
- ⑨ パラメータの値は、オペレータが設定する。ただし、オペレータに明示的に設定されないフレームの値は補間により算出する。補間には、図6に示すようなConstant, Step, Linear, Splineの4つを設ける。
- 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、使用可能とする。

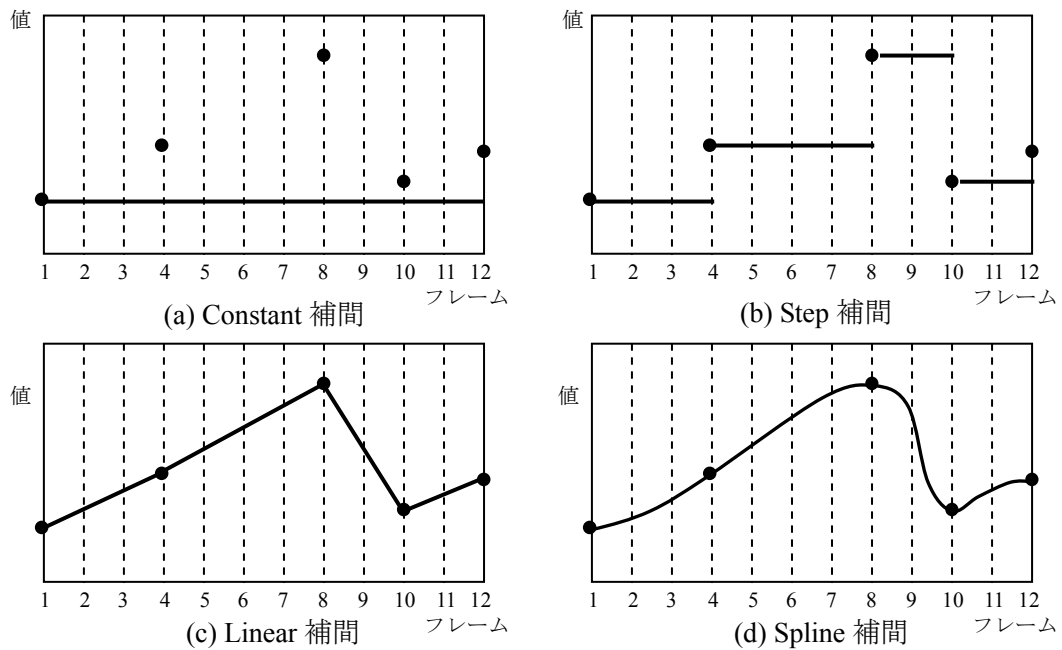


図6 パラメータの補間モード

3 IPD 映像表示のために必要な画像処理

スクリーン上の広い範囲に影響する映像効果と、狭い範囲に影響する映像効果を効率的に付与することができるように、本システムではスクリーンの視野角を考慮して、基準となる射影モデルを選択し、全ての素材画像をその射影モデルおよびカメラパラメータに合わせて編集する。そして、このように作成した画像を分割することによって、各プロジェクトが投影すべき画像を作成する (図7)。

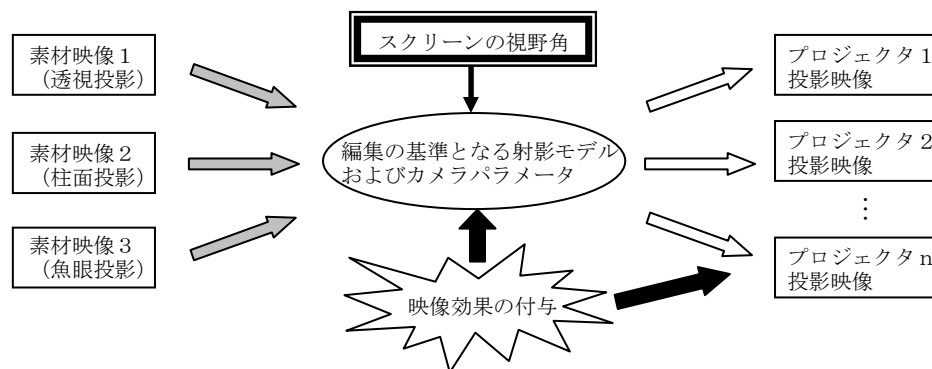


図7 本システムにおけるIPD映像編集

素材画像を基準となる射影モデルおよびカメラパラメータに統一した画像に対して映像効果を付与することによって、複数のプロジェクトから投影される複数の画像に影響するような映像効果を付けることができる。また、1台のプロジェクトが投影する画像に対してだけ映像効果を付けたい場合には、各プロジェクトが投影する画像に分割した後の画像

や単位球面上の画素を入力画像上の画素に変換する処理（図 8-②）を作成していく方法が挙げられる。以下、この方法を方法 1 と呼ぶことにする。この場合、ペースト処理をシステムで取扱可能な全ての射影モデルに対応させるには、9 個の画像処理を作成しなければならない。これは、背景画像と前景画像の射影モデルとして、それぞれ透視投影、柱面投影、魚眼投影の 3 つがあるため、その組み合わせの数の画像処理が必要になるということである。

また、この場合、システムで取扱可能な射影モデルを増やすときにも多くの画像処理を追加しなければならない。例えば、本システムで取扱可能な射影モデルを 1 つ追加することを考えると、上記ペースト処理をシステムで取扱可能な全ての射影モデルに対応させるためには、背景画像に 4 通り、前景画像に 4 通りの射影モデルがあるため、全部で 16 個の画像処理を作成する必要がある。新たな射影モデルを追加する前に 9 個の画像処理が既に作成されているため、新規に 7 個の画像処理を追加する必要があることになる。このようにシステムで取扱可能な射影モデルが増えるにつれて、新規に追加しなければならない画像処理も多くなる。

以上のように、方法 1 で画像処理を作成することは、画像処理開発者の負担を大きくしてしまう。また、類似の画像処理を増やすことになるため、オペレータの混乱を招く原因にもなる。

次に、出力画像の画素を単位球面画像の画素に変換する変換と、単位球面画像の画素を入力画像の画素に変換する変換を DLL（幾何変換 DLL）として作成し、画像処理アルゴリズム中で行う変換処理を、幾何変換 DLL に行わせる方法を考える。以下この方法を、方法 2 と呼ぶことにする。例えば、システムで取扱可能な射影モデルを透視投影、柱面投影、魚眼投影の 3 つとした場合、透視投影画像の画素から単位球面画像の画素への変換、柱面投影画像の画素から単位球面画像の画素への変換、魚眼投影画像の画素から単位球面画像の画素への変換を行う 3 つの幾何変換 DLL（幾何変換 DLL1-3）を作成する。また、単位球面画像の画素から透視投影画像の画素への変換、単位球面画像の画素から柱面投影画像の画素への変換、単位球面画像の画素から魚眼投影画像の画素への変換を行う 3 つの幾何変換 DLL（幾何変換 DLL4-6）を作成する。幾何変換 DLL1-3 に属する 3 つの変換には、これを制御するために共通の関数を持たせる。同様に幾何変換 DLL4-6 に属する 3 つの変換にも、これを制御するために共通の関数を持たせる。そして、図 8 に示したアルゴリズムの例で例えば、①は幾何変換 DLL1-3 の共通関数を使用し、②は幾何変換 DLL4-6 の共通関数を使用して処理するように記述するのである。

この方法で画像処理を作成した場合、使用する変換 DLL の実態を交換することで、1 つの画像処理を様々な投影形式の画像に対応させることができる。そのため、画像処理開発者が作成しなければならない画像処理数を最小限に抑えられる。また、システムで取扱可能な射影モデルを追加する場合には、その射影モデルの画像を処理するときに用いる幾何変換 DLL を追加するだけで、方法 2 で作成した全ての画像処理を新たな射影モデルに対応させることができる。

前述したプロジェクタ分割処理と合成処理を透視投影、柱面投影、魚眼投影の画像に対応させる場合、開発しなければならない画像処理数を比較すると表 1 のようになる。また、

方法 2 では、全ての画像処理で共通に使用できるため、一度作成してしまえばシステムで取扱可能な射影モデルを追加しない限り新規開発の必要がない。そのため、本システムでは可能な限り方法 2 で作成することにする。

表 1 開発画像処理数の比較

画像処理名	開発画像処理数	
	方法 1	方法 2
プロジェクタ分割処理	3 個	1 個
合成処理	9 個	1 個
合計	12 個	2 個

6 画像処理フロー分割部の自動生成

IPD 用のコンテンツ制作に用いられる画像処理フローの例を図 9 に示す。通常、IPD 用のコンテンツ制作で使用される画像処理フローは、大きく入力部、編集部、分割部の 3 つに分類することができる。入力部は、その画像処理フローで使用される入力画像データを読み込む部分である。編集部は、入力部で読み込んだ入力画像の射影モデルをスクリーン形状に応じて変換したり、変換後の画像を合成することによって、スクリーンに表示したい画像を作成する部分である。分割部は、編集部で作成された画像を分割し、プロジェクタアレイの各プロジェクタに供給する画像を作成する部分である。

ここで、入力画像はコンテンツごとに変わり、編集部で使用される画像処理もコンテンツのシナリオや設計書、スクリーンの形状に応じて変わる。そのため、入力部や編集部はコンテンツごと、シーンごとに異なるのが普通である。しかし、これまで我々が行ってきたコンテンツ制作を分析した結果、分割部においては、図 10、図 11 に示した 2 つの処理が頻繁に使用されることがわかった。

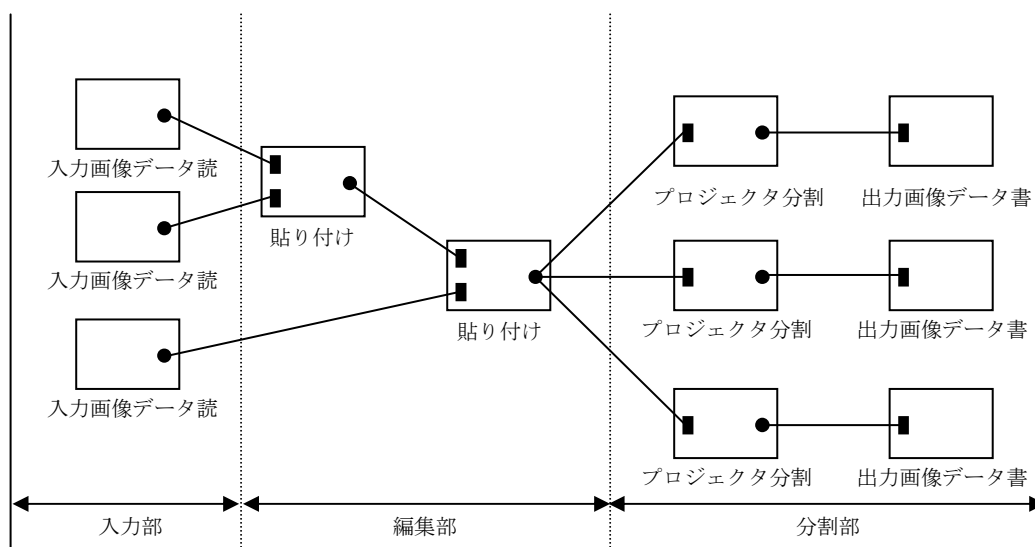


図 9 画像処理フローの例

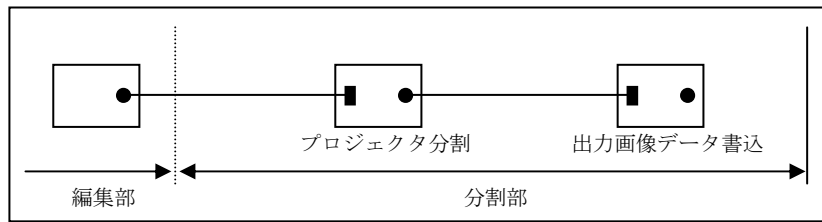


図 10 画像処理フローの分割部で頻繁に使用される処理（例 1）

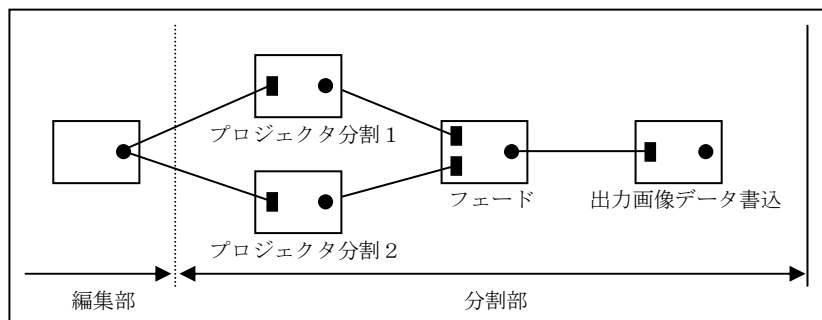


図 11 画像処理フローの分割部で頻繁に使用される処理（例 2）

図 10 は、編集部で作成された柱面投影の画像からあるプロジェクタが投影する画像を作成する場合などに使用されるものであり、これまで我々が行ってきた IPD 用のコンテンツ制作で最も使用頻度が高かったものである。

また図 11 は、編集部で作成された透視投影画像を、あるシーンでは正確な透視投影画像として扱い、他のシーンでは仮想的に柱面投影画像として扱うことにより、スクリーンに表示される領域を制御したい場合などに使用されるものである。編集部で作成された画像を正確な透視投影画像として扱う部分と柱面投影画像として扱う部分を滑らかにつなぎ合わせるために、前景画像の透明度を制御することによって、背景画像と前景画像を合成するフェード処理を用いている。

図 10 や図 11 に示す処理は、1 台のプロジェクタに与える画像を作成するためのものである。そのため、画像処理フローの分割部には、プロジェクタアレイを構成するプロジェクタの台数分だけ図 10 や図 11 に示す処理が必要となる。

プロジェクタ分割処理は、入力画像のカメラパラメータに関する設定と出力画像のカメラパラメータに関する設定を持つ。そして、入力画像側の設定は、編集部で作成される画像の射影モデルおよびカメラパラメータに依存して変化するが、出力側の設定は、プロジェクタアレイのスクリーンと各プロジェクタの配置などから決まる設備固有の値であるため、ある設備を想定した場合不変である。そのため、あるプロジェクタアレイシステムを対象としたコンテンツを制作する場合には、常に同じ設定値が使用されることになる。

そのため、本システムでは、各プロジェクタに供給する透視投影画像のパラメータ設定を記述した設定ファイルから、分割部を自動生成する（図 12）。これにより、オペレータが何度も同じ画像処理を並べたり、同じパラメータを設定したりする必要がなくなる。

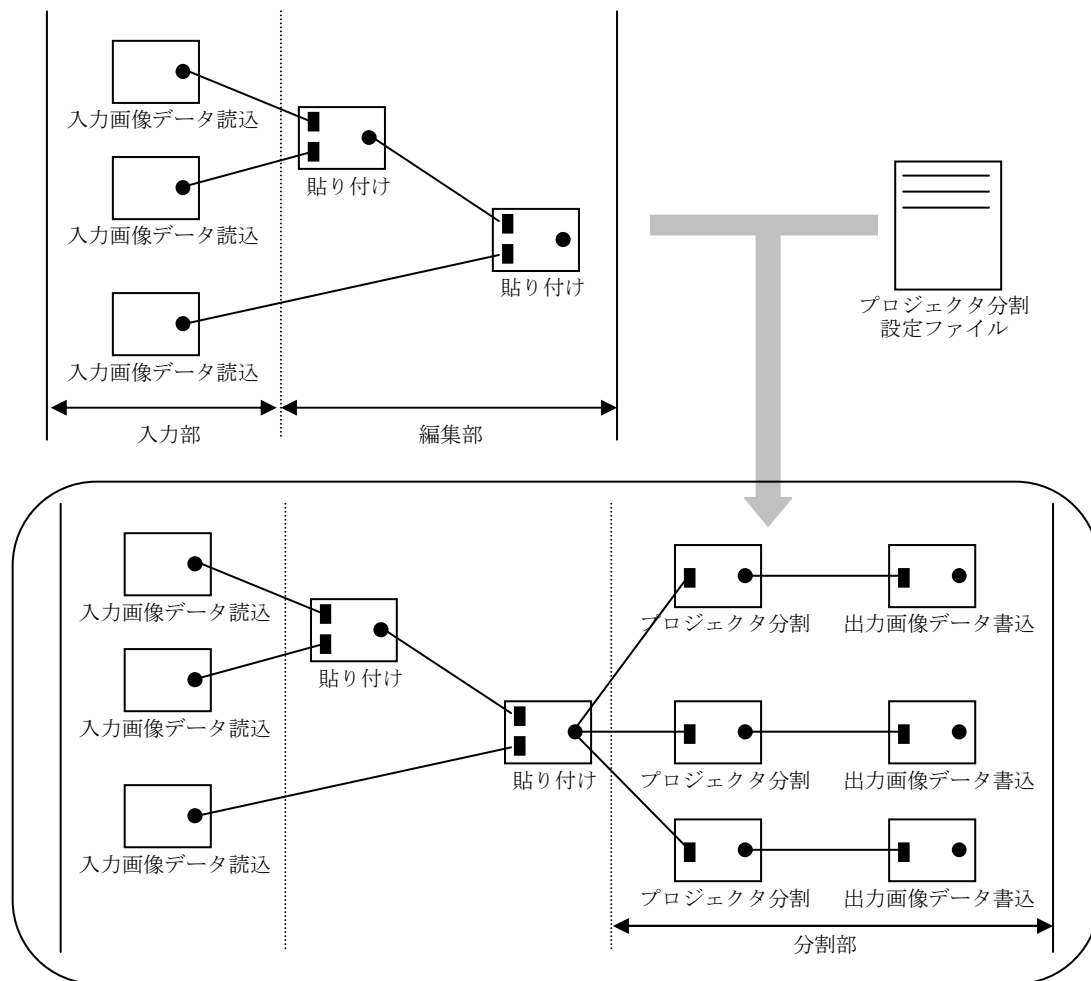


図 12 分割部の自動生成

(2) カメラワーク管理方法の研究

本システムでは、その映像を投影するスクリーンの視野角を考慮し、これを完全に表現可能な射影モデルを 1 つ選択する。そして、使用する素材画像をその射影モデルに合わせた上で合成処理やエフェクト処理を適用する。このとき、幾何学的整合性を考慮し、素材画像の射影モデルを正しく変換するには、その入力画像を撮影したときのカメラパラメータを画像処理のパラメータに設定する必要がある。また、何らかの意図があって、幾何学的整合性を考慮しない場合でも、撮影時のカメラパラメータを無視した値を設定すると映像として成り立たなくなるため、撮影時のカメラパラメータを参考にして、そこからの微調整で値を設定していくのが普通である。

ただし、画像処理のパラメータ設定に使用されるカメラパラメータは、撮影時のものだけではない。画像処理フロー中では画像の射影モデルを変更する画像処理を使用することがある。このような画像処理を使用した場合、この画像処理によって画像の射影モデルおよびカメラパラメータが変更される。そのため、画像処理フロー中のどの位置で画像処理が使用されるかによって、その画像処理のパラメータ設定に使用されるカメラパラメータ

も変わってくる。

このように、IPD の映像制作では、画像処理のパラメータ設定にカメラパラメータを使用する機会が多い。このような設定作業をオペレータの手作業で行うことは、編集の効率を低下させる原因になる。

そのため、本システムでは、映像の撮影時または作成時のカメラパラメータを保存しておき、それを用いて画像処理のパラメータを設定する機能（パラメータ設定支援機能）を設ける。この機能は、素材データの撮影時のカメラパラメータに対してだけでなく、画像処理フロー中で変化していくカメラパラメータについても使用できるようにする。ただし、カメラパラメータを使用した画像処理のパラメータ設定を必要とするのは、射影モデルおよびカメラパラメータの変更を伴う画像処理である。そのため、パラメータ設定支援機能の対象は、画像読込処理、画像書込処理、画像編集処理のうち射影モデルおよびカメラパラメータの変更を伴うものに限定して考えることにする。

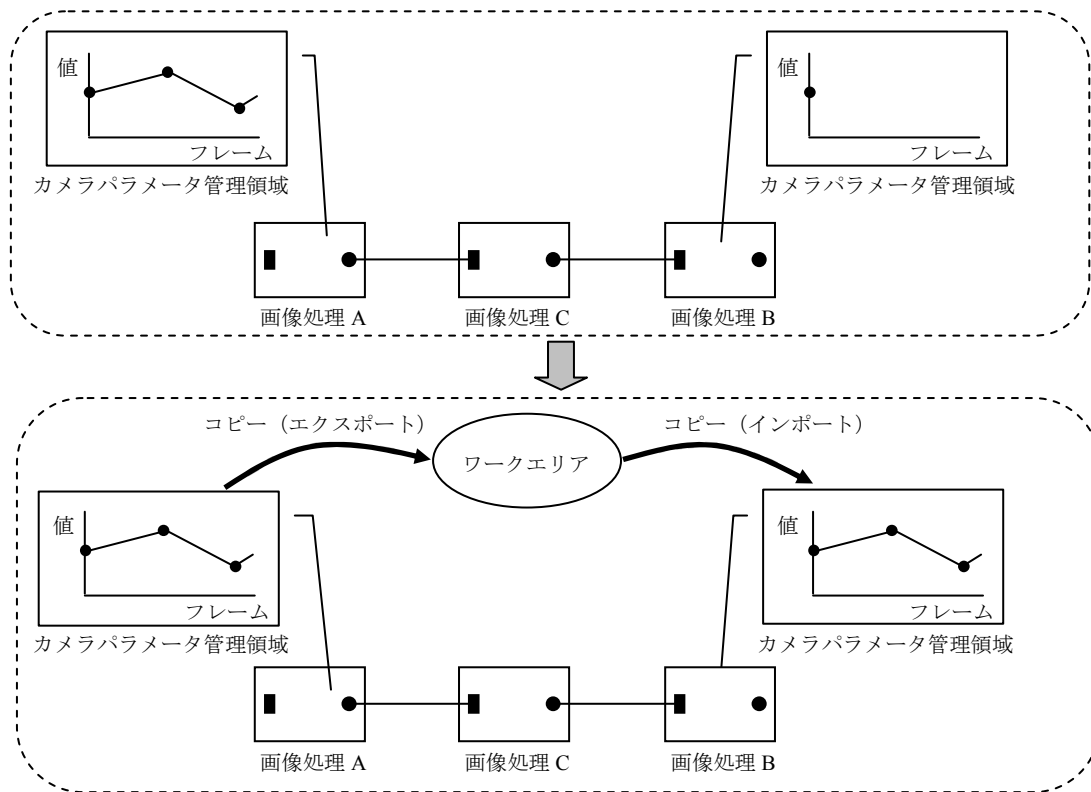


図 13 カメラパラメータのインポート・エクスポート操作

図 13 は、パラメータ設定支援機能の説明図である。画像処理にカメラパラメータの管理領域を設け、個々の画像処理に自分自身のカメラパラメータを管理させる。そして、画像処理 A のカメラパラメータ管理領域の内容をワークエリアにコピー（エクスポート操作）したり、ワークエリアから画像処理 B のカメラパラメータ管理領域にデータをコピー（インポート操作）できるようにする。ここで、カメラパラメータをコピーされた側の画像処理は、そのカメラパラメータを用いて自分自身のパラメータを設定する。

さらに、このような支援機能を持った画像処理が接続された場合には、上記のエキスポート、インポート操作を自動で行い、接続された画像処理が持つパラメータの初期設定を行う。

1 カメラパラメータ管理領域とその更新

上述したように、パラメータ設定支援機能の対象となる画像処理は、画像読込処理、画像書込処理および画像編集処理の中で、入力側と出力側とで射影モデルやカメラパラメータを変更する必要がある画像処理である。図 13 では、1 つの画像処理ノードが持つカメラパラメータ管理領域は 1 つしかないように描いたが、実際には、入力側と出力側に分けて持つ。また、入力ポートごとに異なる射影モデルおよびカメラパラメータの画像を入力したり、出力ポートごとに異なる射影モデルおよびカメラパラメータの画像を出力するような画像処理ノードを作成した場合でも、ポートごとにパラメータ設定支援機能を使用できるように、入力側のカメラパラメータ管理領域の数と出力側のカメラパラメータ管理領域の数は任意に設定可能とする（図 14）。

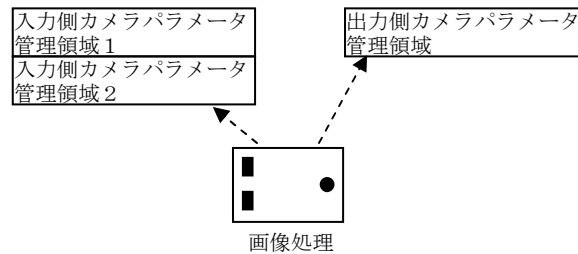


図 14 カメラパラメータ管理領域を複数持つ例

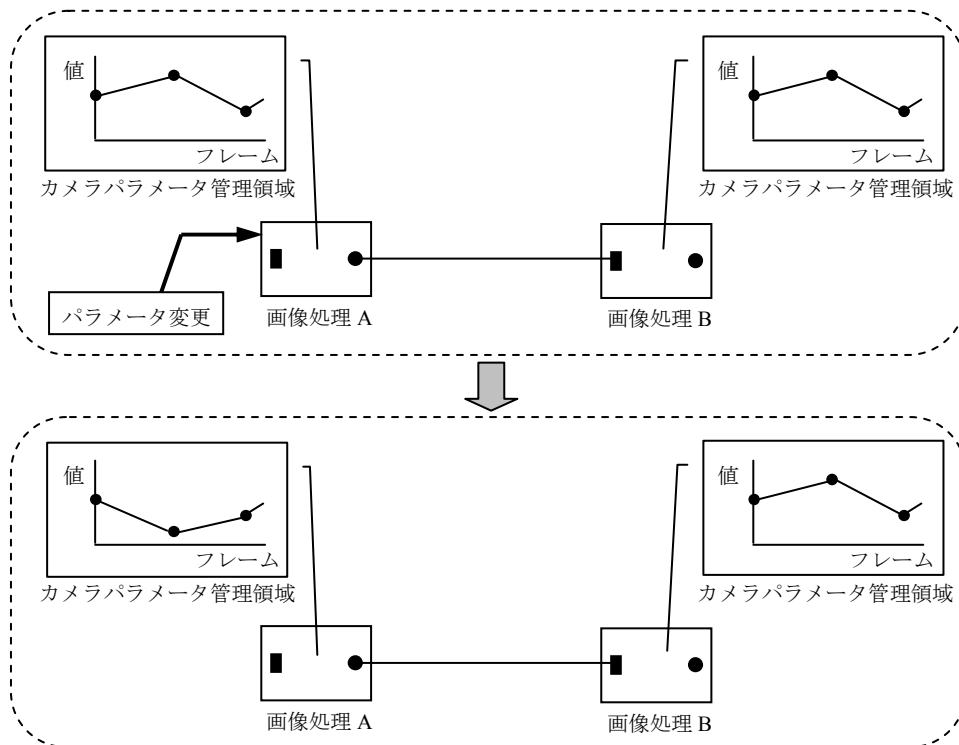


図 15 幾何学的整合性を考慮しない映像を作成するときの設定方法

入力側のカメラパラメータ管理領域の更新は、カメラパラメータのインポートが実行されたとき、パラメータ設定支援機能の対象となる画像処理ノードどうしが接続されたときに行う。また、出力側のカメラパラメータ管理領域の更新は、画像処理ノードが持つパラメータ設定値が変更されたときに行う。

図 13 において、幾何学的整合性を考慮した映像を作成しようとした場合には、画像処理ノード A で出力される画像の射影モデルおよびカメラパラメータと、画像処理ノード B に入力される画像の射影モデルおよびカメラパラメータを同じものとして扱わなければならない。それは画像処理ノード A の出力ポート側と、画像処理ノード B の入力ポート側のパラメータで、射影モデルやカメラパラメータに関する値を同じ値に設定することを意味する。逆にこのパラメータ設定を画像処理ノード A と B で変えることで、幾何学的整合性を考慮しない画像が作成できる。もし画像処理ノード A のパラメータの変更に合わせて、画像処理ノード B のパラメータも変更されると、このような設定が困難になってしまう。

表 2 カメラパラメータとデータ形式

No.	パラメータ名称	形式 (バイト数)	No.	パラメータ名称	形式 (バイト数)
1	バージョン情報	文字列 (9)	16	予備	整数型 (4)
2	画像形式	整数型 (4)	17	↑	↕
3	画角／半径	倍精度型 (8)	18		整数型 (4)
4	画像中心位置偏差 (水平方向)	↑	19		倍精度型 (8)
5	画像中心位置偏差 (垂直方向)		20		
6	カメラ位置偏差		21		
7			22	↓	↓
8			23	予備	倍精度型 (8)
9	カメラ回転角度				
10					
11					
12	投影面の平行移動量	↓			
13		倍精度型 (8)			
14	予備	整数型 (4)			
15	予備	整数型 (4)			

そのため、図 15 に示すようにパラメータ設定支援機能の対象となる 2 つの画像処理が接続されている状態で画像処理ノード A のパラメータが変更され、画像処理ノード A が管理するカメラパラメータに変更が生じた場合でも、その変更は画像処理ノード B に反映し

ない。そして、このような幾何学的整合性を考慮しない設定から考慮する設定に戻すには、カメラパラメータのインポート・エクスポート操作を行えば簡単に設定を戻すことが可能である。

カメラパラメータ管理領域で管理する項目は、上で示したプロジェクタ分割処理や合成処理が持つカメラパラメータに関する設定を考慮して表 2 のようにした。なお、今後システムで使用する射影モデルが増えた場合に管理項目の追加が予想されるため、整数型、倍精度型それぞれに予備領域を設けている。

2 撮影時のカメラパラメータの保管方法

ここでは、映像撮影時のカメラパラメータの保管方法を考える。カメラパラメータは 1 枚 1 枚の画像ごとに異なる可能性があり、画像データとカメラパラメータの対応が取れるように保管する必要がある。この保管方法には、画像自身に保管する方法（方法 1）とデータベースなどの特別な領域に保管する方法（方法 2）の 2 つが考えられる。

本システムでは、画像データとカメラパラメータの対応の取りやすさ、カメラパラメータのバックアップの取りやすさに重点を置き、方法 1 を採用する。通常、画像データは図 16 に示すように、ヘッダ部分とデータ部分で構成される。BMP 形式の画像データを用いて調べたところ、ヘッダ部分にカメラパラメータを挿入した画像データを読み込むことのできない画像処理アプリケーションが複数存在した。しかし、図 17 に示すように画像データの後にカメラパラメータを追記したところ、前述の方法で読み込むことのできなかつたほとんどの画像処理アプリケーションで読込可能となった。そのため、画像データの後にカメラパラメータを保存することにした。

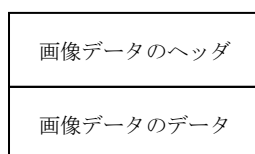


図 16 一般的な画像ファイルの構成

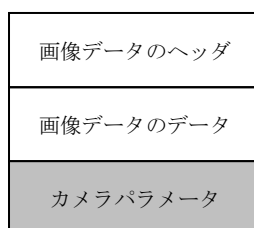


図 17 本システムでのカメラパラメータ保管方法

ただし、この結果のみで、全てのアプリケーションで読み込むことができるとはいえない。また、アプリケーションで加工し保存し直された場合には、カメラパラメータが失われる可能性がある。そのため、カメラパラメータを含んだ画像データを読込むことができないアプリケーションへの対策および加工後にカメラパラメータ失われることへの対策とし

て、図 18 に示すような 2 つのツールを用意する。すなわち、カメラパラメータを含む画像データを、カメラパラメータを含まない通常の画像データに戻すツール（カメラパラメータ除去ツール）と他の画像処理アプリケーションで加工した後の画像データに対して、カメラパラメータを除去する前の画像データを基に、カメラパラメータを含む状態に戻すツール（カメラパラメータ復元ツール）である。

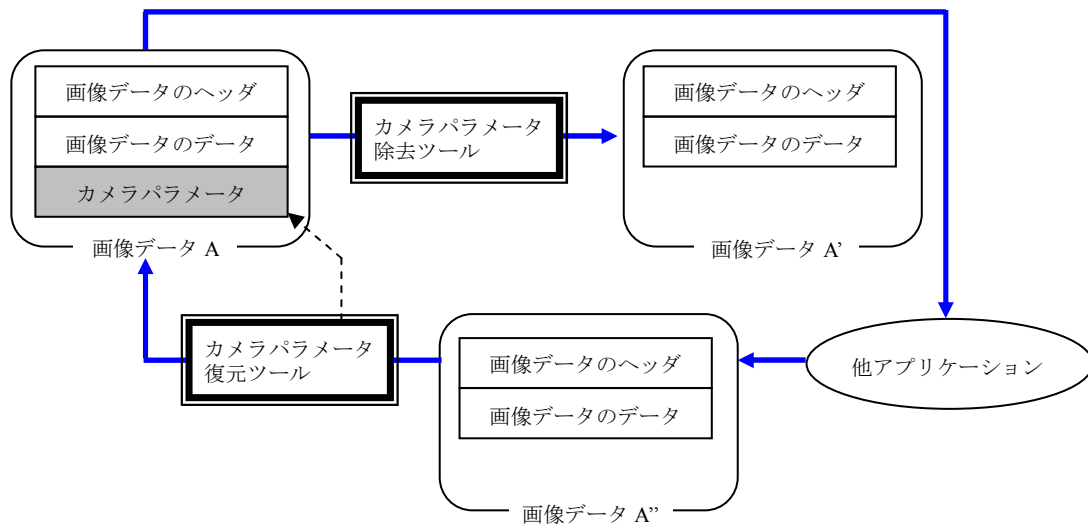


図 18 カメラパラメータ除去ツールと復元ツールの働き

(3) 実験コンテンツの制作

1 実写コンテンツの制作実験

1.1 スクリーン形状

本コンテンツを投影するスクリーンの形状を図 19 に示す。

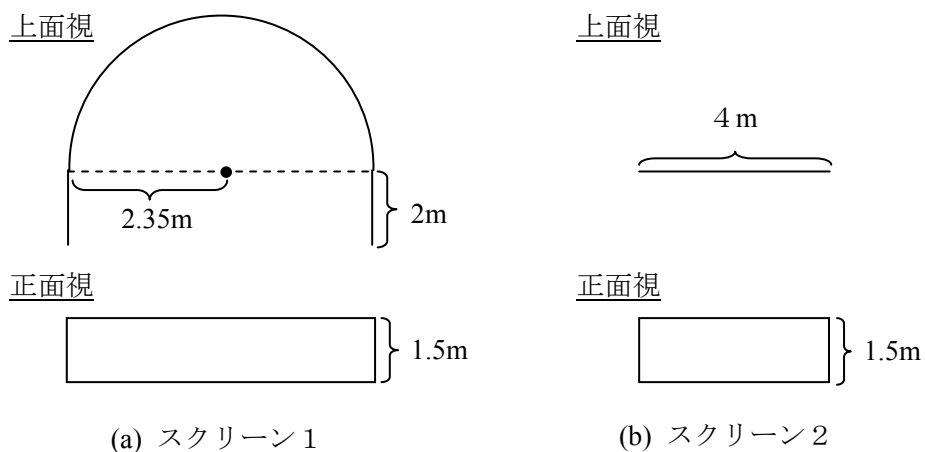

















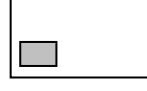
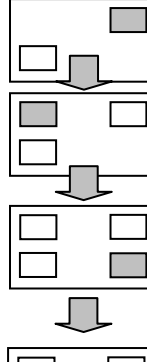
図 19 コンテンツを表示するスクリーン


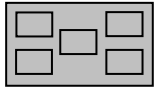
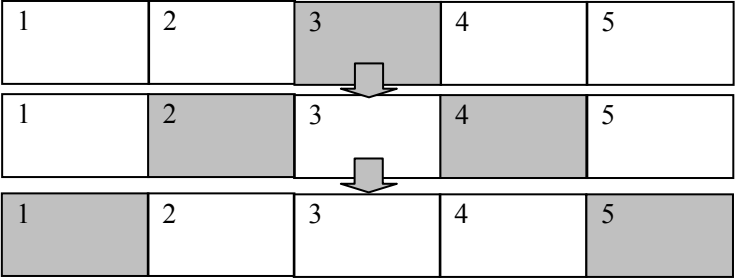
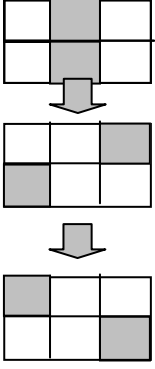
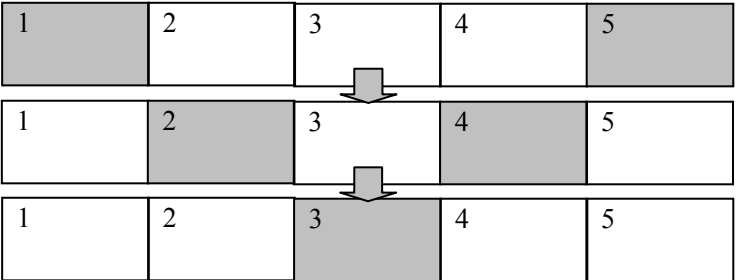
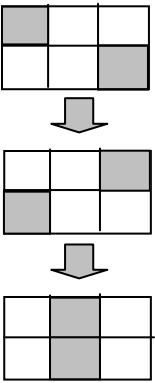
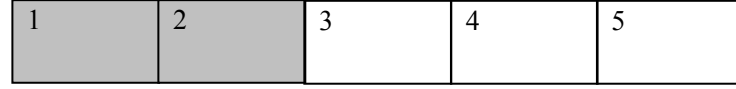
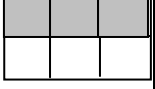

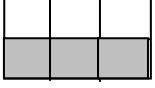


図 19 に示すように、本コンテンツを表示する2つのスクリーンは、水平方向の視野角が大きく異なるため、シナリオは共通とするが、それぞれのスクリーン用のコンテンツごとに映像効果の付け方を変えることにした。そのため、画像処理フローもスクリーン別に編集した。

1.2 シナリオ

以下にコンテンツのシナリオを示す。なお、スクリーン1については、1つの映像をスクリーン全体に表示する部分と5つの異なる映像を横に並べてスクリーン全体に映像を表示する部分で構成する。

No.	スクリーン1	スクリーン2	時間 [秒]
1			1
	<p>鮮やかなハイビスカスの映像を表示する。スクリーン1については、1、2および4、5の部分はカーテンを表示し、スクリーンが隠されている状態を演出する。</p>		
2			6
	<p>No.1の映像を背景にして、タイトルを表示する。タイトルは1秒のフェードインで表示し、表示後4秒表示し、1秒でフェードアウトさせる。</p>		
3			2
	<p>スクリーン1については、2で表示していたカーテンが左から右に移動することにより、グスク時代の遺跡（グスク時代遺跡1）が表示される。スクリーン2については、フェードインにてグスク時代遺跡1が表示される。 No.3全体の長さは2秒とする。スクリーン1では、カーテンが左から右に移動を始め、グスク時代遺跡1が完全に表示されるまでの時間を2秒とする。スクリーン2では、1秒のフェードインでグスク時代遺跡1を表示する。</p>		
4			9
	<p>グスク時代遺跡1の上にグスク時代とその年代を文字で表示する。2秒のフェードインで文字を表示する。その後3秒表示し、1秒のフェードアウトで文字を消す。フェードアウト後の3秒継続する。</p>		
5			2
	<p>フェードインにて、グスク時代の別の遺跡（グスク時代遺跡2）を表示する。フェードインは1秒とする。フェードイン後も1秒継続する。</p>		

6	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		2																														
1	2	3	4	5																																		
<p>スクリーン1については、4 で表示していたカーテンが右から左に移動することにより、三山時代の遺跡（三山時代遺跡1）が表示される。 スクリーン2については、フェードインにて三山時代遺跡1が表示される。 No.6 全体の長さは2秒とする。スクリーン1では、カーテンが左から右に移動を始め、三山時代遺跡1が完全に表示されるまでの時間を2秒とする。 スクリーン2では、1秒のフェードインで三山時代遺跡1を表示する。</p>																																						
7	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		9																														
1	2	3	4	5																																		
<p>三山時代遺跡1の上に三山時代とその年代を文字で表示する。2秒のフェードインで文字を表示する。その後3秒表示し、1秒のフェードアウトで文字を消す。フェードアウト後も3秒継続する。</p>																																						
8	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		2																														
1	2	3	4	5																																		
<p>フェードインにて、三山時代の別の遺跡（三山時代遺跡2）を表示する。フェードインは1秒とする。フェードイン後も1秒継続する。</p>																																						
9	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		1																														
1	2	3	4	5																																		
<p>首里城をフェードインで表示する。フェードインの長さは、1秒とする。</p>																																						
10	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		7																														
1	2	3	4	5																																		
<p>首里城の映像の上に、琉球王国とその年代を文字で表示する。2秒のフェードインで文字を表示する。その後3秒表示し、1秒のフェードアウトで文字を消す。文字がフェードアウトした後も1秒継続する。</p>																																						
11	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5		1.5																														
1	2	3	4	5																																		
<p>首里城の映像の上にフェードインで守礼門を表示する。フェードインは0.5秒とする。フェードイン後も1秒継続する。</p>																																						
12	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td colspan="3"></td> <td style="text-align: center;">↓</td> <td></td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td colspan="2"></td> <td style="text-align: center;">↓</td> <td colspan="2"></td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> <tr> <td colspan="2"></td> <td style="text-align: center;">↓</td> <td colspan="2"></td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5				↓		1	2	3	4	5			↓			1	2	3	4	5			↓			1	2	3	4	5		2
1	2	3	4	5																																		
			↓																																			
1	2	3	4	5																																		
		↓																																				
1	2	3	4	5																																		
		↓																																				
1	2	3	4	5																																		

	首里城の装飾品を順番に表示する。各映像はカットインで表示し、表示後 0.5 秒で他の映像をカットインさせる。		
13			7
	全体の明るさを落とし、沖縄県設置とその年号を表示する。2 秒のフェードインで文字を表示する。その後 3 秒表示し、1 秒のフェードアウトで文字を消す。フェードアウト後も 1 秒継続する。		
14			1.5
	海中、民家などの映像を順番に表示する。各映像はカットインで表示し、表示後 0.5 秒で他の映像をカットインさせる。		
15			1.5
	平和記念公園などの映像を順番に表示する。各映像はカットインで表示し、表示後 0.5 秒で他の映像をカットインさせる。		
16			6
	アメリカ統治時代とその年代を示した文字を表示する。2 秒のフェードインで文字を表示後に 3 秒表示し、1 秒のフェードアウトで文字を消す。		
17			6
	沖縄返還とその年号を示した文字を表示する。2 秒のフェードインで文字を表示する。その後 3 秒表示し、1 秒のフェードアウトで文字を消す。		
18			5
	スクリーン全体に海岸を表示する。No.20 のシーンからの切り替えはフェードインとし、切り替え時間は 1 秒とする。フェードイン後 4 秒継続する。		

19	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5	<table border="1"> <tr> <td></td> </tr> </table>		8
1	2	3	4	5					
<p>スクリーン全体に桜の映像を表示し、その上にそこで暮らす人々の映像が表示される。No.18 からの切り替えは、フェードインとする。切り替え時間は1秒とし、その後7秒継続する。</p>									
20	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5	<table border="1"> <tr> <td></td> </tr> </table>		5
1	2	3	4	5					
<p>琉球舞踊を踊る人々を表示する。No.19 からの切り替えは、フェードインとする。切り替え時間は1秒とし、その後4秒継続する。</p>									
21	<table border="1"> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table>	1	2	3	4	5	<table border="1"> <tr> <td></td> </tr> </table>		6
1	2	3	4	5					
<p>エイサーを踊る人々を表示する。No.23 のシーンから No.24 のシーンへの切り替えは、下記に示すように、スクリーンの真ん中から上下方向に No.23 の映像が No.24 の映像に少しずつ切替る。切り替え時間は1秒とし、その後5秒継続する。</p>									

1.3 評価および検討

以下、実験コンテンツ1の制作に本システムを使用した結果を踏まえ、本システムの評価および検討を行った。従来システムにおいては、レンダリングしようとする画像処理フローで使用される全ての入力画像データを、ディスクアレイ上に読み終わらなければレンダリングを実行することができなかった。このとき、入力画像データを読み込むために必要なディスクアレイの空き容量が不足する場合、不要なデータの削除やテープへの移動の後、入力画像データの読み込みを行っていた。これらの処理は、オペレータの手動操作を必要としていたため、オペレータの拘束時間が多くなっていた。しかし本システムでは、入力画像データを置くために必要な空き容量がディスクアレイ上にない場合、不要なデータの削除やテープへの移動を行い、入力画像データをテープからディスクアレイに読込んだ後、画像処理フローを実行するという一連の作業を自動で行うことが可能である。そのため、実験コンテンツ1の制作において、オペレータはデータの削除やテープへの移動といったディスクアレイ上の空き容量を確保するための作業に束縛されることなく、画像処理フローの編集に専念することができた。

制作時間に関しては、レンダリング PC を複数台使用することによって短縮することができた。しかし1台あたりの画像処理速度が必ずしも高速化されているわけではないので、その対応が今後の課題である。また、画像処理フローの編集においては、分割部の自動生成機能の効果が大きかった。今回のコンテンツ制作では、各スクリーン用のコンテンツごとに5つの画像処理フローを編集した。これまでの編集環境においては、これら5つの画像処理フローに対して、同じ分割部をオペレータが手作業で編集していた。そのため、分割部で使用される画像処理のパラメータ設定でしばしば設定ミスが生じていた。しかし、本システムでは、1つのプロジェクトシステムごとに1つの設定ファイルを作成し、それを用いて分割部を自動生成できる。そのため、今回の実験コンテンツの制作では、上記のような設定ミスは1度も生じなかった。

2 CGコンテンツの制作実験

試作した映像編集システムの、CGコンテンツ制作への応用を試みる。そこで必要となる作業工程と編集環境の効果を確認するため、実際の映像コンテンツ制作業務で行われる作業工程に沿って、ユーザテストを行った。

2.1 作業環境とソフトウェア

① 3DCG ソフトウェア

3DCG ソフトとしては、広く CG 業界で用いられる汎用ソフトウェアを用いた。このソフトは、プラグインなどの提供数も多く、またネットワークレンダリング作業の速度や、スクリプトを用いたカスタマイズ性の高さなどにも定評がある。また、柱面映像や球面映像を生成する機能もデフォルトで用意されている。ただし、この作成過程においては、立方体上にマッピングされる 6 面の正方形画像でのレンダリング映像を作成してから幾何変換を行う手法を採用している。このため、幾何変換前の画像には十分な解像度の映像をレンダリングしないと画質の劣化が起こる可能性も存在する。

② IPD 用ビューア

結果を確認するビューアとして、最終表示用の動画ビューア、複数の画像に対しての合成処理をエミュレートする調整用静止画ビューア、リアルタイムでの色調整・変換を可能とする動画ビューアの 3 つを用意した。これらを適切に使い分けることで、IPD の実機を用いてのプレビュー作業の回数を減らし、効率のよいフィードバックを行うことが出来る。

③ IPD 映像表示環境

IPD 表示環境としては、180°視野角（横方向）の球面スクリーン(図 20)によるものを利用した。このシステムでは複数プロジェクタ（6 台）による表示が行われる。



図 20 実験対象に用いた 180°球面スクリーン

2.2 コンテンツ制作の工程フロー

図 21 に示す。本テストコンテンツ制作における工程フローを示す。

広視野角・高精細なスクリーンに表示されたオブジェクトを直接指し示し、そのオブジェクトに対して操作を行なうことが可能なポインティングデバイスのようなユーザインタフェースについて考える。

ここでは、3次元計測装置と市販のゲームコントローラを用いて実現することを試みた。超音波センサにより3次元位置を、角速度センサ（ジャイロ）により角速度成分（Pitch, Roll, Yaw）を取得することができる。この3次元位置と、角速度成分の組からなる視線ベクトルと、スクリーンの形状から、指し示している点を算出することで、ポインティングデバイスとして使用することを可能とする。

本研究で検討を行ったUIの概略を図22に、UIの外観を図23に示す。ゲームコントローラを基本に、上部には超音波センサを2個、下部には各速度センサを1個設置している。角速度センサが大きいので、外装をパテで整形し、銃のような形にしている。ゲームコントローラは全部で12個のボタンがついているが、今回はそのうちの8個のみを使用している。また、このゲームコントローラにはトリガもついており、このトリガもボタンの一つとして使用する。

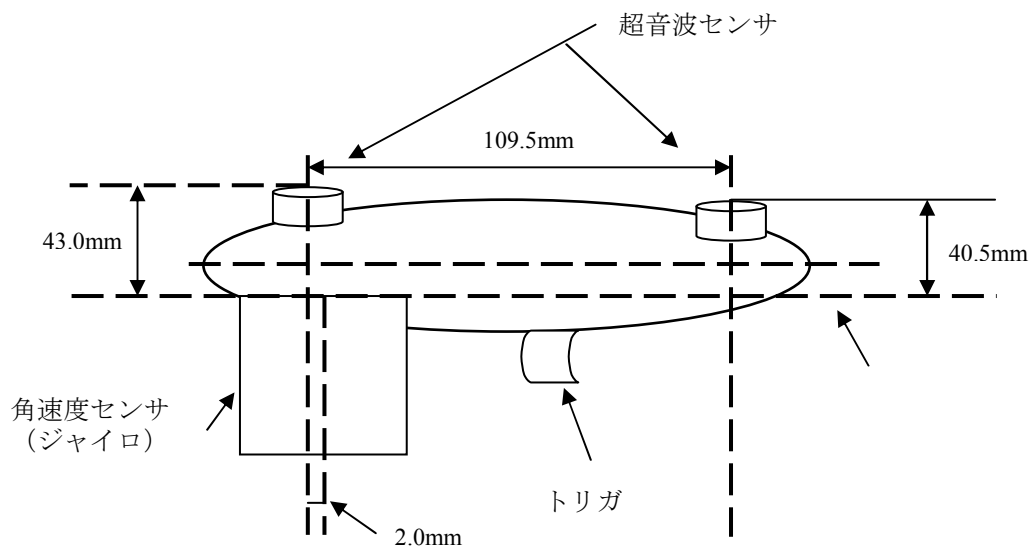


図22 UIの概略

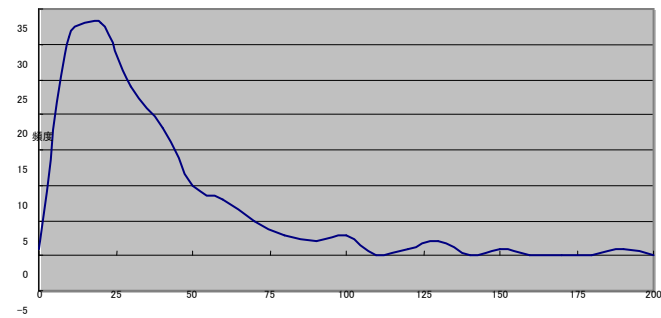


図23 UIの外観

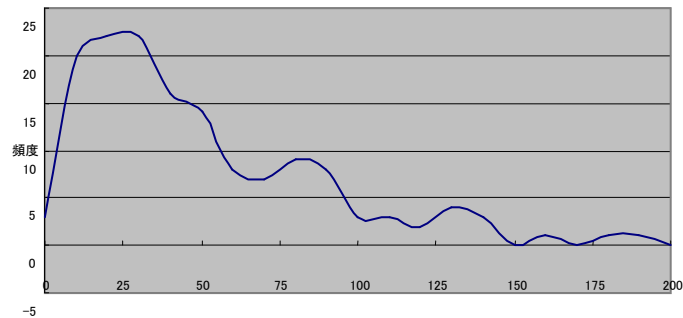
3.2 IPD 用 UI の評価実験

ここでは、前節で述べた UI システムの、ポインティングの精度評価実験を行なった。

その結果を図 24 に示す。スクリーン上の画像のずれを水平方向と垂直方向に分けて誤差を求めている。単位は pixel である。結果を見ると、水平方向、垂直方向ともに 20~50pixel 程度であることが分かる。これは、実際にレーザポインタ等を照射して比較すると違いが分かるが、ユーザがポインティングデバイスとして、単体で動かしている分においては違和感を感じることなく使用することが可能であり、実用的なレベルであると言える。



スクリーン方向の画像のずれ(水平方向)



スクリーン上の画像のずれ(垂直方向)

図 24 IPD 用 UI のポインティング精度誤差

課題ア-4 リアルタイム映像作成技術

本課題では、複数のビデオカメラで撮影した映像を遠隔地に送信し、リアルタイムに幾何変形などの画像処理を施して、マルチプロジェクタによる IPD を用いて表示するシステムに関して、その実現に必要な各種技術の研究を行った。

(1) では、光学中心が一致していないマルチカメラシステムで撮影した映像を、画像処理によって見た目に違和感なく繋ぎ合わせ、1 枚の高精細・広視野角な映像を得る方法、並びに、そのようなマルチカメラシステムと没入型ディスプレイとの連携方法について、検討・開発および実験を行った。具体的には、カメラのレンズ歪みを除去するための技術、カメラの内部・外部パラメータを推定するための技術、撮影系と表示系とを統合した系での歪み補正パラメータ算出技術などの研究に取り組んだ。またそれらに加え、繋ぎ合わせの精度向上を目的として、複数のカメラによって撮影されている被写体の 3 次元的な位置を推定する技術についても検討を行った。

(2) では、課題アおよび課題イの組み合わせで実現できるアプリケーションの検討と、その実機による検証を目的とし、実験ベースでの研究を行った。具体的には、マルチカメラで撮影したライブ映像を、遠隔地に設置した IPD で表示するという想定の下で、映像信号を圧縮・送信する仕組みを開発し、さらに通信路には課題イで開発した技術を適用し、通信品質の制御を行うようにした実験システムを構築し、システム性能の評価並びに実用化に際しての課題の抽出を行った。

(1) 複数カメラによる撮影映像のリアルタイム変換技術

1 概要

大型のスクリーンを用いてユーザの視野の大部分を覆うような広視野角映像を表示し、ユーザに高い臨場感を与えることを目的とした、没入型ディスプレイ (IPD: Immersive Projection Display) の 1 つの実現形態として、我々は複数のプロジェクタから投映された映像を滑らかに繋ぎ合わせ 1 つの大きな映像として表示するシステム (プロジェクタアレイシステム) を開発した[Yamasaki 2002]。

このプロジェクタアレイシステムに表示するコンテンツについては、多くのタイプのもので考えられるが、それらは大きく実写映像を使うものと CG を使うもの、あるいは、リアルタイム型のものでパッシブ型のものという分類を行うことができる。この中で、実写映像を用いかつリアルタイムに表示するコンテンツに関しては、今まで有効な技術を確立していなかった。

そこで本研究では、高精細・広視野角な実写映像の撮影システムを構築するとともに、広い範囲を大画面でリアルタイムに観察する遠隔監視システムや、遠隔地のコンサートや講義を臨場感あふれる映像として体験できるライブ映像中継システムなどへの応用を睨み、高精細・広視野角な実写映像を、撮影から表示まで一貫して行えるようなシステムを構築することを目的とした。

2 全体システム構成

本研究で構築したシステムは大きく (1)撮影システム、(2)伝送システム、(3)表示システ

ム、以上の 3 つの部分で構成される。ここでは、まずシステム全体の構成について概略を述べた後、撮影システムと表示システムについて実際に本研究で用いたシステムの具体例を紹介し、最後にシステム全体の調整方法についてその方針を説明する。なお、伝送システムの詳細な説明は (2) に譲る。

2.1 概要

本研究で構築したシステムの全体構成を図 1 に示す。

外界の状況は同期機構を持った 6 台のカメラで撮影され、MPEG エンコーダに送られる。MPEG エンコーダは、6 本の映像データを同期を取りつつ MPEG2 形式で圧縮し、1 本のデジタルデータとして MPEG デコーダに向けて送信する。(遠隔地にあると想定した)MPEG デコーダは、ネットワークを介して先のデジタルデータを受信すると、MPEG2 のデコード処理を行い、6 本の映像データとして同期を取って、スキャンコンバータに向け出力する。スキャンコンバータは、受信した映像データに対して撮影時の映像フォーマットから投映時の映像フォーマットへの変換を行い、映像変換ハードウェアに向け出力する。映像変換ハードウェアでは、受信した映像データに対して、あらかじめ登録された変換パラメータに従って形状・明るさ・色合いの補正を行い、補正後の映像データをプロジェクタに向けて出力する。このような流れで各々のプロジェクタから投映された映像を重ね合わせた結果が、ユーザが鑑賞する映像となる。なお実際のシステムでは、上記映像処理過程で現れる機器の他に、映像変換ハードウェアの制御(変換パラメータの登録など)を行うための PC が必要となる。

上記において、本研究のシステム固有の特徴は、大きく下記の 3 点である。

第 1 の特徴は、撮影システムの構成である。本研究では、表示システムとして使用するプロジェクタアレイのプロジェクタ台数に関するスケーラビリティを有効に活用するために、高精細・広視野角な実写映像の撮影システムとして、表示システムと同様の考え方、すなわち、複数のカメラを異なる方向を向くように配置し、それらの撮影した映像を繋ぎ合わせることで目的とする映像を得るといった考え方をとった。さらに、撮影システムから表示システムまで全体のシステム構成を単純にするために、プロジェクタとカメラの台数を同一にして 1 対 1 に対応させデータの授受を行うようにした。こうすることによって、複数のカメラの撮影映像をデジタルデータとして合成する必要がなくなるため、精細度や視野角の点で制約のない、安価で遅延の少ないシステムを構築することができる。

第 2 の特徴は、MPEG エンコーダと MPEG デコーダから成る部分である。遠隔監視アプリケーションへの応用を視野に入れ、(1)映像信号としてではなく長距離伝送時の取扱いが容易で品質劣化を起こしにくいデジタルデータとして伝送するようにしたこと、(2)伝送路には 1 本のデジタルデータとして流すようにしたこと、(3)複数の映像ストリームをリアルタイムに圧縮・伸張処理しつつも表示システムの入力段階まではそれらの同期を維持するようにしたこと、が特徴となる。伝送システムに関しては、(2)にて詳細に説明する。

第 3 の特徴は、映像変換ハードウェアの部分である。最終的にスクリーン上に映される映像が(単一視点制約を満たしている場合に)滑らかにつながって見えるためには、

- (1) 撮影システム・表示システムの設計値から決まる理想的な形状補正；

- (2) プロジェクタの設置誤差・レンズ歪みの補正；
- (3) プロジェクタの個体差による色調や明るさの違いの補正；
- (4) スクリーン上で複数の映像が重なっている部分の明るさの補正；
- (5) スクリーンの設置誤差・形状歪みの補正；
- (6) カメラの設置誤差・レンズ歪みの補正；
- (7) カメラの個体差による色調や明るさの違いの補正；

などが必要となる。これらをリアルタイムに処理するために、本研究では、既開発の映像変換ハードウェアにこれらを補正する個々の写像の合成写像としての変換パラメータを登録して処理することにした。なお該変換パラメータを作成する具体的な方針としてはいくつかのものが考えられるが、本研究では実際の運用を考慮し、表示システムに関する変換パラメータが既に求められている、つまり没入型ディスプレイはすでに調整済みである、という前提の下で、撮影システムに関する変換パラメータを作成し両者を合成する、という観点で考察を行っている。また、カメラの個体差による色調や明るさの違いに関しては、個々のカメラでテストチャートなどを使って事前に調整しておくこととし、撮影システムに関する変換パラメータとしては幾何学的な補正を行うものだけに絞って検討を行った。

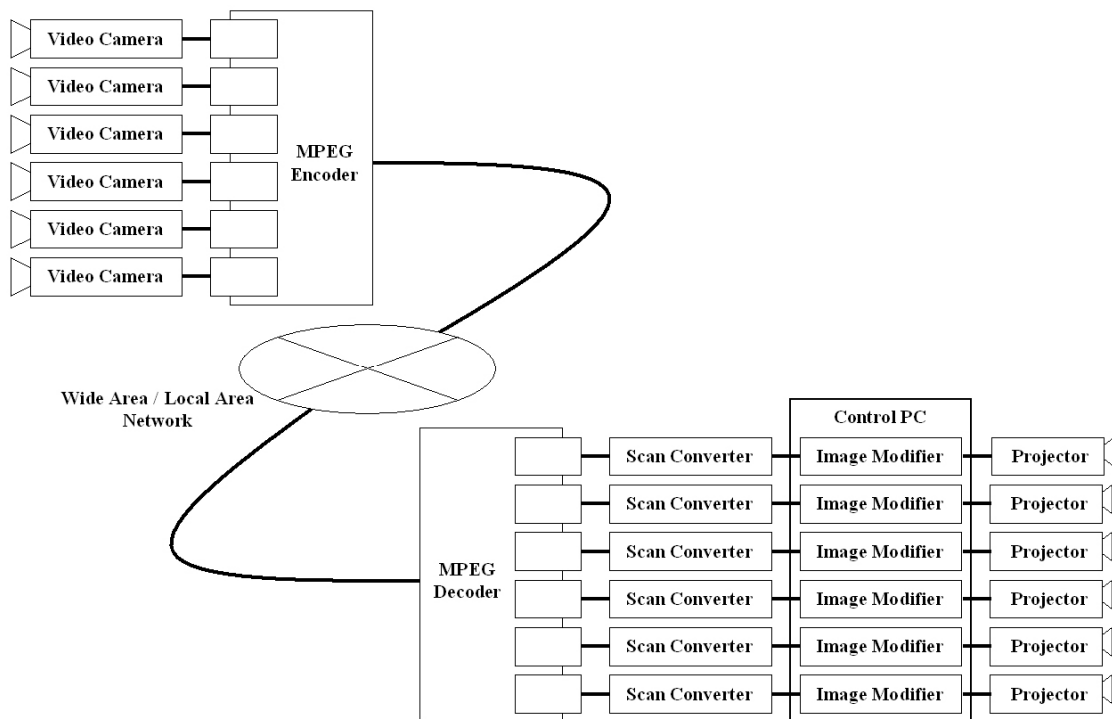


図1 全体システム構成

2.2 撮影システム

本研究では、カメラ6台を使って撮像システムを構築した。その撮影システムの写真を下記に示す。先に述べたとおり、本研究では、単一視点制約を満たさない撮影システムを

研究対象とした。また、光軸同士が成す角度などの具体的な配置は異なるが、カメラとプロジェクタとの1対1対応を実現させるために、後述の表示システムと同じ「横3台×縦2台」の構成をとっている。なお実際には、各カメラの映像はスキャンコンバータで横1024画素×縦768画素の画像に変換された上で処理される。色調に関する個々のカメラの個体差については、図のようなパターンを撮影し、目視によって調整した。

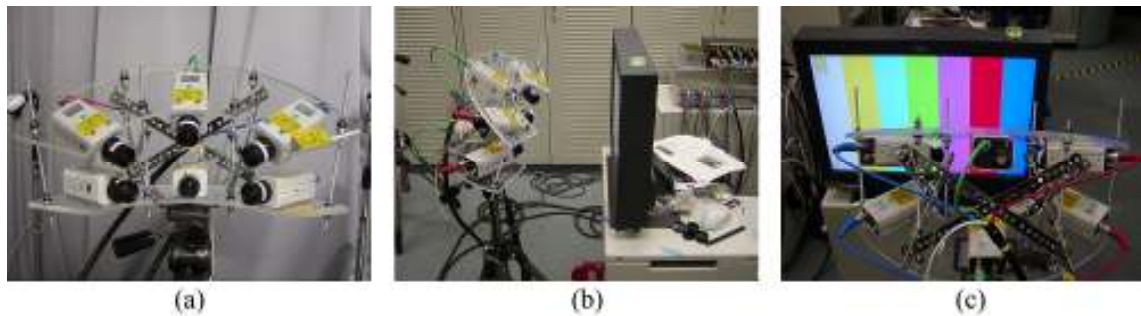


図2 本研究で使用した撮影システム：(a)外観 (b)撮影風景 (c)色調整パターン

上図のように、6台のカメラは3台を1セットとして保持板の上に固定した(図2(a))。隣接するカメラの光軸の成す角度は約50度である。左右のカメラの撮影領域の端には反対側に位置するカメラが写り込むが、その部分に関しては、後述のレンズ歪み除去の段階で表示範囲から除かれるものと予想されるため、できるだけ画角を広く取ることを優先して上記のような設計とした。この配置での左右方向の画角はおよそ160度である。カメラを固定した保持板は、中央の2台のカメラの位置関係が図2(b)の関係となるように固定する。上下のカメラの光軸の成す角度を約40度としたため、撮影システム全体としての上下方向の画角はおよそ90度となる。また、このとき、上下あるいは左右に隣接するカメラ同士の光学中心間の距離は、およそ10cm程度である。

このようなマルチカメラシステムでは、システムをキャリブレーションした後、カメラ間の幾何学的な関係をキャリブレーション時のままに維持することが必要となる。本研究ではその容易さを重視し上記のような設計としたが、この設計方針には欠点もある。それは、左右に位置するカメラの傾きである。各カメラの配置(重複して撮影する画角の割合)は表示システムとの連携を考慮した上で決める必要があるが、上の写真のような配置では、中央上下のカメラ間の角度をちょうどよく合わせると、左側上下・右側上下のカメラ間で重複して撮影する部分の画角の割合が大きくなりすぎてしまう。これは、実際問題として撮影解像度の低下や撮影画角の低下に繋がることになるため、それらの条件が厳しい場合には各カメラの位置や姿勢を個別に調整できるような架台とすることを優先すべきである。

2.3 表示システム

本研究では、プロジェクタ6台(横3台×縦2台)を使った表示システムを使用した。なお、表示システム全体での表示解像度は2580(H)×1350(V)程度、視野角は水平180度・

垂直 80 度程度である。

3 マルチカメラのキャリブレーション

システム全体のキャリブレーション時のデータフローを図 3 に示す。キャリブレーション時のシステム構成は、実際の運用時のシステム構成に加えて、制御用 PC からの制御信号を受け取ることのできるパターンジェネレータ(PC)と、パターンジェネレータの生成したキャリブレーションパターン画像を表示可能な大画面高精細液晶ディスプレイとを備えたものである。大画面高精細な液晶ディスプレイとしては、画面サイズ 22.2 インチで横 3840 画素×縦 2400 画素の表示が可能であるものを使用した。液晶ディスプレイを使用したのはパターン表示面の平面性が高いことが期待できるためである（後に、レンズ歪みのキャリブレーションとカメラの内部パラメータ・外部パラメータのキャリブレーションにおいて、平面パターンを使用する手法をとる）。以下、システムのキャリブレーションの流れを説明する。

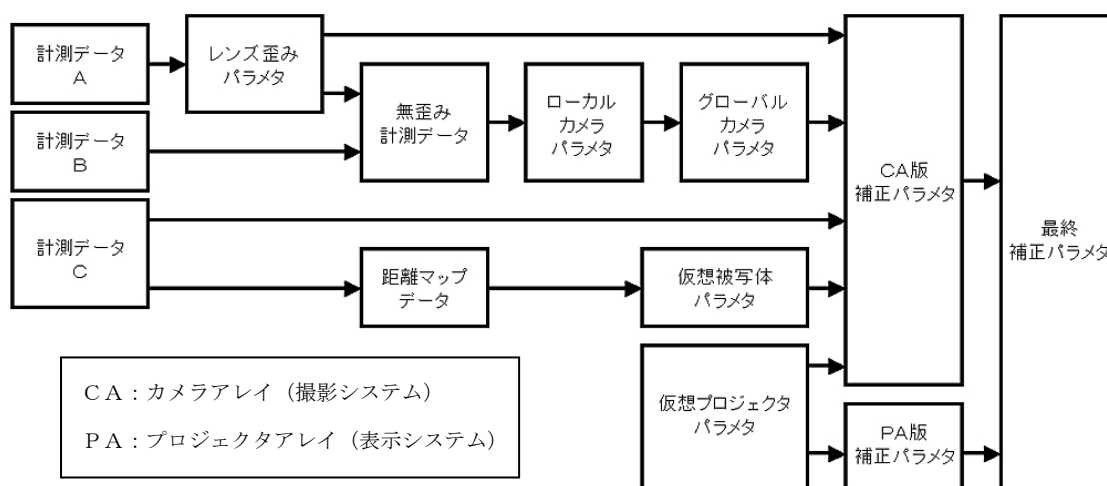


図 3 全体システム調整のデータフロー図

計測データ A は、レンズ歪みを補正するためのものであり、パターンジェネレータから出力したパターン画像に対して各単体のカメラで撮影を行う。レンズ歪みを補正する方法としては、[Tsai 1987]の方法の他、いくつかの方法が提案されている。本研究においては、映像の周辺部まで含めた映像全体の歪みを取るべく、各カメラの撮影領域全体にパターン画像が写るように注意して撮影を行い、補正パラメータの算出を行った。補正前後の画像をそれぞれ図 4 (a)と図 4 (b)に示す。

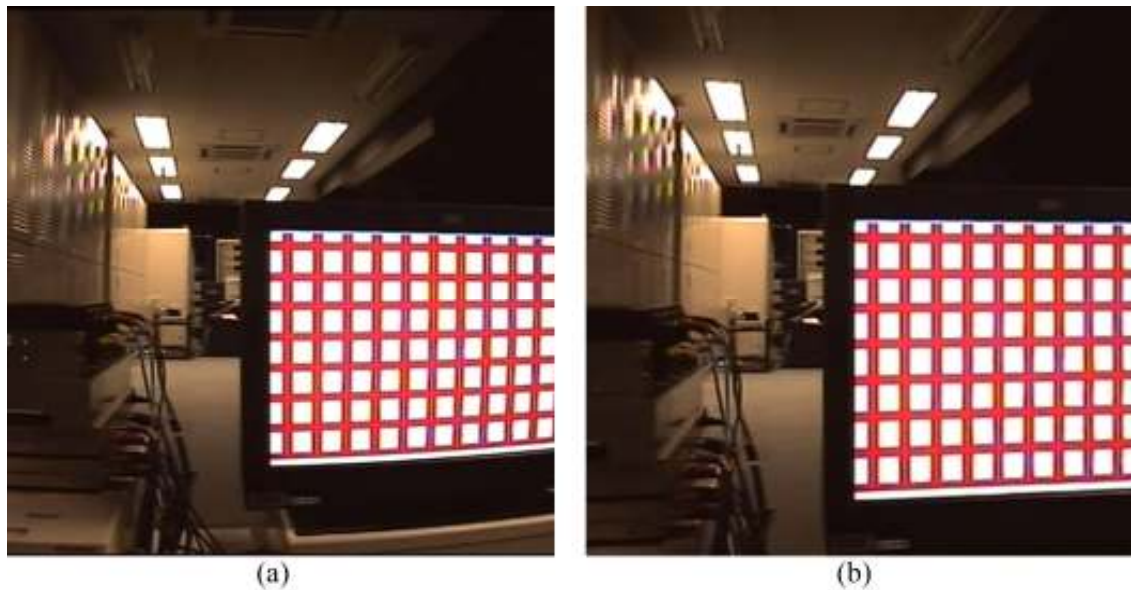


図4 レンズ歪み補正の効果

計測データ B は、各カメラの内部パラメータ（焦点距離，画像中心，アスペクト比，軸の非直行性歪み）・外部パラメータ（ワールド座標に対する位置・姿勢）を推定するためのものである。パターンジェネレータから出力したパターン画像に対して、隣接するカメラペア毎に撮影を行う。ここで、3 台以上のカメラの組に対して同時に撮影せず、2 台ごとの撮影としたのは、カメラの台数が増えた場合やカメラの配置が変わった場合にも同様の手順でキャリブレーションが可能となるように考慮してのものである。各カメラの外部パラメータの推定は、まずは隣接するカメラ 2 台の間の相対関係を求めた後に、（厳密にはいくつかのカメラのパラメータを参考にするが、そこからどう決定するかは自由である、という意味で）それと独立に基準となる座標系を決め、基準座標系と各隣接カメラ間の相対関係とから最終的なパラメータを決定する、という手順で行った。計測データ A を使わずに、計測データ B だけから各カメラのレンズ歪み・内部パラメータ・外部パラメータのすべてを非線形最適化によって求めることも可能であるが、レンズ歪みに関しては単独で精度を評価することが可能であり、また、上記非線形最適化における再投影誤差最小化の考え方がレンズ歪みの推定時の評価としては適切でないという考えから、本研究では個別に推定する方針をとった。計測データ B に対して先に求めたレンズ歪みの補正を施すことで、歪みの影響の少ない高品質のデータが期待できるため、そのデータを使って内部パラメータや外部パラメータの推定を行うようにすることで線形計算ベースでも各パラメータの推定値が良好なものとなることが期待できる。各カメラの内部パラメータの推定と、各カメラペア毎の外部パラメータの推定は、カメラの個数が 2 個の場合に限定した上で植芝の方法[植芝 2002]をアレンジして実装し、以下のような手順で行った。ここで、植芝の方法は、平面パターンを用いて複数のカメラを同時にキャリブレーションするものであり、Zhang による単体カメラ向けの手法[Zhang 1998][Zhang 2000]を拡張したものである。植芝の方法では、

各カメラからすべてのモデル平面が見えていることが必要条件となる。下記の方法は、その条件を満たしつつ撮影データの有効活用を図ったものである。計測時間の面での効率化を考えた場合には、縦横各 2 台・合計 4 台のカメラを 1 組にして相対関係を求めていく方針なども考えられ、先に述べた拡張性の点でもほとんど遜色ないと思われるが、今回、それでも 2 台 1 組での撮影としたのは、パターンを表示するディスプレイのサイズとカメラの画角や配置を考慮してのものである。

- (1) 計測データ B から各カメラに関するデータをそれぞれすべて抽出する。
- (2) 乱数を使い各カメラに対して異なるサンプル点の集合を 2 つずつ生成する。
- (3) 各カメラに対して、上記 2 つのサンプル点集合を 2 つの別のカメラとみなして射影変換行列を生成し、植芝の方法を適用して、内部パラメータを求める。
- (4) 各カメラペアに対して植芝の方法を適用し、外部パラメータを求める。その際、先に求めた内部パラメータに基づいてサンプル点の撮像面上の座標に前処理を施し、内部パラメータ行列が単位行列で表現されるようなカメラについての推定と等価な推定とする。

レンズ歪みの補正を行ったデータを使っているため、上記 (3) における推定結果は、

- (3-a) アスペクト比は 1 対 1 であり；
- (3-b) 軸の非直行性歪みは 0 であり；
- (3-c) 回転行列は単位行列であり；
- (3-d) 並進移動ベクトルは零ベクトルである；

という条件を満たしているべきであり、また、上記 (4) における推定結果は、

- (4-a) 内部パラメータ行列は単位行列であり；
- (4-b) 基準を逆にして求めた回転行列は互いに逆行列の関係にあり；
- (4-c) 基準を逆にして求めた位置ベクトルは互いに逆の関係にある；

という条件を満たしているべきである。実験では、Levenberg-Marquardt 法による非線形の最適化を行わない状態で、(4-c) 以外の条件に関して有効数字 4 桁程度の精度で上記の条件を満たすことができた。

以上のようにして隣接するカメラ間の相対関係を求めた後に、撮影システム全体として 1 つに統合された座標系での表現を求める。手順としては、次のようになる。

- (1) 基準となるカメラを決める。
- (2) 上記、基準カメラからの市街地距離が短い順に、基準カメラの座標系での外部パラメータの表現を求める。複数の経路で外部パラメータが求められる場合には、それらの経路でそれぞれ求めた値の平均値（回転行列の場合、回転角度としての平均）を採用する。
- (3) 基準カメラの座標系での表現から基準座標系での表現に変換するための写像を、各カメラの外部パラメータに対して施す。

計測データ C は、実際にこの撮影システムで撮影したいシーンを撮影したものである。

用途としては、重複して撮影している領域に存在する被写体までの距離の算出と、同領域における合成映像内のずれを補正するためのパラメータの算出である。前者については後に検討するため、ここでは説明を省略する。後者は、上述の外部パラメータの推定誤差に起因するずれや、後に述べる仮想被写体と実際の被写体とが一致しないことに起因するずれを、映像面での 2 次元平行移動によって目立たなくすることを目的としたものである。エッジ抽出とテンプレートマッチングによる補正パラメータの自動算出も試みたが、安定した結果を得ることができなかつたため、今回は手動でマッチングを行い、補正パラメータを求めた。

4 撮影システムと表示システムの接続

先に述べたシステム構成から明らかのように、本研究で構築したシステムでは撮影システムと表示システムの画角が一致していない。これは、カメラやプロジェクタの単体での性能が異なってしまうことと、マルチカメラシステムにおける各カメラの配置の自由度がマルチプロジェクタシステムにおける各プロジェクタのそれに比して低いことが主な原因である。[続 2001]のシステムのように制約の緩い表示システム側を制約の厳しい撮影システム側に合わせて設計することでこの問題を回避することは可能であるが、没入感などの別の要因から表示システムの仕様が決まってしまうことも多いため、度々繰り返しになるが、本研究では表示システムの仕様ありきという条件の下でこの問題の解決を図った。

我々の考え方は、次のようなものである。

まず、表示システムに関しては、与えられた理想視点位置から見たときに意図した形状に見えるように、補正パラメータを生成する。これは、

- (1) 表示システムの設計値から決まる理想的な形状補正；
- (2) プロジェクタの設置誤差・レンズ歪みの補正；
- (3) プロジェクタの個体差による色調や明るさの違いの補正；
- (4) スクリーン上で複数の映像が重なっている部分の明るさの補正；
- (5) スクリーンの設置誤差・形状歪みの補正；

を同時に補正するパラメータであり、デジタルカメラを使った計測ベースの方法で求める。

次に、撮影システムに関しては、表示システムのアナロジーを考える。すなわち、撮影された世界全体に写っている被写体が実は 1 つの被写体（「仮想被写体」と呼ぶ）であるとして、複数のカメラでその仮想被写体を写していると考える。2 つのカメラの光学中心が一致していないことから、被写体の見え隠れの関係が異なってくる部分については不整合が生じてしまうが、どちらのカメラにも写っている被写体に関しては正しい仮想被写体を生成することで不整合の発生を抑えることができる。計測データに基づいて、あるいは、ヒューリスティックな手法によって該仮想被写体を決定することで、幾何学的な補正用のパラメータを求める。

以上の準備が終了した段階で、表示システム側も撮影システム側もそれぞれ 1 点の理想視点を原点としたそれぞれの座標系における光線方向を考えて定義した写像が決定される。

その上で、表示システム側の座標系と撮影システム側の座標系とをつなぐ写像を考える。

このような写像のうちもっとも単純なものとしては、回転のみで定義される写像が考えられる。これは、表示システムで表示可能な領域のうち、撮影システムの撮影領域だけに映像を表示し、残りの部分には「黒」を表示することに相当する。この写像を使った場合、次の2つの問題が発生する。第1の問題は、表示領域の一部に映像が表示されないことによる、没入感の低下である。ユーザが仮想世界に囲まれている際、たとえ周辺視の領域であっても、スクリーンのような現実世界の物体を知覚させることは没入感の大幅な低下につながり、システム全体の価値を下げる原因となる。第2の問題は、プロジェクタ1台に対してカメラ1台の撮影領域では賄えなくなるケースが発生するということである。この写像を使った場合に、プロジェクタとカメラの1対1対応を成立させるためには、(いずれも理想視点位置から見た場合に) 個々のカメラの撮影領域が対応するプロジェクタの表示領域を覆っている必要があるが、一般には、先に述べたように、この条件が成り立たない場合が多い。

したがって、表示システム側の座標系と撮影システム側の座標系とをつなぐ写像には、何らかの工夫が必要となる。そのひとつの例に、擬似射影モデル[Moriya 2002]の考え方がある。擬似射影モデルとは、撮影時の実際の射影モデルと、得られた画像の射影モデルとを異なるものと解釈して扱う考え方で、見た目の違和感を少なく、かつシステムティックに補正を施すことを目的としたものである。この方法によってプロジェクタとカメラの1対1の対応が保証されれば、撮影座標系と表示座標系で見え方に違いは生じるものの、重複撮影領域も含めて、その他の点においては不具合のない、好適な写像を生成することができる。

以上の検討内容に基づいてシステムを実装し、実験を行った。実験では、仮想被写体の形状を円筒面として、実測により該円筒の半径を決定し、補正パラメータの算出を行った。図5に比較的遠距離にある被写体での実験結果を、図6に近距離にある被写体での実験結果を示す。

図5(a)は6台のカメラで撮影した映像をそのまま6台のプロジェクタで表示したものであり、図5(b)はシステム全体のキャリブレーションで求めた補正パラメータによって映像を補正して表示したものである。この例では、遠距離にある被写体に対して、滑らかにつながった広視野角な映像が得られていることがわかる。

また、図6(a)では、近距離においても一見滑らかにつながった広視野角な映像が得られているように見えるが、図6(b)に見られるように、想定した距離以外の部分に被写体がある場合、つまり、仮想被写体と実際の被写体とが大きくかけ離れている場合、原理どおり、映像上では重複して撮影されている部分が大きくずれてしまうことがわかる。

これらの実験結果から、本研究で検討したシステムは、遠距離の被写体を撮影する監視システムや、近距離の被写体を撮影する場合には、被写体までの距離が静的に定まる会議システムのような用途に適しており、コンサートやスポーツの試合など主要な被写体までの距離が刻一刻と変化するような用途に対しては、被写体までの距離をリアルタイムに測れるようなセンサによる補助が必要であると言える。このセンサとして撮影システムとしての複数カメラをそのまま用いる場合の技術について、(まずはリアルタイム性を重視せず、

会議システムのような準静的なアプリケーションを想定したものを) 検討する。

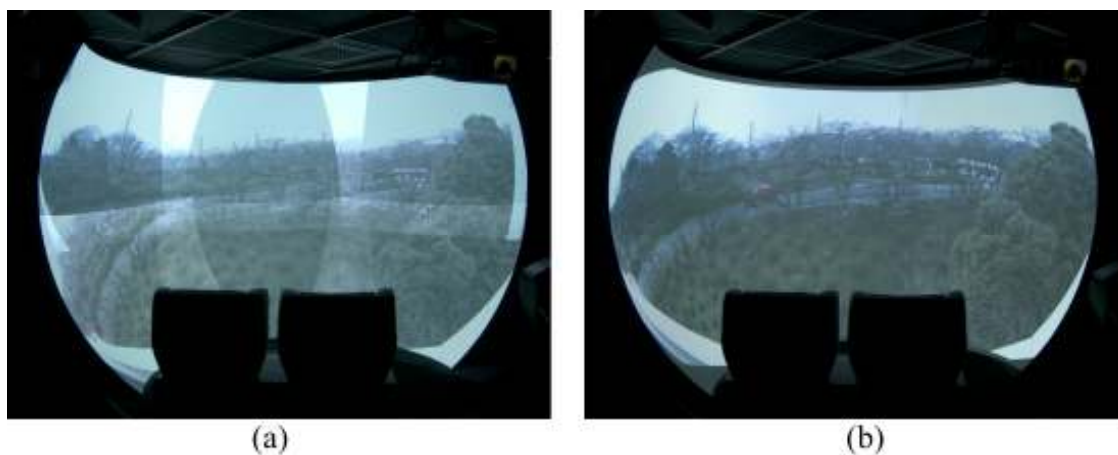


図 5 遠距離にある被写体での実験結果



図 6 近距離にある被写体での実験結果

5 被写体位置の推定

本研究の撮影システムでは、撮影された世界全体に写るすべての被写体を仮想的に1つの被写体であるとみなし、各カメラはその仮想被写体を撮影しているものと考えた。単一視点制約を満たさないマルチカメラシステムにおける合成映像上での不整合をできるだけ小さくしようというのが狙いである。ここまでの議論では、ヒューリスティクスを用いて仮想被写体の生成を行っていたが、ここでは、ヒューリスティクスではなく、自動で仮想被写体を生成するための技術の検討を行う。具体的には、重複撮影領域中の被写体が3次元空間中でどのような曲面上に位置するのかを、各カメラの撮影した映像と、カメラ間の位置関係をはじめとする外部パラメータとを用いて推定する技術の検討を行った。

5.1 概要

仮想被写体を1つの曲面と考え、それを仮想被写体面と呼ぶ。この仮想被写体面に合成

画像をマッピングすることになる。そこで、マルチカメラシステムから得られる重複撮影領域を持つ画像の重複撮影領域を利用して、より正確な仮想被写体面を推定する。具体的には、仮想被写体面を、仮想被写体の形状を代表する点群の全てを通る滑らかな曲面であると考え、最初に仮想被写体の形状を代表する点群をカメラの撮影した画像内で2次元的に推定し、次にカメラ同士の位置関係を用いてその点群の3次元位置ベクトルを推定し、そうして得られた3次元空間中の点群の全てを通る滑らかな曲面を生成する。

まず、互いに重複撮影領域を持つ2枚1組の画像に対して、ステレオマッチングを適用する。ここで、着目点同士の対応関係を推定する方法には、テンプレートマッチング（画像相関法など）[高木 1991]が基本的であり、非常によく用いられているが、全ての画素に時間のかかる処理を行う必要があることや、対応を取る対象となる画素が密なことによって誤対応が増えることから、ここでは、構造マッチングの1つであるポイントパターンマッチング[Barnard 1980]を用いることにする。ポイントパターンマッチングとは、各画像において、周囲と比較して色の変化の大きい画素を特徴点として複数個抽出し、2枚の画像内で、同じ被写体の同じ部分を示すと推定される特徴点の組を対応点ペアとして選択し、その位置関係に対して三角測量を行って3次元位置ベクトルを得る方法である。この方法は、被写体の角を特徴点として検出することができるため特徴点が密になり過ぎず、誤対応の危険が減ることや、画像の歪みに対して不変量である色の濃淡レベルや撮影環境の変化に影響されにくい周囲との色の変化分を用いているため撮影上の誤差を吸収しやすいことや、仮想被写体面を生成するためには多数の対応よりも仮想被写体の形状を代表する特徴点同士の対応が取ることが重要であることなどの点において今回の目的に適していると考えられる。このステレオマッチングから得られた重複撮影領域中の対応点ペアとして選択された3次元空間中の特徴点を制御点と呼ぶことにする。

次に、上記制御点群を全て通過する滑らかな曲面を生成する。曲面の生成方法としては、ベジエ曲面やBスプライン曲面[杉原 1995]が有名であるが、ここでは曲面の制御点群が必ずしも規則的な分布にはならないという条件下においても、制御点群を全て通過する滑らかな曲面を生成できる方法[Murotani 2002]を利用する。

5.2 仮想被写体面の生成方法

ステレオマッチングで推定した、制御点群の3次元位置ベクトルを用いて、仮想被写体面を生成する方法について詳しく述べる。仮想被写体面は、全ての制御点群を通る滑らかな曲面である。しかし、単純にベジエ曲面やBスプライン曲面を生成しても、全ての制御点群を通るという条件を満たす曲面は得られない。そこで、制御点群を頂点とするメッシュを生成し、そのメッシュに対して生成した曲面パッチ同士を滑らかにつなぐ、曲面の生成法[Murotani 2002]を利用する。この曲面生成法は、制御点が規則的に並んでいない場合にも有効である。

以下、全ての制御点を通る滑らかな曲面を生成する方法について述べる。まず、制御点群を x - y 平面上に落とした点群に対して Delaunay 三角形分割を適用し、得られた x - y 平面上の三角形メッシュに z 座標を与えた3次元空間中の三角形メッシュを生成する。次に、図7のように、各三角形パッチに15点の制御点（このうち、三角形の3頂点は元からある制御点）を設定し、4次ベジエ曲面パッチを生成する。新しく加えた12点の仮設の制御点（こ

これらの制御点を通る必要はない) の位置を、主曲率の 2 乗和の総和が最小になるという条件下での有限要素法や、隣り合う 2 つのパッチ間が滑らかにつながるといった条件下での線形結合を用いて決定することにより、それらの制御点によって定まる 4 次ベジエ曲面パッチ同士が滑らかにつながり、全体としては全ての制御点を通る滑らかな曲面が生成される。このような仮想被写体面の生成法を全ての重複撮影領域に対して適用することで、カメラの台数分の大きさの仮想被写体面が得られる。

以上、仮想被写体面の生成方法について述べたが、実際にマルチカメラシステムに適用する場合には、仮想被写体面上に合成画像をマッピングするため、任意の視点から仮想被写体面上の任意の点までの距離を導出してやればよい。

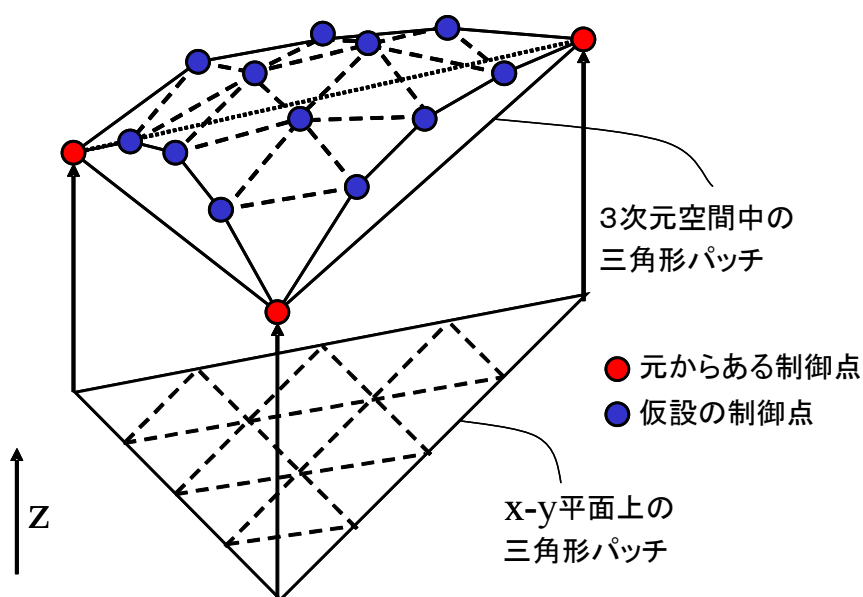


図7 三角形パッチに設定する仮設の制御点

6 まとめ

光学中心が一致していないマルチカメラシステムで撮影した映像を、画像処理によって見た目に違和感なく繋ぎ合わせ、広い視野角を持った1枚の高解像度映像を得る方法と、そのような撮影システムと没入型ディスプレイとの連携方法について、検討・開発および実験を行った。遠距離あるいは近距離にある被写体の撮影を実際に行うことによって、今後本研究成果を適用するに適したアプリケーション分野や、現在のシステムの課題とその改善方法について有効な知見を得ることができた。

(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

1 プロトタイプシステムの全体構成

1.1 概要

(1) で述べた複数カメラによる撮影映像のリアルタイム変換技術と、課題イで開発した L2Shaper を用いたアプリケーション技術の検証を目的とし、映像監視システムのプロト

タイプを開発した。

本プロトタイプシステムでは、広視野角映像を、異なる方向を向いた 6 台のビデオカメラで撮影する。さらに撮影された映像は、6 台プロジェクタから構成されるマルチプロジェクタシステム（プロジェクタアレイシステム）によって、水平方向 180 度の表示範囲を有する球面形状スクリーンに表示される。

6 台のビデオカメラ映像は圧縮して伝送され、受信側ではデコード映像が同期して再生される。また、本アプリケーション以外の通信トラフィックが存在する一般的な通信路を想定して、Jam トラフィック発生装置を通信路に付加し、通信路の帯域不足や伝送遅延を模擬できる環境を構成した。このような帯域不足や伝送遅延による映像品質の低下を抑止するために、通信路に課題イで開発した L2Shaper を設置し、L2 パケット制御機能を利用した映像品質確保の実験を行った。

1.2 仕様

映像入力装置として、6 台の NTSC カラービデオカメラを用いて、水平画角約 160 度、垂直画角約 90 度の撮影画角を実現する。

映像表示環境としては、6 台の 1024×768 解像度を持つ DLP 方式プロジェクタを用い、水平方向 180 度の表示範囲を持つ球面スクリーンへ投影できるプロジェクタアレイシステムを用いる。

単一視点制約を満たしていない 6 台のカメラを用いた映像をシームレスに接続するためには、(1) に述べた幾何変形処理を必要とする。このような幾何変形処理は、表示系のシステムが有する専用画像処理ボード内でリアルタイムに実現する。

また、遠隔監視を想定したアプリケーションでは、撮影ビデオ映像を監視ポイントまでに如何に伝送するかという伝送技術が重要となる。さらに、本プロトタイプシステムのように、複数の NTSC ビデオカメラ映像を、表示環境においてシームレスに接続して表示するためには、伝送した映像情報を映像信号として再生する時点で、撮影時の映像間同期が再現される必要がある。そのために本プロトタイプシステムでは、6 台ビデオカメラは、相互に同期を取った撮影系となる構成とした上で、専用ハードウェアによる MPEG2 圧縮により、1 カメラ当たり約 6 Mbps の伝送レートでのネットワーク伝送を可能とし、なおかつ受信側で MPEG2 を復号する際に、6 台の受信復号器間で映像フレーム番号情報を共有し、互いに同期のとれた映像信号を再現可能とした。

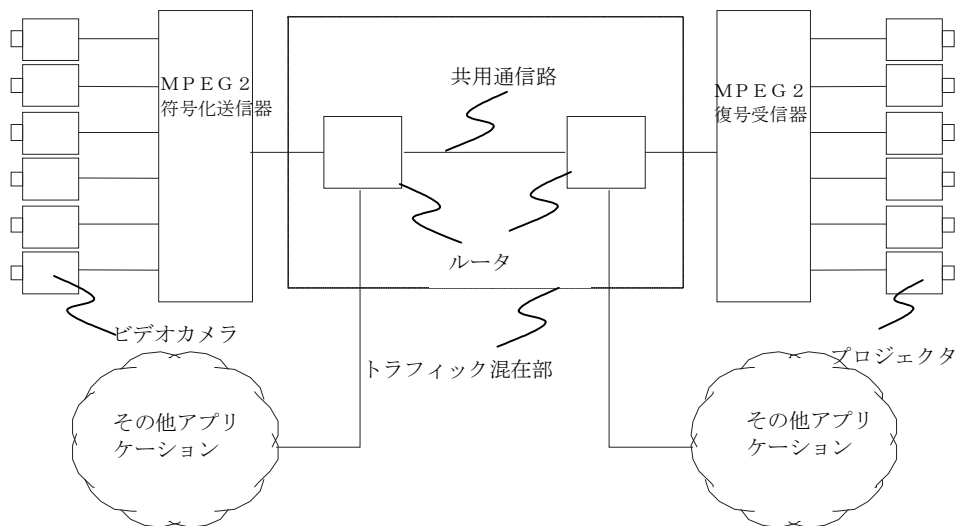


図 8 マルチカメラ映像伝送の想定通信環境

監視用ビデオカメラ映像を監視サイトまで伝送するネットワーク経路としては、図 8 に示すように、本プロトタイプ以外のアプリケーションのトラフィックとの混在環境での使用を想定する。マルチカメラ映像品質保証を目的として課題イで開発した L2Shaper を用いて、監視用ビデオカメラ映像情報の通信パケットの優先的帯域割り当てを可能とする。試作した L2Shaper が 1 台なので、想定通信環境のトラフィックの混在する部分を 1 台の L2Shaper に置き換えた通信路を、プロトタイプでは設定し実験するものとする。

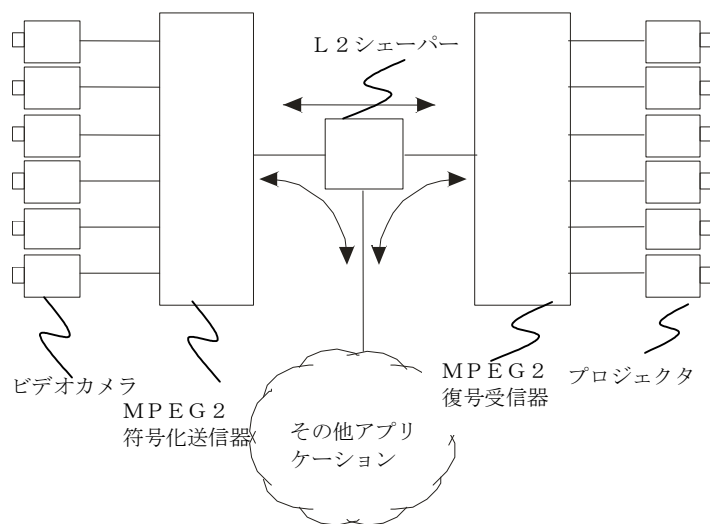


図 9 プロトタイプシステムの通信環境

1.3 ハードウェア構成

前節に述べた遠隔監視システムプロトタイプの仕様を満足する実験システムの、ハードウェア構成は以下のとおりである。映像撮影用の 6 台のビデオカメラは 3 台ずつ 1 台のカメラドライブユニットに接続し、合計 2 台のカメラドライブユニットを使用する。各カメ

ラドライブユニットは、カメラへの電源供給と映像信号制御を行っている。同一のカメラドライブユニットに接続されているビデオカメラの撮影は同期して行われる設定としている。カメラドライブユニット間は、一方のカメラドライブユニットを基準として、その同期信号を他方のカメラドライブユニットへ外部同期信号として供給し、ユニット間の同期制御を実施している。これにより、6台のビデオカメラは、互いに同期した撮影が可能となる。

カメラドライブユニットから出力される映像信号は NTSC カラーコンポジット信号である。この映像信号は、MPEG2 映像圧縮送信装置へ入力される。MPEG2 映像符号化送信装置は、入力映像信号は互いに同期が取れていることを前提として、6台すべての映像信号が有効になった時点から、6台に共通のフレーム番号管理を開始し、映像信号を MPEG2 ストリームへ圧縮する際に、該フレーム番号を付加して、受信側でフレーム番号を参照できるようにしている。NTSC カラーコンポジット映像は、それぞれ独立に圧縮され、一ストリーム当たり約 6 Mbps に圧縮されて、ネットワークへ送信される。6本の映像ストリーム通信は、6本独立なストリームとして制御され、MPEG2 復号受信装置へ宛てて送信されている。6系統のストリームには、それぞれ受信と送信に個別の IP アドレスを割当てて、1系統の送信と受信の IP アドレスのペアは、事前に静的に設定しておく。

相互にフレーム番号管理を行っている、6系統の MPEG2 映像符号化送信装置からのネットワーク通信信号は、一旦、L2 ハブで一本にまとめられた後、課題イで試作した L2Shaper へ接続している。L2Shaper には、その他に、MPEG2 映像復号受信装置と、ネットワーク通信路を共用しているアプリケーションを模擬するための L2 トラフィックジェネレータが接続されている。

MPEG2 映像符号化送信装置から送信された通信パケットは、L2Shaper を介して、MPEG2 映像復号受信装置へ伝送される。MPEG2 映像復号受信装置は、6系統の MPEG2 映像復号器を有しており、それぞれの復号器には、事前に設定してある系統毎の送信と受信の対応関係にある送信側ストリームの通信パケットが供給される。各映像ストリーム系統では、独立に映像の復号が行われる。各映像ストリーム中の映像フレームには、送信時に付加された共通フレーム番号が付けられており、6台の映像復号器は、相互にフレーム番号を参照し合い、6系統で同一のフレーム番号を持つ映像が映像信号として出力されるように、制御されている。これにより、MPEG2 映像復号受信装置から出力される 6系統の NTSC カラーコンポジット信号は、それぞれの信号中の映像フレームが相互に同期がとれた映像信号となる。

MPEG2 映像復号受信装置から出力された 6系統の NTSC カラーコンポジット信号は、それぞれ、6台のスキャンコンバータへ入力され、各 NTSC カラーコンポジット信号は、1024×768 解像度を持つ RGB カラーコンポーネント信号へ変換され、プロジェクタアレイシステムへ供給される。

1024×768 解像度を持つ RGB カラーコンポーネント信号は、リアルタイムで任意幾何変形と画素単位での任意色変調が可能なプロジェクタアレイ専用画像ボードへ供給される。6台の該専用画像処理ボードは、入力された映像信号と同一規格の映像信号を出力し、それぞれ、6台の 1024×768 解像度を持つ DLP プロジェクタへ映像信号を供給する。専用画像

処理ボードでは、プロジェクタから投影される映像がスクリーン上でシームレスにつながった映像となるための、幾何変形処理と色変調処理がリアルタイムで行われる。この幾何変形処理は、単一視点制約を満たさない 6 台のカメラ映像を特定の被写体面ではシームレスにつながるようにする幾何変形処理と、プロジェクタとスクリーンの設置位置関係に依存して、投射映像がスクリーン上でつながるための幾何変形処理が合成されたものとなる。

図 10 に MPEG2 符号化送信装置、MPEG2 復号受信装置を示す。6 系統の映像信号を、それぞれ約 6 Mbps の MPEG2 ストリームに符号化し、復号受信装置の対応するチャンネルへ送信する。



図 10 MPEG2 符号化送信装置



図 11 L2Shaper 試作機（左）およびトラフィック発生装置（右）

MPEG2 のコーデックは、6 系統の映像ストリーム毎に 1 対ずつ使用される。

MPEG2 符号化送信装置と、MPEG2 復号受信装置は、それぞれ、100 Mbps のイーサネットケーブルで、図 11 に示す L2 レベルフロー制御機能を有する L2 スイッチ(以下、L2Shaper と記す)試作機に接続されている。L2Shaper の詳細仕様は、課題イの報告書で述べる。さらに、L2Shaper には、図 11 に示す Jam トラフィック発生装置が接続されている。

MPEG2 映像復号受信装置から出力された 6 系統の NTSC カラーコンポジット信号は、それぞれ、スキャンコンバータにより、1024×768 解像度を持つ RGB カラーコンポーネン

ト信号へ変換され、IPD（プロジェクタアレイシステム）へ供給される。

2 実験および検討

2.1 実験

2.1.1 概要

本実験の目的は、(1)プロトタイプシステムを用いた実験アプリケーションを実際に稼働させ、映像品質や監視し易さなどの主観的な側面の課題と、パフォーマンスの実用化に向けた課題を抽出することと、(2)課題イで開発したネットワーク制御機能の本実験伝送路に組み込み、実験アプリケーションでの運用上の効果の確認と課題を抽出することである。

この目的に沿って、まず応用シーンを想定して、それに類似した環境を模擬したプロトタイプシステムを稼働させた。次に模擬監視者に本プロトタイプシステムを用いて、屋外広視野監視の擬似体験を実施した。そこにおいて、課題イで開発したネットワーク制御機能の各種設定による、本プロトタイプシステムの動作性能へ及ぼす効果と影響を検証した。

2.1.2 遠隔監視映像配信と表示デモンストレーション実験

遠隔での屋外監視を想定して、屋外の監視対象エリアとして、本研究サイトである(株)日立製作所システム開発研究所敷地内を、研究棟最上階から、カメラ 6 台によるマルチカメラにより水平視野角 160 度で撮影し、撮影映像を圧縮符号化し、建物に既設の LAN ケーブルを利用して IPD 設置フロアまで伝送し、IPD 上に監視映像をシームレスにリアルタイム合成表示した。図 12 にマルチカメラ設置場所と設置状況を示す。



図 12 カメラ設置個所と設置状況

マルチカメラから撮影映像を IPD へ表示する実験システムのシステム構成は、前節で述べたとおりである。マルチカメラ映像は、研究棟の 2F に設置した IPD 表示設備まで、100 Mbps のイーサネット LAN を用いて伝送する。

本実験では、監視エリア内で発生する敷地内へ入退する人物や自動車などの動きを監視する監視アプリケーションとしての機能を、以下の項目に注目して検証する。

- (1) 6 台の同一視点制約を満たさないマルチカメラ映像を擬似的に同一視点から観測した映像として表示しているが、継目部分の画質に問題はないか。
- (2) 6 台のカメラ映像は単一の広角映像と比べてどのような良さを持つか。
- (3) マルチカメラ撮影映像の同期再生を可能とする、映像圧縮・伝送・再生装置を開発し用

いたことにより、同期ズレによる、動物体の継目部分での映像のズレは防げているかどうか。

- (4) IPD 装置へのマルチカメラ映像の再生環境が、想定している監視業務にとって、状況把握という観点からどのような効果を持っているか。

2.1.3 QoS 制御による安定な映像配信の実験

課題イで開発した L2 レベルフロー制御機能を有する L2 スイッチ（以下、L2Shaper と記す）を用い、MPEG2 符号化送信装置、MPEG2 復号受信装置、トラフィック発生装置をネットワークで相互に接続し、ネットワークデータをアプリケーション間で相互に伝送する。ネットワークを共用している遠隔監視アプリケーション以外のアプリケーションからのトラフィックは、限られた帯域の通信回線において、遠隔監視アプリケーションの伝送映像データの損失を発生させる。このようなデータ損失は、表示映像品質の劣化現象を招く。その他のアプリケーションに相当する通信パケットを、トラフィック発生装置を用いて発生させ、このようなデータ損失現象を再現する。

L2Shaper の以下に説明するフロー制御機能を用いて、遠隔監視アプリケーションのレイヤ 2 での帯域を優先的に確保し、遠隔監視アプリケーションの映像品質確保が可能なことを実験で検証する。

図 13 に、実験に用いた L2Shaper の機能構成を示す。3 本のネットワークインタフェースを持ち、インタフェース毎に、優先制御モジュールを備える。各優先制御モジュールは、フロー識別モジュールとキュー制御モジュールより構成される。

フロー識別モジュールは、1 入力 N 出力のモジュールであり、レイヤ 2 以上の情報を利用して、種々の条件に基づいてパケットフローの識別を行い、フローごとに後段へ続く各キューへパケットを送付する。

キュー制御モジュールは、N 入力 1 出力のモジュールであり、N 個の入力を指定したキュー制御アルゴリズムに従って合流させて出力する。指定可能なキュー制御アルゴリズムには、重み付きラウンドロビン(WRR)アルゴリズムと、優先度別送信アルゴリズム(PQ)を選択できる。実験では、この 2 つのアルゴリズムをそれぞれ用いてフローの状態を計測する。

このような優先制御モジュールを持つ 3 本のネットワークストリームには、MPEG2 符号化送信機、MPEG2 復号受信機、トラフィック発生装置がそれぞれ接続されている。この状態で、以下の項目の実験を行った。

- [1] トラフィック発生装置を停止して、MPEG2 符号化送信器と、MPEG2 復号受信器の間を伝送しているパケット数をカウントし、トラフィックを計測する。
- [2] フロー制御無しで Jam トラフィックを混入した場合の映像品質の変化の観測
- [3] フロー識別ルールを設定し、重み付きラウンドロビン(WRR)アルゴリズムによるキュー制御を実施した場合の、Jam トラフィック混入による影響調査
- [4] フロー識別ルールを設定し、優先度別送信アルゴリズム(PQ)アルゴリズムによるキュー制御を実施した場合の、Jam トラフィック混入による影響調査

上記実験項目[3]と[4]でのフロー識別ルールは、表1に示すようなキューへの振り分け条件による設定をした。ただし、MPEG2 符号化送信装置には、IP アドレス 192.168.7.101~106 を付与し、MPEG2 復号受信装置には、IP アドレス 192.168.7.201~206 を付与した。

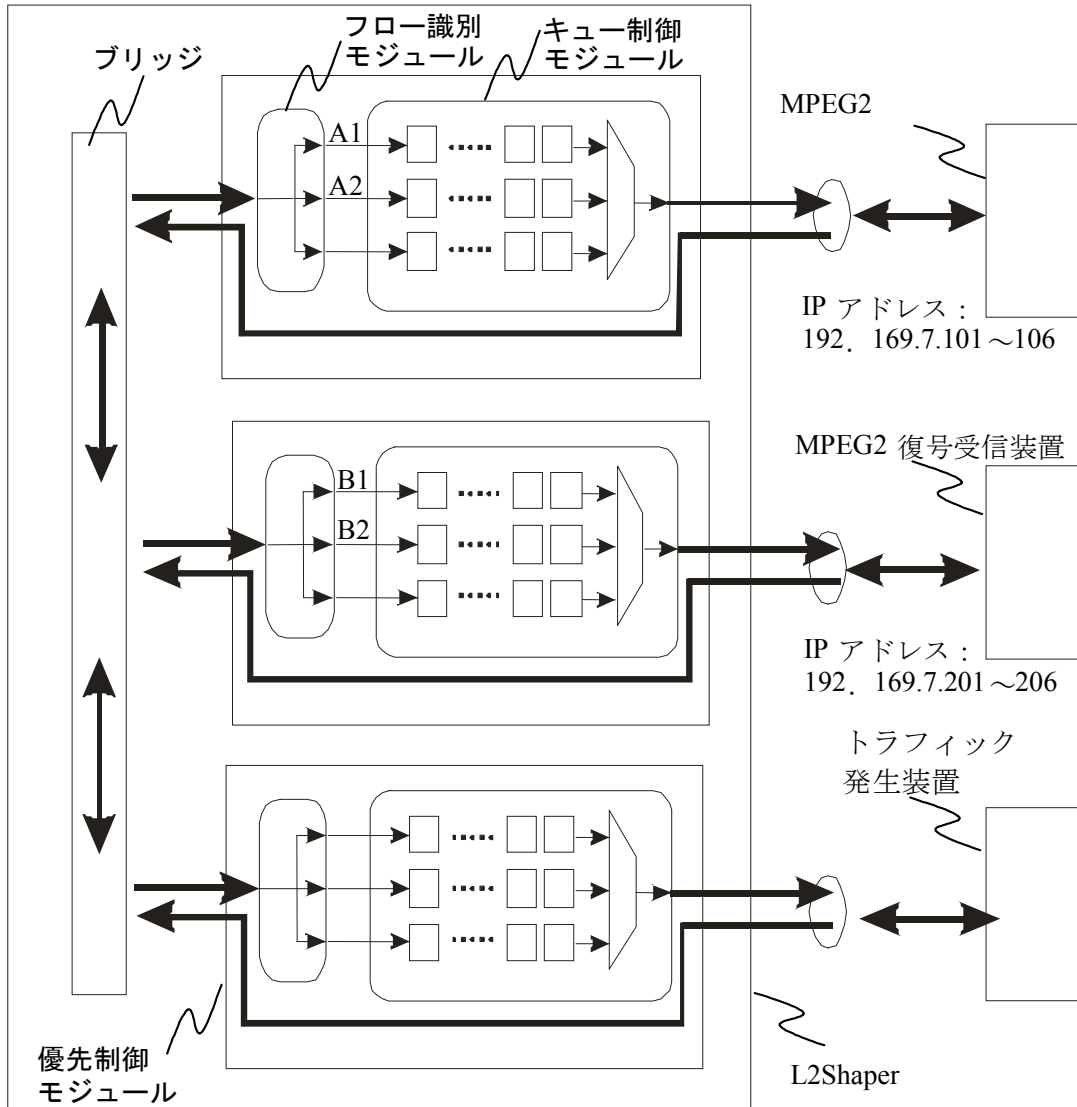


図 13 L2Shaper 機能構成

表 1 実験項目 [3] [4] に使用したフロー識別ルール

識別名	フロー識別ルール
A 1	宛先 IP アドレスが、192.168.7.101~106 を持つパケット
A 2	A 1 以外のパケット
B 1	宛先 IP アドレスが、192.168.7.201~206 を持つパケット
B 2	B 1 以外のパケット

2.2 実験の結果

2.2.1 遠隔監視映像配信と表示デモンストレーション実験

実験項目別の結果を以下に記す。

- (1) 6 台の同一視点制約を満たさないマルチカメラ映像を擬似的に同一視点から観測した映像として表示しているが、継目部分の画質に問題はないか。

IPD 設置場所まで伝送されてきたマルチカメラ撮影映像にを、画像処理せず、球面スクリーン上に投影した状態と、伝送されて来た映像に対して、以下に記す画像処理を専用画像処理器によりリアルタイム処理し、球面スクリーン上に投影した結果を、図 14 に示す。同一視点制約を満たしていないながらも、6 台のカメラ映像がスクリーン上でシームレスに一体化されて表示されている事が確認できる。シームレス接続のためにカメラ映像に施された画像処理は、事前に計測して決定しておいたカメラのレンズ歪、透視変換パラメータ、カメラ相互の位置の情報に基づいた幾何変形処理と、プロジェクタ投影画面の相互の重複関係に基づいて決定した重複部のエッジブレンディング処理の二種類からなる。

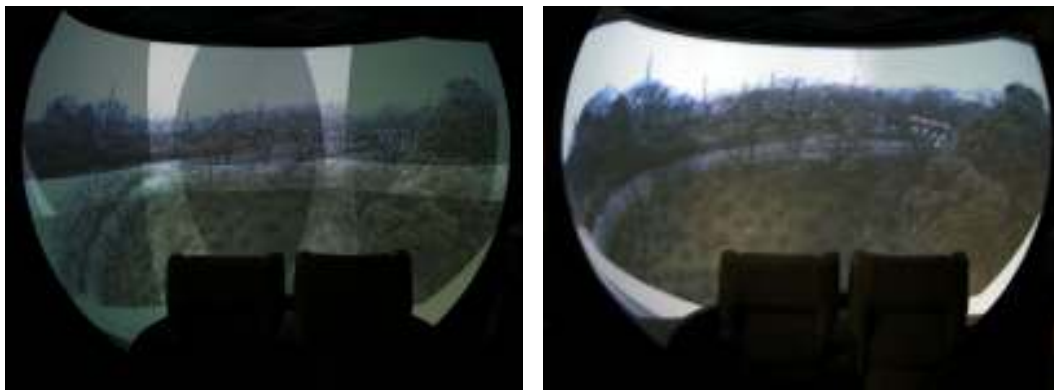


図 14 マルチカメラ映像の無変換投影とマルチカメラ映像シームレス投影

- (2) 6 台のカメラ映像は単一の広角映像と比べてどのような良さを持つか。

単一光学系による広角映像の撮影例を図 15 に示す。図 15 は、魚眼レンズを装着したデジタルカメラを用いて、本実験で設置したマルチカメラの設置位置と同一個所から撮影した監視エリアである。本実験のマルチカメラによる撮影映像と比較すると、魚眼レンズによる撮影映像は、空の部分と撮影地点の真下の地面部分との合計が映像全体の 1/2 近くのエリアを占めている。監視の目的には重要でない部分が、撮像面のかなりの部分を占めるため、相対的に監視対象エリアの有効解像度が減少している。すなわち、限られた解像度の一部しか注目エリアの撮影に利用できず、情報取得効率が悪化する。一方、マルチカメ

ラによる広視野角映像撮影では、撮影対象エリアの形状に合わせて、カメラ配置を設計でき、なおかつ複数の撮像デバイスを用いるため、撮影対象エリアを効率的に撮影できることが分る。



図 15 魚眼レンズによる監視領域の撮影映像

(3) マルチカメラ撮影映像の同期再生を可能とする、映像圧縮・伝送・再生装置を開発し用いたことにより、同期ズレによる、移動体の継目部分での映像のズレは防げているかどうか。

マルチカメラを構成するそれぞれのカメラの撮影タイミングは、カメラドライブユニットが制御し同期制御されている。また、映像符号化送信装置においても、同期情報を伝送データに付与し、映像復号受信装置側では同期情報を基に撮影映像を同期再生しているために、スクリーン上に投影されている 6 台のカメラの映像は、正しく同一時刻の映像が再生されているはずである。このような同期の確認は、移動体を撮影し、スクリーンの継目部分で該移動体が正しい形状で表示されていることを確認することで可能である。実験では、消防車を時速 30 Km で監視エリア内を移動させ移動体とした。この移動体の形状が継目部分で正しく表示されていることを確認することで、同期が取れていることを確認した。その例を図 16 に示す。これらの画像に写っている消防車は、どちらも形状に差がないことが分る。すなわち、同期ズレが発生していないことが確認できた。



図 16 継目通過直前の消防車と、継目部分通過中の消防車の表示映像

(4) IPD 装置へのマルチカメラ映像の再生環境が、想定している監視業務にとって、状況把握という観点からどのような効果を持っているか。

視界全体に監視範囲全域が捉えられるため、監視域内で発生している事象把握が、容易であった。また、発生した事象部分への注視と詳しい観察は、単にその部分に注目すれば、単眼の広角カメラ撮影映像に比して高い解像度の画像であるため、十分な情報が得られる場合が多い。単眼のカメラによる監視のように、画面のズームアップ操作が不要で、監視作業が容易化する。また、注視点を拡大してさらに詳しく観測したい場合、電子的ズーム操作(画像処理による拡大処理)が一般であるが、IPD との組合せによる監視作業では、注視点画面を、拡大鏡(オペラグラスなど)により画面を拡大して観測することで同様なことが実施可能である。

IPD を共有して複数人で監視作業を実施している場合には、前記のズームアップ操作に代る拡大鏡の使用による特定部分の拡大観察を観測者毎に独立に実施できるので、誰か一人によるズームアップなどの画面操作により画面状態を変更されることがなく、操作の優先権制御などが不要である、さらに、監視者にとっても、各自独立した監視を干渉されることがないという効果がある。

2.2.2 QoS 制御による安定な映像配信の実験

実験項目別の実験結果を以下に記す。

[実験 1] トラフィック発生装置を停止して、MPEG2 符号化送信器と、MPEG2 復号受信器の間を伝送しているパケット数をカウントし、トラフィックを計測する。

表に、MPEG2 コーデック間で送受されているトラフィック量の計測結果を記す。トラフィック量は、単位時間当りのパケット数(pps)と単位時間当りの通信量(bps)で表示している。試作した L2Shaper では、基本的にはパケット単位の処理をしており、計測できるのは、単位時間当りのパケット数である。パケット長は、アプリケーションに依存して一定していないので、単位時間当りの通信量は、主要な通信のパケットをキャプチャして、そのパケット長を基に推定した通信量となる。

フロー識別名 B の、MPEG2 符号器から送信されるパケットは、MPEG2 エンコードされた映像データで、チャンネル当り 7 Mbps 弱に圧縮されていることが分る。復号器から符号器へ送信されているパケットは、フロー制御のためのデータであり、チャンネル当り 6 Mbps 程度ある。

表 2 MPEG2 コーデック間のトラフィック計測値

フロー種別	計測トラフィック	
	パケット数/秒 (pps)	メガビット/秒 (M bps)
A(復号器→符号器)	440	3.7
B(符号器→復号器)	3524	41.7

[実験 2] フロー制御無しで Jam トラフィックを混入した場合の映像品質の変化の観測

L2 トラフィックジェネレータを用いて発生した Jam トラフィックを通信路に混入した場合の、MPEG2 符号器および復号器へ伝送されるパケット量の計測値と画面状態を、表 3 へ記す。トラフィック計測値は、1 秒当りのパケット数で表記してある。Jam トラフィック量が 60 Mbps を越えた時点で、IPD 画面上にノイズの発生が始まった。

表3 Jam トラフィック混入時のトラフィック計測値(pps)と画面状態

フロー種別	Jam トラフィック量 (M bps)				
	0	30	40	50	60
A(復号器→符号器)	440	2940	3781	4616	5413
B(符号器→復号器)	3524	6030	6863	7714	8481
画面状態	良好	良好	良好	良好	ノイズ発生

表2に示したように識別子Bのフロー(符号器→復号器)は41.7Mbpsあり、これにJamトラフィックが60Mbps以上加わると、伝送限界の100Mbpsを越えてしまう。そのため、容量限界を越えた分だけJamトラフィックおよび映像データが区別なく破棄されたため、映像データの損失結果として画面ノイズが発生したと考えられる。

[実験3] フロー識別ルールを設定し、重み付きラウンドロビン(WRR)アルゴリズムによるキュー制御を実施した場合のJamトラフィック混入による影響調査

実験3においては、各ネットワークインタフェースの送信伝送容量を70Mbpsに設定して実験した。なぜならば、一般にネットワークインタフェースが表示仕様どおりに正確に100Mbpsの伝送容量を実現できている保証がないので、十分な安全マージンを見て70Mbpsの性能を持つネットワークインタフェースとして動作させ、フロー制御実施下での各フローを計測したからである。識別子A1, A2, B1, B2の各フロー識別ルールは、表4に示したようなキューへの振り分け条件とした。各キューの重み付けは、MPEG2符号器から復号器への映像に関するデータ伝送容量(フロー識別名B1)に、実験1で計測した41.7Mbps以上の容量を確保しておく必要があり、総伝送容量70Mbpsの70%に当る49Mbpsを割り当てた。一方、フローAのMPEG2伝送装置間の通信には過大な容量割り付けにあたるが、フローBと同様の重み付けを設定した。

表4 実験項目[3][4]に使用したフロー識別ルール

識別名	フロー識別ルール
A1	宛先IPアドレスが、192.168.7.101~106を持つパケット
A2	A1以外のパケット
B1	宛先IPアドレスが、192.168.7.201~206を持つパケット
B2	B1以外のパケット

表5に、重み付きラウンドロビン制御(WRR)実施時のトラフィック計測値と画面状態を示す。フロー識別子別のトラフィック計測値は、1秒当りのパケット数で表記してある。Jamトラフィック容量によらず、識別名A1とB1のフロー、即ちMPEG2コーデックに関するデータ伝送部分は、変動せず一定量となっている。そのため、Jamトラフィックの影響を受けずに、画面状態も常に良好となっている。識別名A2およびB2のフロー、即ち、MPEG2コーデックに無関係なフローは、Jamトラフィック量が、30Mbps以上では一定となっている。これは、 $70\text{Mbps} \times 30\% = 21\text{Mbps}$ の容量制限により、30Mbps以上のJamトラフィックデータに関しては、21Mbpsを越えるデータが破棄された結果、一定となっている。

表5 重み付きラウンドロビン制御実施時のトラフィック計測値(pps)と画面状態
(各チャネルのスループットは最大 70 Mbps に制限)

フロー種別 (重み付け%)	Jam トラフィック量 (Mbps)					
	0	20	30	40	50	60
A1 (70%)	433	433	432	532	432	432
A2 (30%)	0	1672	1756	1756	1756	1756
B1 (70%)	3531	3531	3531	3531	3531	3531
B2 (30%)	0	1672	1756	1756	1756	1756
画面状態	良好	良好	良好	良好	良好	良好

[実験 4] フロー識別ルールを設定し、優先度別送信アルゴリズム(PQ)アルゴリズムによるキュー制御を実施した場合の、Jam トラフィック混入による影響調査

実験 4 においても実験 3 と同様に各ネットワークインターフェースの送信伝送容量を 70 Mbps に設定した。識別子 A1, A2, B1, B2 の各フロー識別ルールも、実験 3 と同様のキュー振り分け条件とする。フローの優先度は各フロー共 MPEG2 コーデックに関するデータに相当する、フロー識別名 A1 と B1 をそれぞれ、優先度 1 とし、その他のフローをそれ以下の優先度とした。

表 6 に、プライオリティキュー制御実施時のトラフィック計測値と画面状態を示す。フロー識別子別のトラフィック計測値は、1 秒当りのパケット数で表記してある。Jam トラフィック容量によらず、識別名 A1 と B1 のフロー、即ち MPEG2 コーデックに関するデータ伝送部分は、変動せず一定量となっている。そのため、Jam トラフィックの影響を受けずに、画面状態も常に良好となっている。識別名 A2 のフローは、Jam トラフィック量の増加に比例して増加し続けている。これは、A1 の容量が、最大 3.7 Mbps までしか必要としないため、残りの回線容量 66.7 Mbps の範囲内の Jam トラフィックが全てロスなく通過しているためである。一方、識別名 B2 のフローは、Jam トラフィック量が 30 Mbps 以上で、ほぼ一定となっている。これは、B1 の容量に 41.7 Mbps を必要とし、残りの容量 28.3 Mbps しかその他のデータ通信に使えない。そのため、30 Mbps 以上の Jam トラフィックでは、28.3 Mbps を越えるデータが捨てられ、28.3 Mbps に相当する一定のトラフィックしか計測されなくなるためである。

実験 3 の重み付きラウンドロビン制御によるフロー制御の場合のように、事前に確保すべき通信容量を推定した上で適当なマージンの持った回線容量を確保して QoS を実現する場合に比べて、優先度別送信アルゴリズムによるフロー制御では、MPEG2 コーデック関係の通信品質が保証された上で、残る通信容量全てを有効にその他の通信に利用できるという点が、通信回線の有効利用という観点では優れていると言える。

表6 プライオリティキュー制御実施時のトラフィック計測値(pps)と画面状態
(各チャネルのスループットは最大 70 Mbps に制限)

フロー種別 (優先度)	Jam トラフィック量 (Mbps)					
	0	20	30	40	50	60
A1 (1)	433	442	434	433	432	436
A2 (2)	0	1672	2508	3342	4180	5018
B1 (1)	3531	3520	3521	3523	3533	3529
B2 (2)	0	1672	2439	2446	2447	2442

画面状態	良好	良好	良好	良好	良好	良好
------	----	----	----	----	----	----

2.3 今後の課題

今回試作したシステムのように、複数のカメラ映像を用いて一画面を構成するための各映像の幾何学的補正に関しては、レンズ特性が静的であり、事前のキャリブレーションにより精度よく補正できるため、シームレス映像を生成することは比較的容易であった。一方、撮像素子自体の静的特性差以上に、被写体を的確に映像化するために被写体の状態に応じたアイリス制御や映像生成回路の動作特性に依存した動的な特性変動に起因して発生するカメラ間での輝度や色調の差によって、カメラ映像間のシームレス品質が阻害されるという課題がある。自動アイリス制御や、オートゲインコントロール(AGC)のような電子的処理による自動絞り相当の制御は、今回の実験では全て停止した。これにより各カメラの被写体の違いによる面間の輝度差の発生を防いでいる。しかし被写体の明度などが時間とともに大きく変動する場合には、事前に設定した各カメラの絞り値では的確な映像を撮影できないという問題が発生する。

このようなマルチカメラシステムでのアイリス制御の問題を解決するためには、カメラ間でアイリス制御を連動させる機構の実現が必要となる。しかしより根本的には、今回試作したシステムのように、原理的にはいくらかでも広い視野角を有する撮像系を構成可能なシステムでは、撮影映像内に多様な状態の被写体を含む。非常に暗い被写体から、非常に明るい被写体(極端には太陽のような光源)まで輝度バラツキが拡大することになる。このような場合には、高いダイナミックレンジを持つカメラを用いたシステム構築が必要となる。

次に、今回の試作したシステム以上の視野角を実現する問題を検討する。試作システムよりも更に広い視野を実現するためには、現在の配置の単純な拡張では、カメラ自身が他のカメラの視野内に入ってしまうという問題がある。立体角で最大 4π まで撮影できるカメラシステムを実現しようとする、今回の試作システムと異なり、カメラを外向きに配置する必要がある。すると、実現できるレンズ中心間の最小距離は、カメラ本体やレンズの形状に依存することになる。

現在のシームレス接続の方式では、被写体までの距離が一定であれば、いわゆる「同一視点制約」を満たしていなくとも、シームレス映像を作成できる。しかし、任意の距離の被写体に対応させるためには、なるべくレンズ中心間の距離が短い方がよい。そのためには、カメラ本体の形状やレンズの形状が小さなものを選択してシステムを構築する必要があるが、カメラの機種やレンズの選択には限界がある。今後は、このような予想される問題点を考慮して、立体角で最大 4π まで撮影できるカメラシステムを設計し、現実的に実現可能な全立体角撮影システムの到達可能な性能を見積りたい。

続いて、今回試作した映像圧縮同期伝送装置の性能に関して考察する。映像を MPEG2 符号化する場合には、イントラフレームの補間による圧縮処理が行なわれており、複数イントラフレームを受信した後でないと復号できない場合がある。そのため、通常 5, 6 フレーム分に相当するデータのバッファリングを必要とする。さらに、ネットワーク通信のジッターやワンダーの影響を取り除くために一定量のバッファリングを行うことが考えられる。しかし、このような処理では遅延が発生する。試作したシステムでは、最低限のバッ

ファリングに加えて、画面間の同期を取る必要性や、試作システムということで動作マージンを高めに設定したために、1秒弱の遅延が発生している。高いリアルタイム性を要求される場合以外、この程度の遅延は問題にならない。むしろカメラ間の映像再生同期の品質が重要となる。しかし、遠隔会議システムのように双方向のコミュニケーションが必要な場合、遅延が数百ミリ秒を越えると使い勝手を大きく損う要因となる。さらに双方向通信のため、遅延の影響が往復で二倍に拡大されて感じられてしまう。現在の試作システムは、監視等の一方向映像伝送応用では実用的と言える。しかし、遠隔会議システムのような双方向コミュニケーションなど、その他の応用に拡張する場合には、遅延を短縮することが重要な課題となる。

5-1-3 おわりに

高臨場感ディスプレイを対象としたコンテンツ制作やアプリケーション開発では、撮影から編集、配信、表示に至るまでのそれぞれのプロセスにおいて、極めて多くの技術を組み合わせ応用させていく必要がある。本研究では、そこでの基本的な課題を洗い出し、それらを解決させる基盤技術を開発・蓄積した。これらは、今後の具体アプリケーションの開発やコンテンツ制作において有効に利用されていくものとする。

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

5-2-1 はじめに

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

本研究は、特にアクセス網内に構築されつつあるコンテンツ配信ネットワーク(CDN)でのリアルタイム性アプリケーションに対する通信品質保証を中心に、アクセス網での通信品質制御技術を研究開発するものであり、本研究テーマでは、階層フロー識別型通信品質制御の有効性を評価することを目的としている。

階層フロー識別型通信品質制御は、アプリケーション毎のフローや、ユーザ毎のパケットフローなど複数種のフローを識別し、識別されたそれぞれのフロー毎に帯域制御や優先転送制御を階層的に施すことにより、広帯域ネットワーク内で要求に応じた通信品質を確保するものである。

本年度は、前年度までの調査研究結果に基づき、以下の仕様を満たすアクセスゲートウェイのプロトタイプを検討・設計・開発技術を実施した。

- (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実施
- (2) 通信品質保証制御は、優先制御および帯域制御が可能
- (3) 1000フロー以上の通信品質制御の設定、管理が可能
- (4) フローの最大帯域は 100 Mbps が可能

また、平成14年度に方式検討を行ったアプリケーション対応通信品質制御技術の更なる検討を行い、検証用実験コンテンツを開発して、アプリケーション対応通信品質制御技術の効果を検証した。

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

本研究では、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、広域イーサネットの実現技術の高度化に起因する、管理・運用の複雑化問題を解決するために、通信品質制御・管理技術の有効性を評価することを目的とした。

本テーマにおける通信品質制御・管理技術は、通信品質制御技術や Tag stacking や EoMPLS などを用い、マルチベンダ化、大規模化するネットワークにおける、装置設定の省力化並びに人為的ミスの低減、及び制御結果であるところの通信品質の把握を可能とするものである。

- (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な VLAN 管理サーバを開発した。具体的には、平成14年度に開発した L2Shaper 対応機種依存ドライバをベースとして、Extreme 社製イーサスイッチを制御可能とすると共に、ベンダ依存設計モジュール管理部に、機種振り分け論理を組み込んだ。これにより、L2Shaper と Extreme 社製スイッチの混成網を管理可能とした。
- (2) (1) のプロトタイプソフトウェアを、実際に L2Shaper 及び Extreme 社製イーサスイッチとから構成されるネットワークに接続し、組み合わせで動作させ、設定時間の観点から本方式の実用性を検証した。

(3) (1) のプロトタイプソフトウェアに対し、実運用を想定した多数のフローを設定し、管理対象オブジェクト数の観点から、本研究の管理方式の妥当性を検証した。

また、アクセスゲートウェイの、ネットワークプロセッサと呼ばれる専用ハードウェア上に、フロー計測機能を実装し、ギガビットイーサネット回線上のフローの計測が可能であることを確認した。

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

本研究テーマでは、IP ネットワークとイーサネットワークスイッチで構成されるイーサネットワークを接続し、相互の品質制御情報をマッピングすることにより End-End での通信品質を保証することを目的としている。

本年度は、上記目的に対して、平成14年度までに検討した課題イ-1の通信品質制御方式及び課題イ-3のIPネットワークとイーサネットワークの連携方式を弊社ブロードバンドアクセスサーバAGに適用するための詳細検討を行い、その通信品質制御方式及び連携方式に準じた機能をAGに実装し評価試験を行った。

5-2-2 実施内容

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

第1部 階層フロー識別型通信品質制御

平成15年度は、平成14年度に開発した L2Shaper プロトタイプの改良を行い、L2Shaper で優先制御および帯域制御機能を実装し、プロトタイプの性能評価を行った。

L2Shaper に関しては平成14年度の報告書にて報告済みのため、詳細は省略し、優先制御と帯域制御の方式と、評価結果に関して報告する。

優先制御はキュー制御方式としてPQ(Priority Queuing)を採用することで実現し、帯域制御はキュー制御方式としてWRR(Weighted Round Robin)を採用することで実現した。以下、各方式の実装方式に関して報告する。

1 キュー制御実装方式

1.1 PQ(Priority Queuing)

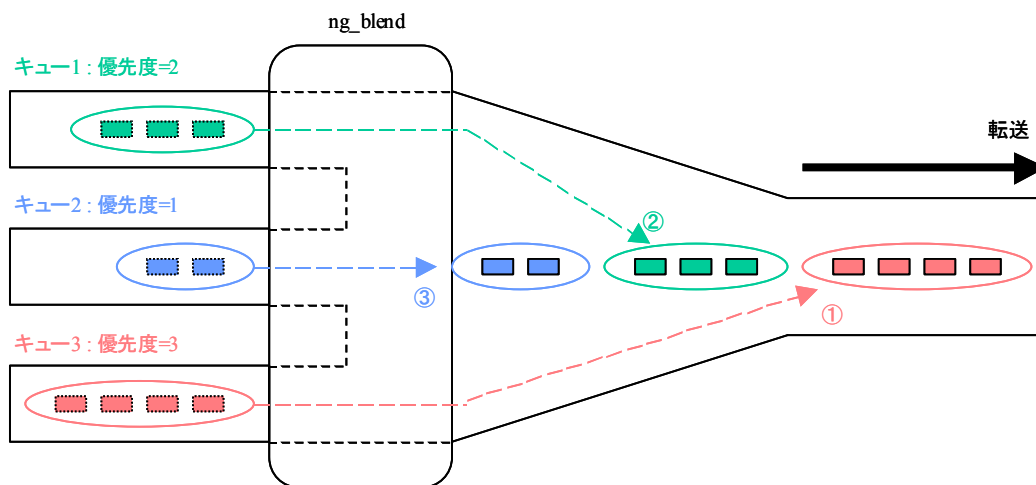


図1 PQの概要図

PQ は、複数のキューにそれぞれ優先度を与え、優先度の高いキューからパケットを送るキューイング方式である。図 1 に PQ の概要図を示す。

今回の設計では、パケットの転送処理にタイマーを用いる。そのタイマー処理で用いるノードの変数を表 1 に、キューの変数を表 2 に示す。

表 1 ノード変数

#	変数名	内容
1	bps	転送レート(bps)。この値が 0 の場合、無制限を表し、自動的に転送レートを算出し、転送する。
2	transmittable_byte	ノードにおいて、タイマー処理 1 周期で転送可能なパケットのバイト数に 16 をかけた値を格納。以下、ノードの転送可能バイト数と呼ぶ。

表 2 キュー変数

#	変数名	内容
1	priority	ユーザによって割り当てられた優先度。PQ では、この値が高いキューからパケットを送っていく。

初期化処理として、生成されているキューの優先度が高い順にソートされた、インデックスを作成する。このインデックスを参照し、以下の PQ タイマ処理を行う。

図 2 に、PQ タイマ 1 周期で行われる処理の概要フローを示す。また、フローの内容を以下に記す。

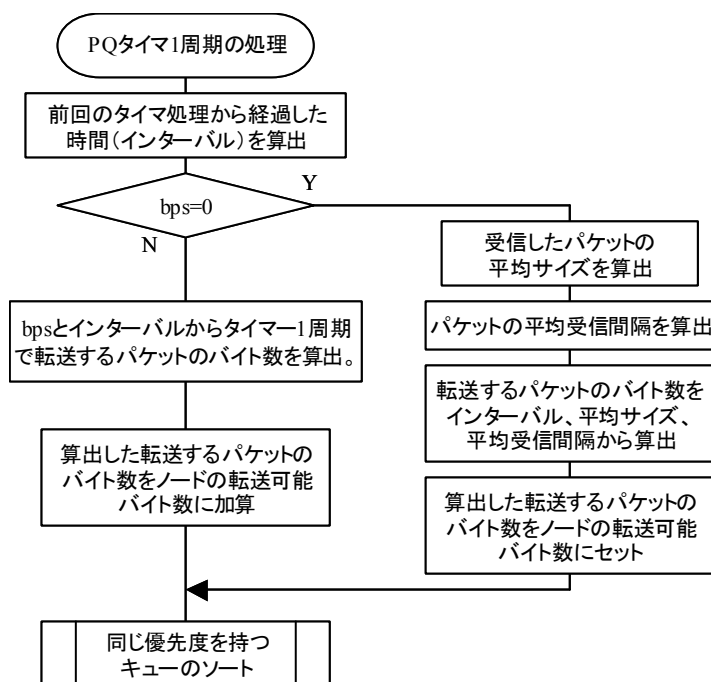


図 2 PQ タイマ 1 周期の処理の概要フロー

前回タイマー処理を行った時刻と現在の時刻から、 μsec オーダーでインターバルを算出する。

ここで、ノード変数の bps の値が 0 であれば、受信したパケットの平均受信間隔を μsec オーダーで算出し、受信パケットの平均サイズを算出する。これらの値から、以下の式を用いて、1 周期に転送するパケットのバイト数を算出する。

$$(\text{補正值}) \times (\text{インターバル}) \div (\text{平均受信間隔}) \times (\text{平均サイズ}) \quad (1)$$

これをノード変数の転送可能バイト数に格納する。ここで、ノード変数の bps に 0 以外の低い値を設定した場合、算出した転送レートが 0 とならないようにするために用いている。そのため、 bps に 0 を設定した場合にも、この値を用いている。

ノード変数の bps の値が 0 でなければ、全てのキューにおいて、転送レートとインターバルから、タイマー 1 周期でキューから転送できるパケットのバイト数を算出する。この、バイト数の計算式は、以下の通りである。

$$(\text{補正值}) \times (\text{転送レート}) \div 8 \times (\text{インターバル}) \div 1000000 \quad (2)$$

この値をノードの転送可能バイト数に加算する。ここで、ノードの転送可能バイト数の最大値は、以下の計算値により得られ、これ以上の値をとらないようにする。

$$(\text{補正值}) \times (\text{転送レート}) \div 8 \times (\text{インターバル}) \div 1000000 + (\text{MTU}) \quad (3)$$

上記の計算が終了すると、同じ優先度を持つキューのソートを行う。

図 3 に転送処理の概要フローを示す。

このフローにおいて、SQ、IQ、EQ は、インデックスを表す。最初に SQ にインデックスの先頭を、EQ にインデックスの最後を指定する。そして、IQ に SQ を指定する (①)。

ノードの転送可能バイト数が 0 以上であることを確認し、0 未満であれば、転送処理を終了する。また、IQ が示すキューにパケットがなく、IQ と EQ が同じ値であれば、終了する。IQ と EQ が異なれば、次のインデックスを IQ で指定し、キューにパケットがあるか確認する。IQ にパケットが存在すれば、転送可能バイト数から、転送するパケットのサイズに補正值をかけた値を引き、パケットを転送する。パケットを転送した後は、インデックスの更新を行う。

インデックス更新の概要図を図 4 に示す。

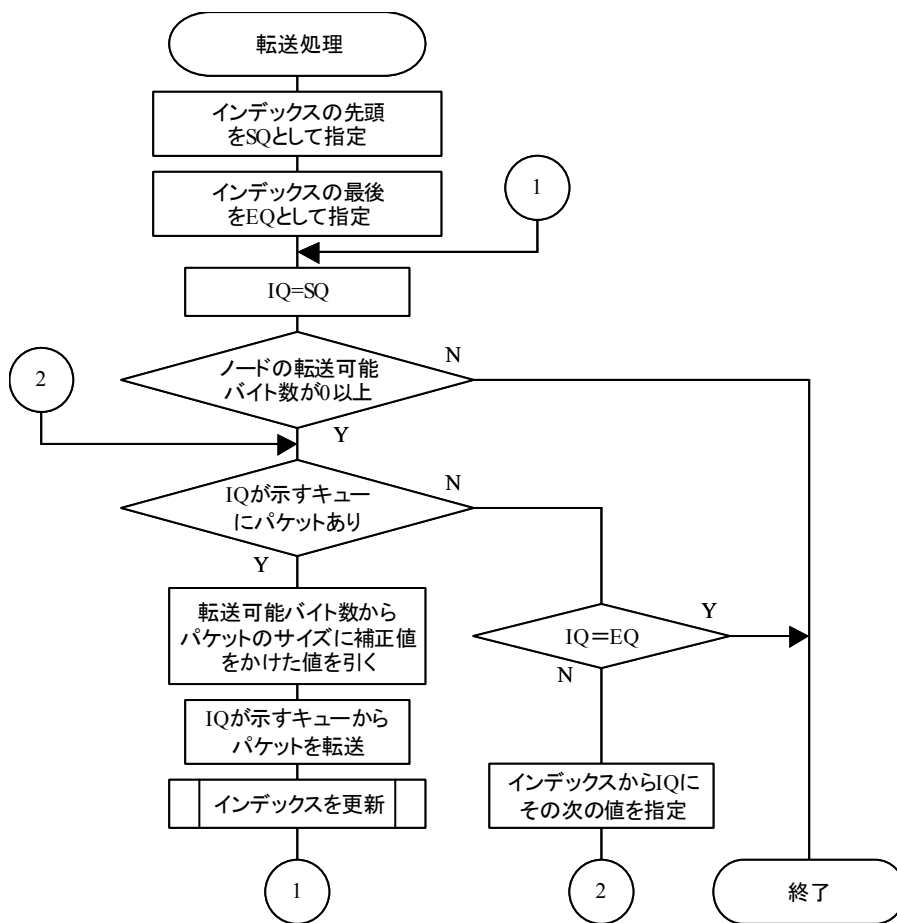


図3 転送フロー概要

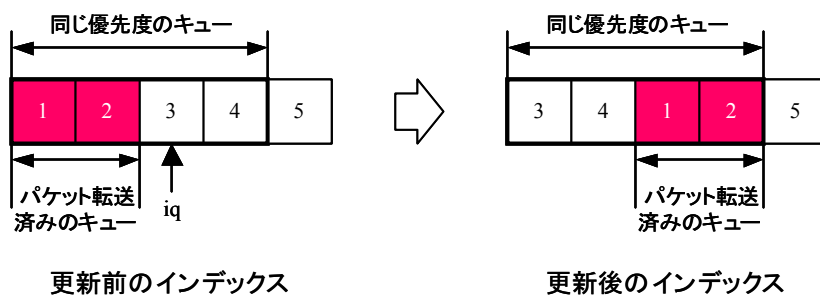


図4 キューのインデックスの更新の概要図

パケットを1つ以上転送していると、iqはパケットを転送していないキューの先頭を指す。パケットを転送していないキューをインデックスの先頭に持つことで、同じ優先度を持ったキューから、均等にパケットを転送することができる。もし、インデックスを更新しなければ、ノード変数のbpsに低い値を設定し、パケットがPQタイマー1周期で、

1 パケットしか転送できないといった状況において、同じ優先度のキューが複数あっても、毎回、同じキューからしかパケットが転送されない。インデックスの更新はこれを解消するために行うものである。

以上が、今回採用した PQ 処理方式である。

1.2 WRR(Weighted Round Robin)

WRR は、複数のキューにそれぞれ重み付けを行い、この重みを考慮した転送レートでパケットを転送するためのキューイング方式である。図 5 に WRR の概要図を示す。

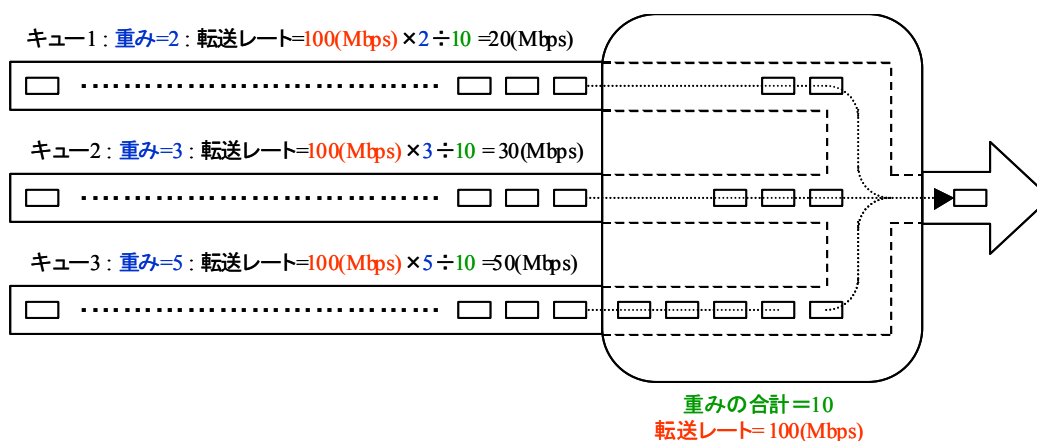


図 5 WRR の概要図

今回の設計では、パケットの転送処理にタイマーを用いる。そのタイマー処理で用いるノードの変数を表 3 に、キューの変数を表 4 に示す。

表 3 ノード変数

#	変数名	内容
1	bps	転送レート (bps)。この値が 0 の場合、無制限を表し、自動的に転送レートを算出し、転送する。
2	transmittable_byte	ノードにおいて、タイマー処理 1 周期で転送可能なパケットのバイト数に 16 をかけた値を格納。以下、ノードの転送可能バイト数と呼ぶ。
3	total_weight	各キューの weight の合計。

表 4 キュー変数

#	変数名	内容
1	weight	ユーザによって割り当てられたキューの重み。以下、重みの初期値と呼ぶ。
2	chk_weight	キュー変数の transmittable_byte を算出するために用いる値。以下、重みと呼ぶ。デフォルト値には、重みの初期値を割り当てる。
3	transmittable_byte	キューにおいて、タイマー 1 周期で転送可能なパケットのバイト数に 16 をかけた値を格納。以下、キューの転送可能バイト数と呼ぶ。

以下に、転送レートに 0 以外の値が与えられた場合に、パケット転送のタイマー処理の

周期毎に行われる処理の概要フローを図 6 に示す。

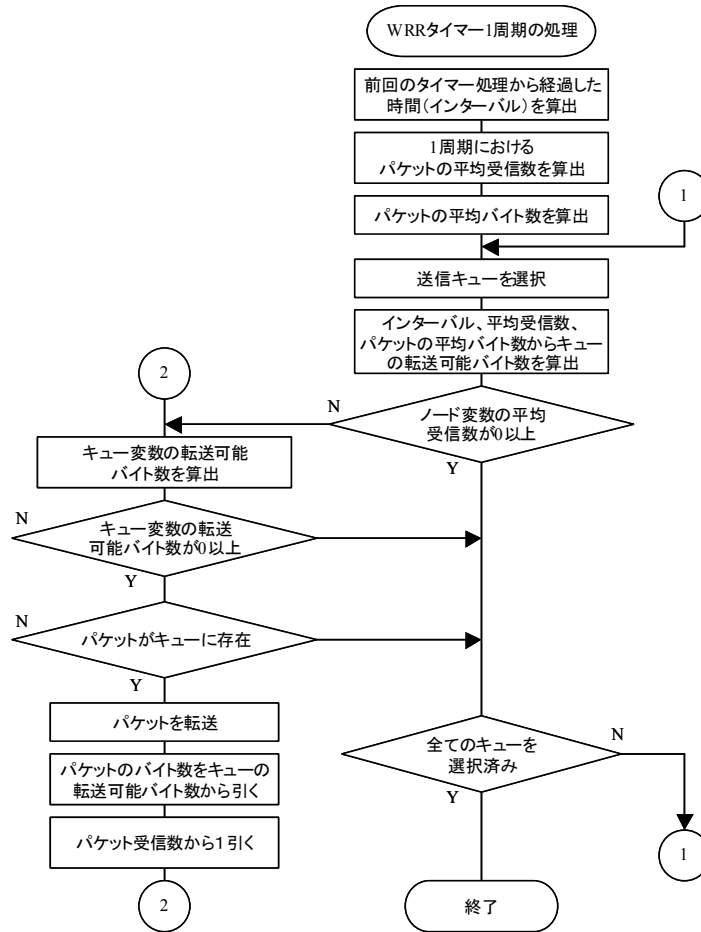


図 6 通信レートが 0 以外のパケット転送タイマー処理の概要フロー

前回タイマー処理を行った時刻と現在の時刻から、 μ sec オーダーでインターバルを算出する。

全てのキューにおいて、転送レート、キューに割り当てられた重み、インターバルから、タイマー 1 周期でキューから転送できるパケットのバイト数を 16 倍で算出する。16 倍する理由は、転送レートが小さい場合に、正しい転送レートでパケットを転送できないためである。この、バイト数の計算式は、以下の通りである。

$$16 \times (\text{転送レート}) \div 8 \times (\text{重み}) \div (\text{重みの合計}) \times (\text{インターバル}) \div 1000000 \quad (4)$$

この値が 16 以上である場合、ノード、キューの転送可能バイト数に加算し、重みに重みの初期値を設定する。16 以下であれば、そのキューの重みに重みの初期値を加算する。ここで、ノードの転送可能バイト数の最大値は、以下の計算値により得られ、これ以上の値をとらないようにする。

$$16 \times (\text{転送レート}) \div 8 \times (\text{インターバル}) \div 1000000 + (\text{MTU}) \quad (5)$$

また、キューの転送可能バイト数も同様に、以下の式で得られ、これ以上の値をとらないようにする。

$$(\text{ノードの転送可能バイト数}) \times (\text{重み}) \div (\text{重みの合計}) + (\text{MTU}) \quad (6)$$

全てのキューで上記の計算が終了すると、以下の転送処理を行う。

パケットが存在するキューを選択し、ノード、キューの転送可能バイト数が0以上である場合、ノード、キューの転送可能バイト数から、16倍した転送パケットのバイト数を引き、パケットを転送する。ここで、キューの転送可能バイト数が0未満である場合、この転送処理が終了した場合、次のパケットが存在するキューを選択し、処理を繰り返す。パケットが存在するキューがない場合、全てのキューの転送可能バイト数が0未満の場合、ノードの転送可能バイト数が0未満になった場合に、このWRRタイマー1周期の処理を終了する。

以下に、転送レートに0が与えられた場合に、パケット転送のタイマー処理の周期毎に行われる処理の概要フローを図7に示す。

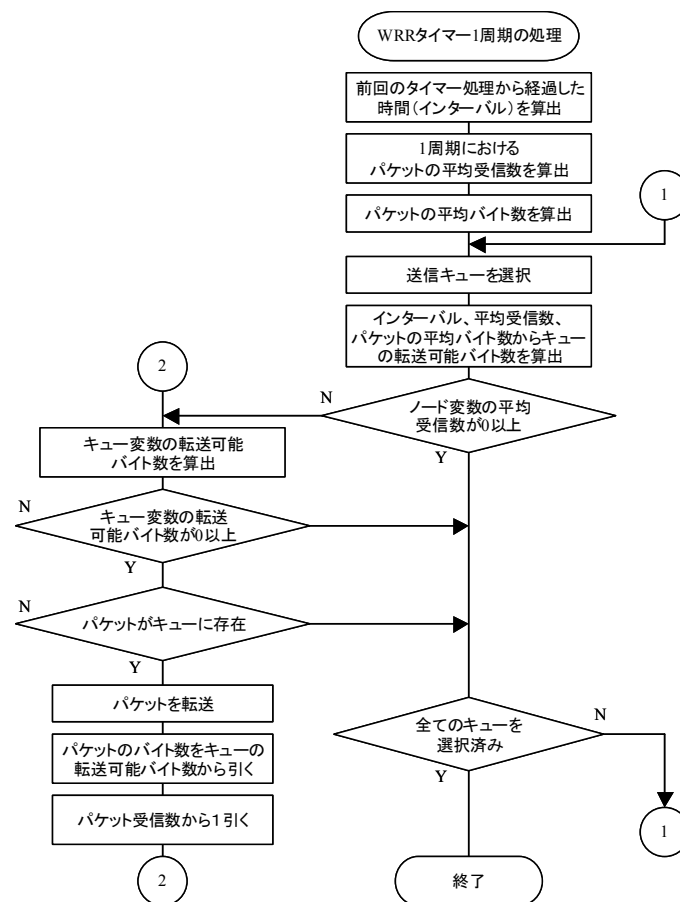


図7 転送レートが0のパケット転送タイマー処理の概要フロー

前回タイマー処理を行った時刻と現在の時刻から μ sec オーダーでインターバルを、タイマー1周期におけるパケットの平均受信数、パケットの平均バイト数を算出する。

パケットの平均受信数が 0 以下である場合、全てのキューの選択が終了した場合に以下の処理を終了する。全てのキューから 1 つキューを選択し、そのキューから転送すべきパケットのバイト数である、キューの転送可能バイト数を重み、インターバル、平均受信数、パケットの平均バイト数から算出し、その値が 0 以上である間パケットの転送を繰り返す。パケット転送後は、毎回、キューの転送可能バイト数からパケットのバイト数を、平均受信数から 1 を引く。パケットがなかった場合、また、キューの転送可能バイト数が 0 未満となった場合、次のキューを選択する。

以上が、今回採用した WRR 処理方式である。

2 キュー制御方式評価

以下に、PQ/WRR それぞれの評価結果を示す。

2.1 PQ 評価

(ア) 測定条件

今回の測定では、図 8 のように、リンク upstream0、upstream1、upstream2 を生成し、これらのリンクからパケットを受信する。その際、ng_blend のノード blend1 のノード・パラメータである bps、upstream0、upstream1、upstream2 のリンク・パラメータである priority に任意の値を設定し、パケットが、出力側のリンクである downstream から、どのように転送されていくのかを纏る。

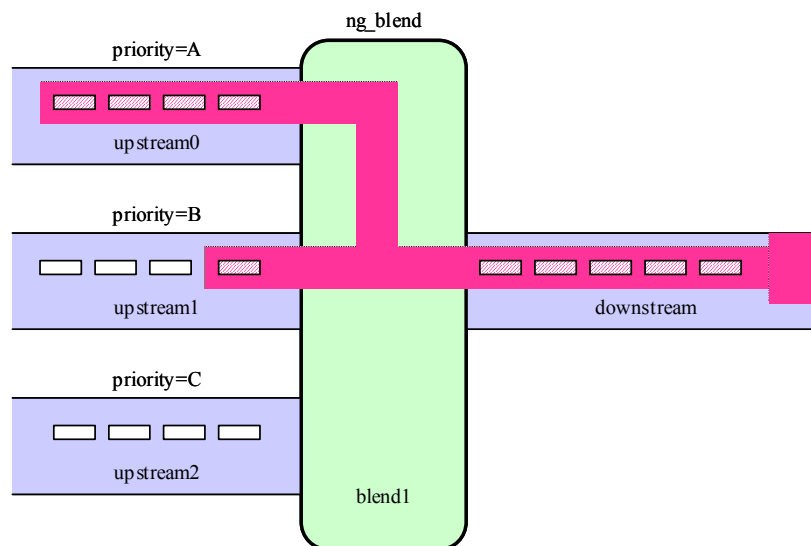


図 8 ng_blend のノードの構成

今回の測定では、上記の構成で、以下に示す 2 通りの方法で測定を行っている。これらの違いは、各リンクに設定する priority の値である。

・測定 1

測定 1 では、各リンクの priority に異なる値を設定し、測定を行う。リンク・パラメータ

の設定値は、表 5 の通りである。

表 5 測定 1 のリンク・パラメータの設定値

#	パラメータ名	設定値		
		upstream0	upstream1	upstream2
1	priority	1	2	3
2	size	100	100	100

また、測定 1 は、表 5 の設定で、パケットの受信レートを変え、表 6 の 3 通りの測定を行う。これらは、受信レートが異なっているが、各リンクが受信するパケットの割合は、同じである。また、これらのパケットのサイズは、全て 260 バイトである。

表 6 測定 1 の項目

#	項目名	パケットの受信レート(bps)		
		upstream0	upstream1	upstream2
1	測定 1-1	約 50M	約 100M	約 150M
2	測定 1-2	約 0.5M	約 1M	約 1.5M
3	測定 1-3	約 5k	約 10k	約 15k

・測定 2

測定 2 では、測定 1 とは異なり、priority の値が同じリンクが存在する場合の測定である。リンク・パラメータの設定値は、表 7 の通りである。

表 7 測定 2 のリンク・パラメータの設定

#	パラメータ名	設定値		
		upstream0	upstream1	upstream2
1	priority	1	2	2
2	size	100	100	100

また、測定 1 と同様に、測定 2 でも表 7 の設定で、パケットの受信レートを変え、表 8 の 3 通りの測定を行う。ただし、この受信レートの合計値は、測定 1 とほぼ同じである。

表 8 測定 2 の項目

#	項目名	パケットの受信レート(bps)		
		upstream0	upstream1	upstream2
1	測定 2-1	約 50M	約 100M	約 150M
2	測定 2-2	約 0.5M	約 1M	約 1.5M
3	測定 2-3	約 5k	約 10k	約 15k

(イ) 測定結果

・測定 1

リンク・パラメータの **priority** の値が、それぞれ異なる場合の測定結果を図 9、図 10、図 11 に示している。リンクごとで、**priority** の値が異なると、優先度が高いキューに格納されたパケットから先に転送される。これらの測定結果の横軸は、**blend1** に設定した **bps** の値で、縦軸は、**blend1** の **downstream** から転送されたパケットのレートを **bps** オーダーで表している。

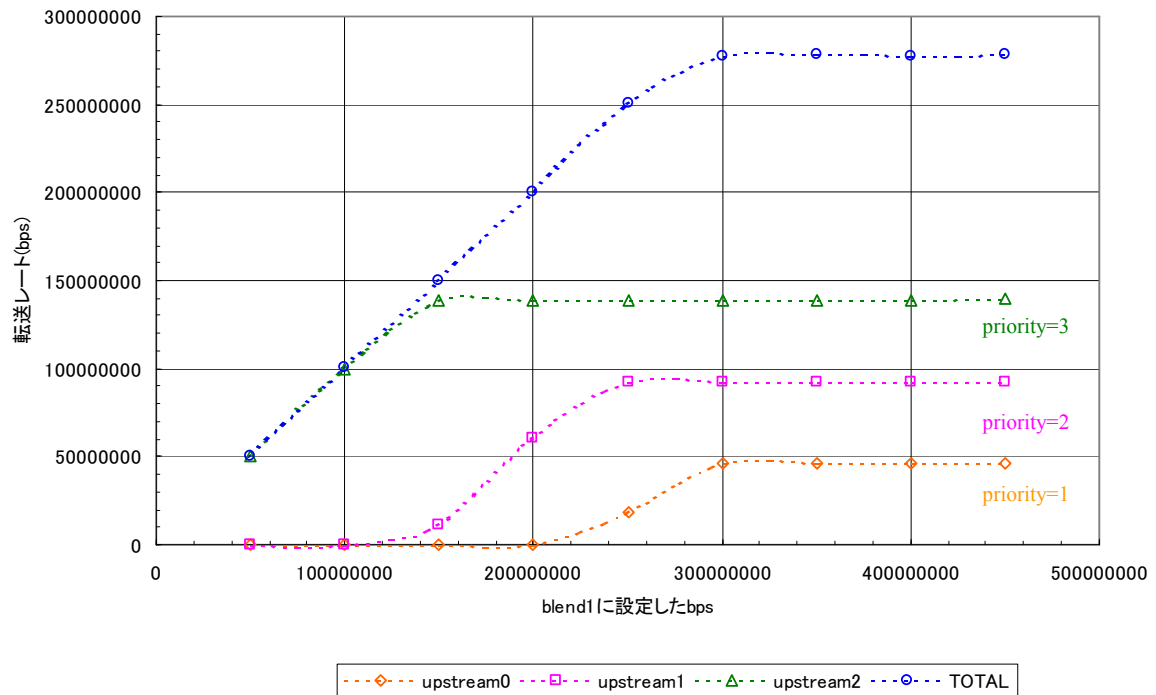


図 9 測定 1-1 の測定結果

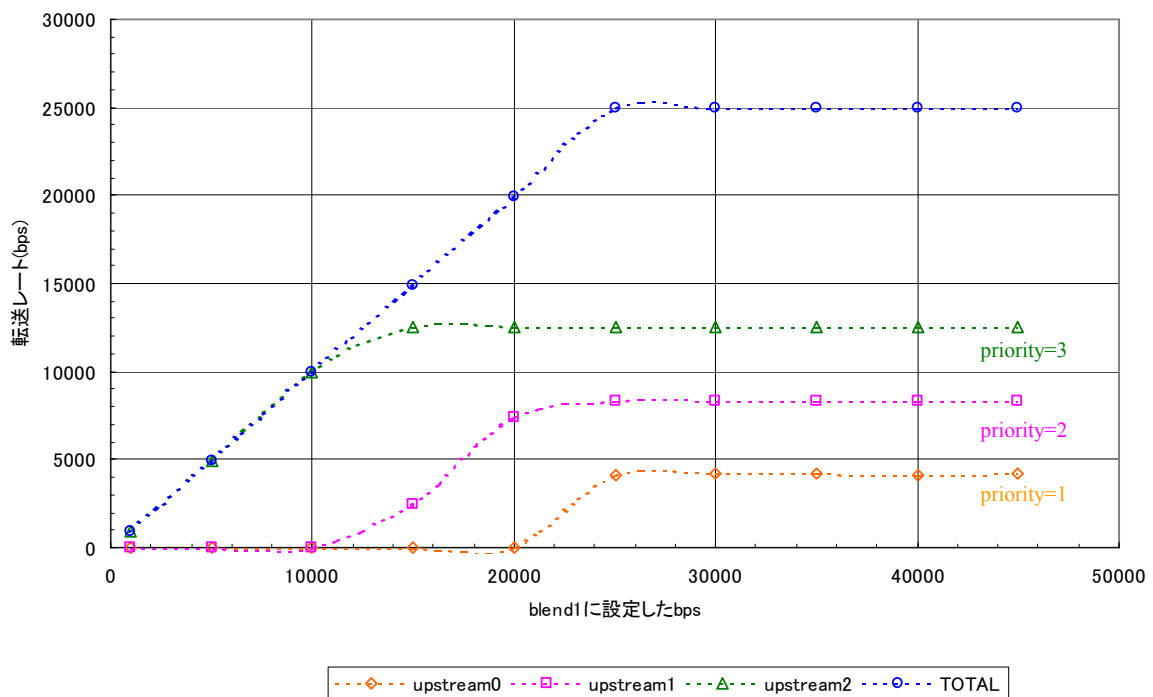


図 10 測定 1-2 の測定結果

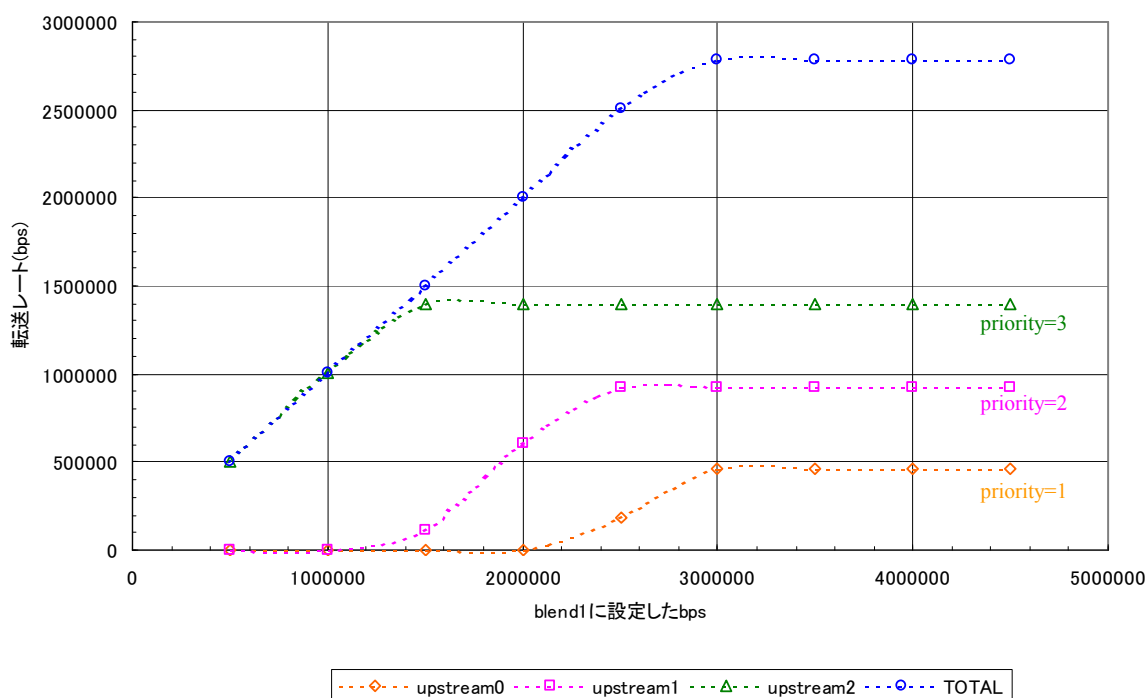


図 11 測定 1-3 の測定結果

これらの結果より、各リンクの priority に全て異なる値を設定した場合、パケットの入力レートを blend1 の bps の値が超えていなければ、受信するパケットのレートに関係なく、bps に設定した値と、ほぼ同じ転送レートで、priority が高いキューから順にパケットを転送していることが分かる。

・測定 2

リンク・パラメータの priority の値が、同じ値を持つキューが存在する場合の測定結果を図 12、図 13、図 14 に示している。同じ priority を持つキューが存在する場合、同じ priority を持つキューはそれぞれ、均等にパケットを転送する。これらの測定結果の横軸は、blend1 に設定した bps の値で、縦軸は、blend1 の downstream から転送されたパケットのレートを bps オーダーで表している。

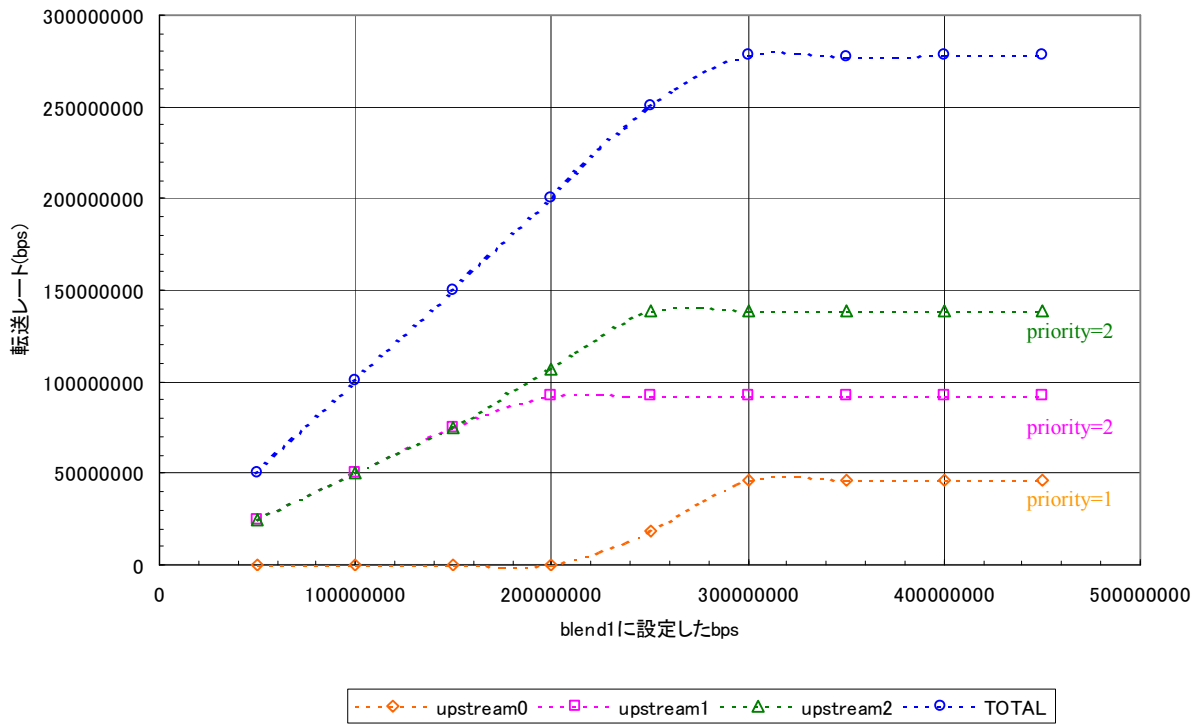


図 12 測定 2-1 の測定結果

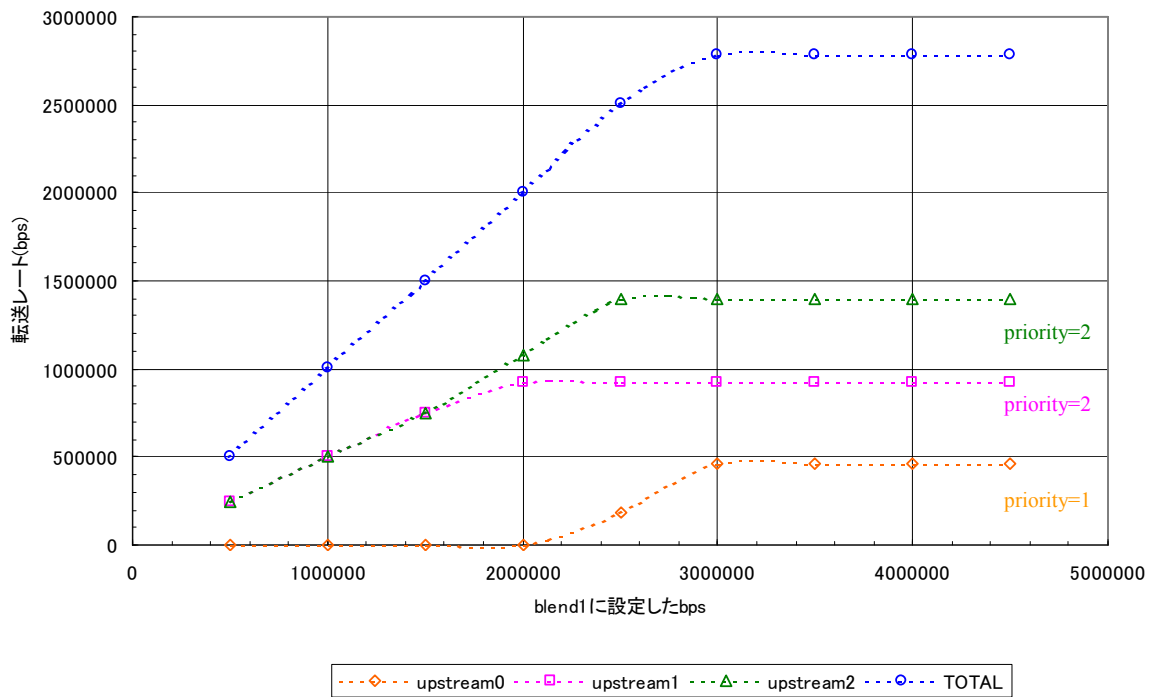


図 13 測定 2-2 の測定結果

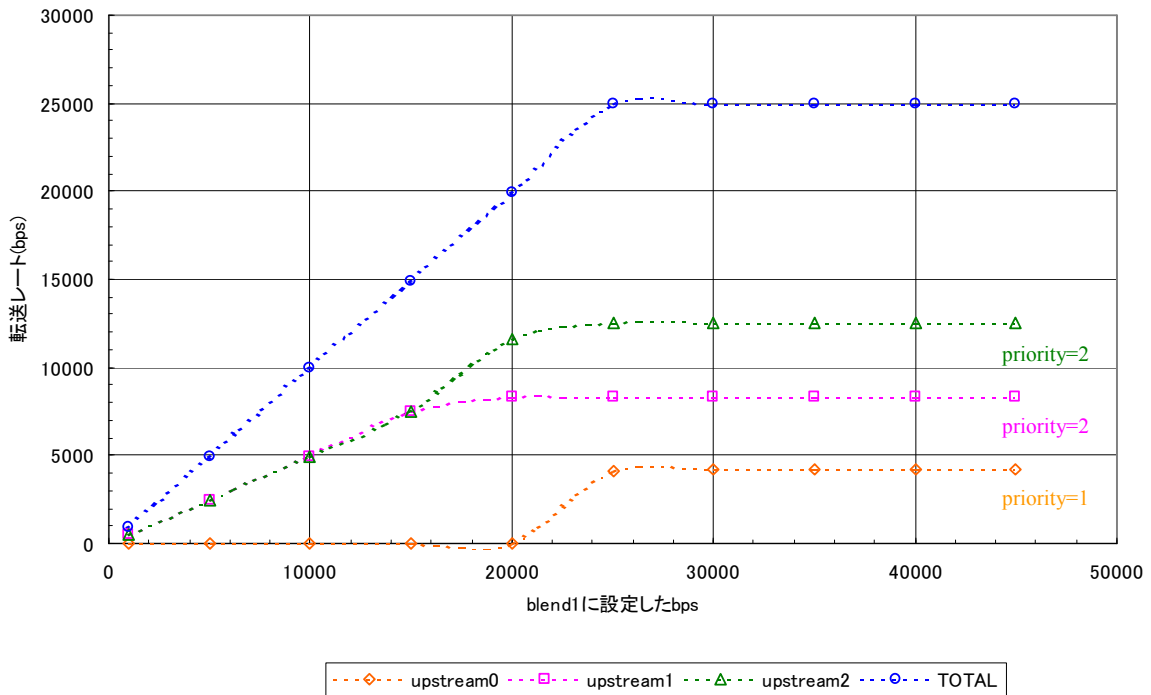


図 14 測定 2-3 の測定結果

これらの結果より、各リンクの `priority` に同じ値が設定している場合、受信するパケットのレートに関係なく、`priority` が同じ値のキューからは、均等にパケットを転送していることが分かる。そして、転送レートの合計値は `bps` に設定されたレートとほぼ同じ値を取っていることが分かる。また、測定 1 の結果と同様に、`priority` の高いキューに格納されたパケットから順に転送していることが分かる。

2.2 WRR 評価

(ア) 測定条件

今回の測定では、図 15 のように、リンク `upstream0`、`upstream1`、`upstream2` を生成し、これらのリンクからパケットを受信する。その際、`ng_blend` のノード `blend1` のノード・パラメータである `bps`、`upstream0`、`upstream1`、`upstream2` のリンク・パラメータである `weight` に任意の値を設定し、パケットが、出力側のリンクである `downstream` から、どのように転送されていくのかを纏る。

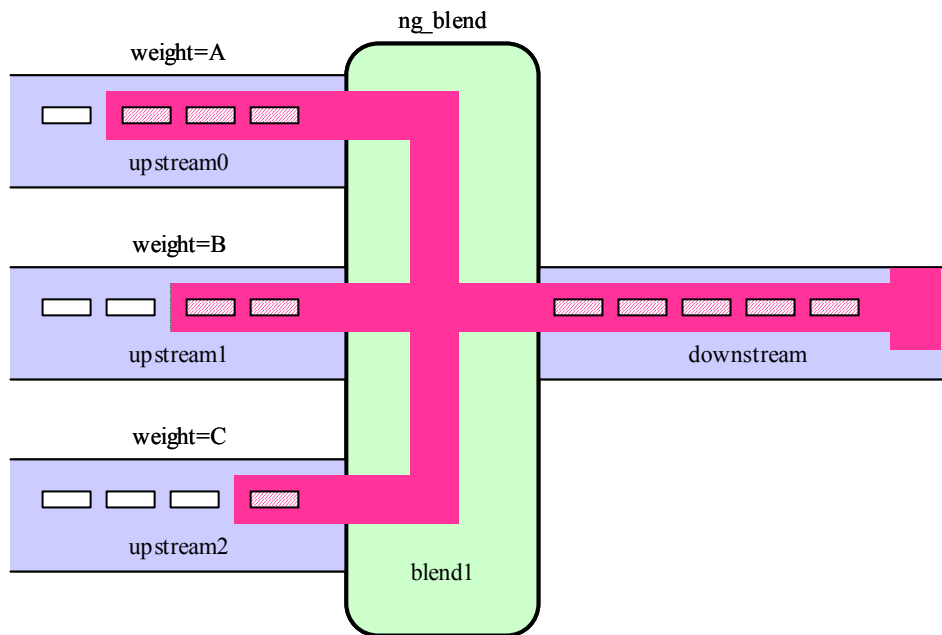


図 15 ng_blend のノードの構成

今回の測定では、上記の構成で、以下に示す 2 通りの方法で測定を行っている。

・測定 1

測定 1 では、各リンクの weight に設定する値の誤差が小さい場合の測定を行う。各リンク・パラメータの設定値は、表 9 の通りである。

表 9 測定 1 の各リンク・パラメータの設定値

#	パラメータ名	設定値		
		upstream0	upstream1	upstream2
1	weight	1	2	3
2	size	100	100	100

また、測定 1 で、各受信側のリンクが受信するパケットの受信レートは、表 10 の通りで、受信するパケットのサイズは、全て 260 バイトである。

表 10 測定 1 のパケットの受信レート

パケットの受信レート(bps)		
upstream0	upstream1	upstream2
約 50M	約 100M	約 150M

・測定 2

測定 2 では、測定 1 とは逆に、各リンクの weight に設定する値の誤差が大きい場合の測

定を行う。各リンク・パラメータの設定値は、表 11 の通りである。

表 11 測定 2 の各リンク・パラメータの設定値

#	パラメータ名	設定値		
		upstream0	upstream1	upstream2
1	weight	1	100	200
2	size	100	100	100

また、測定 2 で、各受信側のリンクが受信するパケットの受信レートは、表 12 の通りで、受信するパケットのサイズは、測定 1 同様、全て 260 バイトである。ただし、weight の比率にパケットの受信レートの比率を合わせているため、測定 1 と測定 2 の受信レートの値は、異なる値を取る。しかし、測定 1、測定 2 共に、受信レートの合計値は、およそ 300 Mbps である。

表 12 測定 2 のパケットの受信レート

パケットの受信レート(bps)		
upstream0	upstream1	upstream2
約 1M	約 100M	約 200M

(イ) 測定結果

・測定 1

各リンク・パラメータの weight の値の差が、小さい場合の測定結果を図 16、図 17 に示している。これらの測定結果の横軸は、blend1 に設定した bps の値で、縦軸は、blend1 の downstream から転送されたパケットのレートを bps オーダーで表している。また、各リンクに設定されているパラメータの値は、表 9 の通りである。

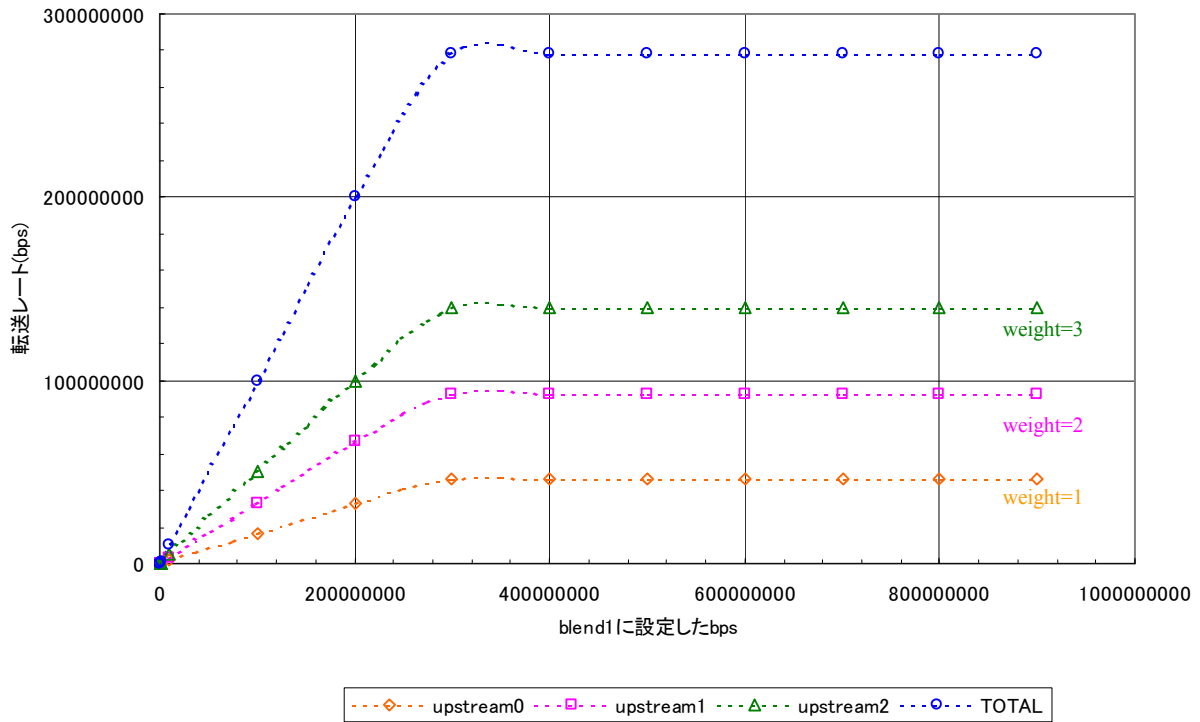


図 16 測定 1 の測定結果 (線形-線形)

また、図 16 の横軸と縦軸を対数にしたグラフを図 17 に示す。

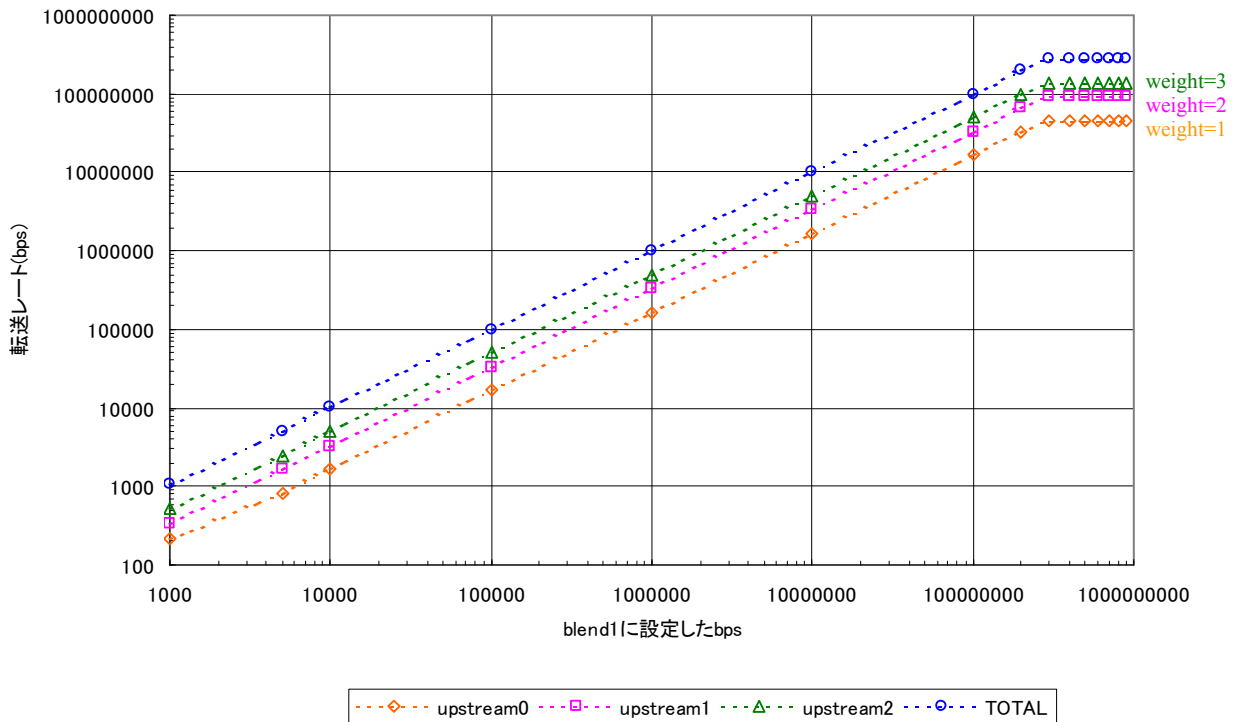


図 17 測定 1 の測定結果 (対数-対数)

図 16 より、転送レートの合計値は、bps に設定されたレートとほぼ同じ値を取っていることが分かる。また、各キューから転送しているパケットの転送レートは、weight の値より、

upstream1 は upstream0 の約 2 倍、upstream2 は upstream0 の約 3 倍となっていることが分かる。また、図 17 より、bps の値が 200000000 となるまで、グラフが平行で、ほぼ直線的に比例していることから、L2Shaper は、bps に小さな値を設定しても weight と bps に設定された値に従い、パケットを転送していることが分かる。

・測定 2

各リンク・パラメータの weight の値の差が、大きい場合の測定結果を図 18、図 19 に示している。

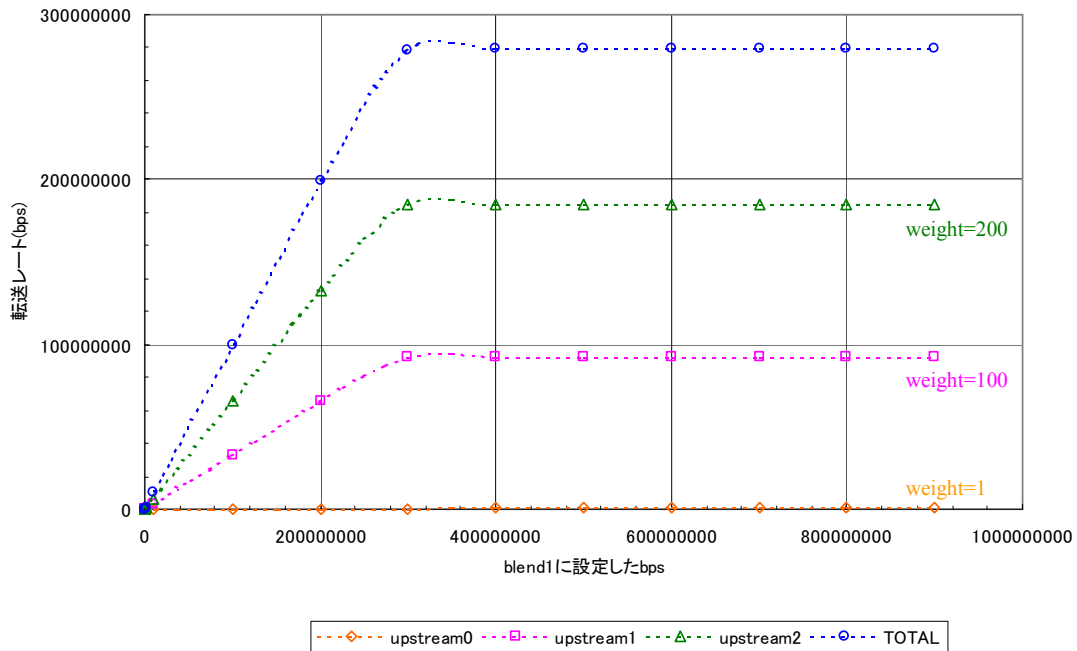


図 18 測定 2 の測定結果 (線形-線形)

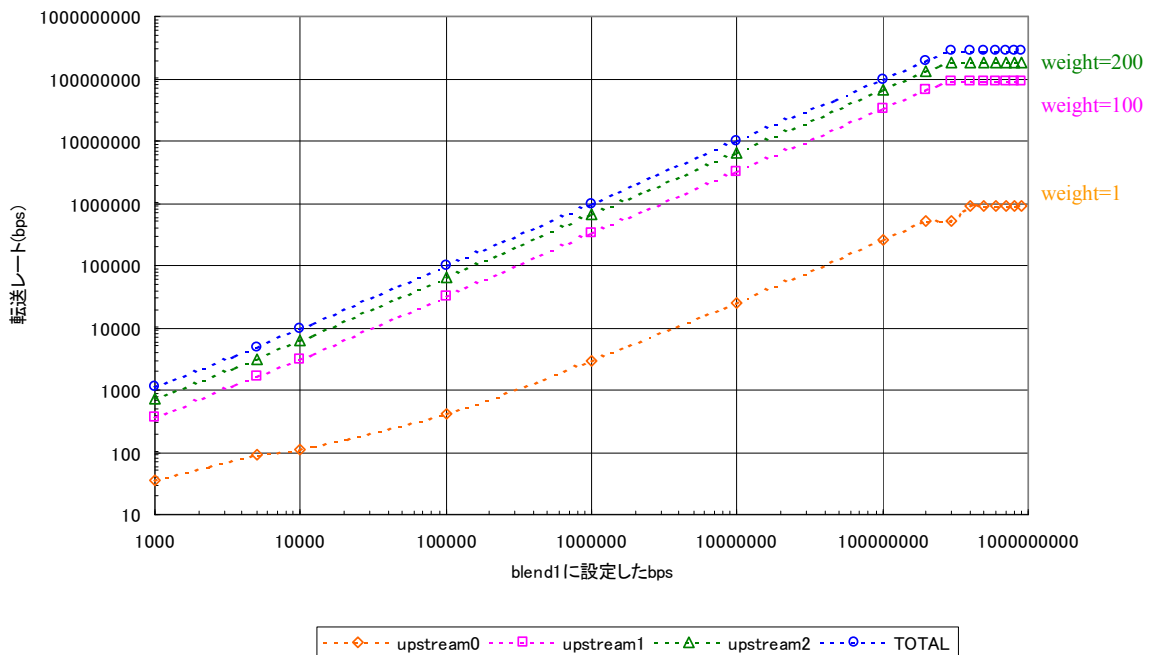


図 19 測定 2 の測定結果 (対数-対数)

これらの測定結果の横軸は、blend1 に設定した bps の値で、縦軸は、blend1 の downstream から転送されたパケットのレートを bps オーダーで表している。また、各リンクに設定されている weight の値は、表 11 の通りである。

また、図 18 の横軸と縦軸を対数にしたグラフを図 19 に示す。

図 18 より、転送レートの合計値は、bps に設定されたレートとほぼ同じ値を取っていることが分かる。また、upstream0 は、転送レートの値が小さすぎて分からないが、upstream1 と upstream2 のレートを比較すると、重みの割合通り、2 倍程度になっていることが分かる。また、図 19 より、bps の値が、10000 となるまでは、upstream0 の転送レートが、weight の割合に比べ大きい値を取ることが分かる。しかし、bps が 100000 を越す辺りから、upstream0 のレートの精度が良くなり、設定した weight の値の割合通り、upstream1 の約 100 分の 1 で、パケットを転送していることが分かる。

2.3 結果のまとめ

PQ では、各リンクに設定された priority の値が高いリンクから、bps の値を上回らない転送レートで、キューに格納されたパケットを転送する。測定結果より、受信するパケットのレートに関係なく、bps に設定された転送レートに従い、priority が高いキューに格納されているパケットから順に転送を行う。また、priority の値が同じキューに格納されているパケットについては、均等に転送を行っていることが見て取れる。

また、WRR では、各リンクに設定されたリンク・パラメータ weight の値から、各キューの転送レートを算出し、この値を上回らないようにパケットを転送する。測定結果より、weight の差が小さければ、ノード・パラメータの bps の値に関係なく weight に比例した転送レートで、各キューからパケットを転送する。しかし、各リンクに設定された weight の差が大きい場合、bps の値に大きな値を設定すると weight に比例した値でパケットを転送するが、bps の値が小さければ、weight の小さなキューの転送レートの精度が落ちる。

いずれの場合でも、フローの最大帯域は 100 Mbps を大きく超えており、当初の目標に到達している。

3 フロー最大数の検証

今回の L2Shaper プロトタイプでは、扱えるフローの最大数はコンパイル時の定数により定められている。1 フロー辺りのデータサイズは約 300 Byte であるため、1000 フローの設定を行う場合の必要メモリサイズは約 300 KByte となり、サーバクラスの PC であれば十分実用可能なメモリサイズに収まっている。

実験ではメモリサイズ 1 GByte の PC を用い、1000 フローの設定が行えることを確認した。

第 2 部 アプリケーション対応通信品質制御技術

平成 14 年度に方式検討を行ったアプリケーション対応通信品質制御技術の更なる検討を行い、検証用実験コンテンツを開発して、アプリケーション対応通信品質制御技術の効果を検証した。

検証用実験コンテンツとしては、大容量の動画データの転送を行い、リアルタイムで送信レートの変更を行うアプリケーションが適切である。今回は上記条件を満たすアプリケーションの一例としてビデオアダプタを採用した。また、動画のストリーミング配信といっ

た、信頼性のないトランスポートで輻輳制御を行うプロトコルとしてDCCPを採用し、DCCP上で動作するビデオアバタで通信品質制御の検証を行った。

1 検証用コンテンツの概要

今回の実験において対象としたアプリケーションは、一般にビデオアバタと分類されるものである[Hirose 2001]。ビデオアバタは、3次元CGにより合成されたVR(Virtual Reality)空間を背景として、その中に撮影されたリアルタイム映像を表示キャラクタ(アバタ)として合成し、相手側に表示することで、相互にコミュニケーションするアプリケーションである。今回はアバタ自身も上半身を撮影画像、下半身をポリゴンモデルを用いたCG画像とのハイブリッドとして作成した。

ハイブリッドとすることで、

- ・自由にVR空間内の移動動作を行える
- ・動画領域を節約することが出来る
- ・撮影設備の安価化、小型化が図れる

といったメリットが生まれる。

ビデオアバタ(VA)においては、既存のTV会議などと同様に、インタラク션을維持するための高いリアルタイム性と、十分な帯域幅を安定して確保する必要がある。

今回の検証用コンテンツでは、撮影された上半身の映像と、VR空間内での視点・移動量といった駆動情報を互いに送りあい、相手側で合成される。このため、表13のような種類の情報が必要となる。

表 13 VAデータの性質と優先度

VAデータの種類	データ量	データロス	優先度
モデル部分の駆動制御	小さい	NG	高
動画映像	大きい(可変)	OK	低

これらの情報にはそれぞれ性質が異なる要素が混在する。CGモデルを駆動制御する情報については、VR空間内の視点情報が欠落して間欠的に送られると、それに伴いビデオアバタ(VA)の移動が不連続で不自然な動きとなるため、情報の量は少ないものの、データのロスが無く、安定的にデータを送ることが求められる。

一方、データ容量上で大きな比率を占める動画像については、データが一部欠損した場合にも一時的な画像の乱れのみで済み、必ずしもデータの信頼性が必要とされない。

また、我々のビデオアバタ(VA)では、動画像は常に高解像度のものが必要とされるとは限らないという特徴がある。特にアバタがVR空間内を移動する場合には、視点位置からアバタが離れることによって小さい解像度しか必要とされない場合や、映像のフレームレートが解像度よりも重要になる場合が存在する。

我々はこの理由から、ビデオアバタにおいては、アバタ間の距離やネットワーク状況に応じて、アバタの動画部分の転送画像サイズやフレームレートを動的に調整することで、ネットワークの状況に応じた最良の品質の動画転送が行えるものと考え、ベストエフォート型のネットワーク上でのホスト間での輻輳制御と、アプリケーションでの動的な転送レ

ート調整の協調により最適な動画転送を実現する事を試みた。

2 ストリーム輻輳制御

現在、インターネットで広く使われているトランスポート層プロトコルは TCP と UDP である。動画映像にはデータの欠損が許容され、また、アプリケーションとしては古いデータを再送されるより最新のデータを送ってもらうことが望ましい。そのため、動画転送には UDP が広く使われている。しかし、UDP は輻輳制御が無いため、転送レートは送信側が一方的に決める必要があり、また、送信側にはどの程度のデータが届いているのか確認できない。従って、イントラネットのように動画転送経路の状況を全て把握できる場合での使用には問題ないものの、途中の経路の品質が全く保証されていないインターネットで動画転送に UDP を利用することは問題が多い。

トランスポート層に UDP を使用し、アプリケーションで独自に輻輳制御を行う方式として TFRC(RFC3488) [Floyd 2000] が提案され、輻輳制御方式としては有用であることが報告されている。しかし、TFRC はプロトコルの提案ではなく、制御アルゴリズムの提案であるため、最初から TFRC の利用を前提にアプリケーションの開発を行った場合はともかく、既存のアプリケーションが個別にこの方式を組み込むのは工数もかかり、現実的とはいえない。そのため、輻輳制御を行う信頼性のないトランスポート層プロトコルとして DCCP(Datagram Congestion Control Protocol) が提案され、現在 IETF の WG [DCCP WG] で審議中である。

今後、ネットワークが高速化するに従って、インターネット上でビデオアバタのようなインタラクティブな動画転送を行うアプリケーションの需要は大幅に増加することは確実である。そこで、我々は今回 DCCP を採用し、輻輳制御をプロトコルスタック側で行い、ネットワーク品質に応じた動画転送を、最小限のアプリケーション変更で実現することを目指した。

DCCP は現在 IETF の WG で審議中のプロトコルであり、プロトコルの大枠は固まったものの、未だ互換性のない仕様の変更が続けられている実験段階のプロトコルである。DCCP の実装としては [AEKL 2002][McManus 2002] が知られているが、これらはいずれも Linux 上での実装である。我々のビデオアバタアプリケーションは Windows 上で動作するアプリケーションであり、検討開始時点で Windows 上の実装は見当たらなかったため、我々で Windows 上に DCCP の実装を行い、アプリケーションと組み合わせて実験を行った。

3 DCCP の実装

DCCP の仕様はまだ完全に固まってはいないが、本実験では、DCCP ドラフト第 05 版 [DCCPSPEC] に基づいた実装を行った。また、輻輳制御方式としては CCID3 [CCID3] のサブセットである CCID3-Thin [CCID3THIN] を採用している。

DCCP の実装にはカーネルのプロトコルドライバとして実装する方法と、ユーザ空間上にライブラリとして実装する方法の二通りが考えられるが、今回の目的は DCCP そのものの評価 (TCP との親和性の評価) ではなく、アプリケーションと組み合わせての動作の評価である為、開発の容易さを優先して、図 20 に示すようにユーザ空間上の共有ライブラリとして DCCP の実装を行った。

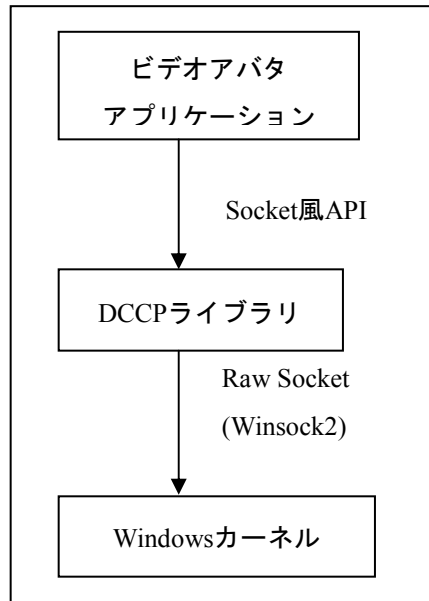


図 20 DCCP ソフトウェア構成図

DCCP は新たなトランスポート層のプロトコルの実装であるため、カーネルとの通信では WinSock2 で提供される RawSocket I/O を使い、IP パケットレベルでのデータの送受信を行っている。また、アプリケーションとのインタフェースであるが、DCCP はプロトコルの規定であって、API を規定しているものではない。とはいえ、コネクション確立やデータの送受信といった、TCP で馴染みの機能に関しては API も同様にしたほうがアプリケーション開発者の負担が小さいと考え、我々の実装でも DCCP ライブラリとアプリケーション間のインタフェースは、Socket インタフェースと同等のインタフェースとした。

以下に開発した DCCP ライブラリが備える外部インタフェース関数を示す。

表 14 DCCP ライブラリ関数の一覧

#	関数名	処理概要
1	dccp_init	DCCP ライブラリの初期化処理を行う。
2	dccp_socket	DCCP ソケットの生成を行う。
3	dccp_connect	サーバ側の DCCP ソケットへの接続を要求する。
4	dccp_bind	DCCP ソケットにアドレスを割り当てる。
5	dccp_listen	クライアントからの DCCP ソケットへの接続を待つ。
6	dccp_accept	クライアントに対し DCCP ソケットの接続受付を行う。
7	dccp_send	DCCP データ・パケットを送信する。
8	dccp_recv	DCCP データ・パケットを受信する。
9	dccp_close	DCCP ソケットをクローズする。
10	dccp_getsockopt	DCCP ソケットに関するオプションの値を取得する。

4 アプリケーションと DCCP の協調

DCCP とアプリケーションの間で情報のやり取りを行い、動的に送信レートを制御するためのインタフェースに関しては、現在も DCCP 研究者の間で様々な意見が出されているが、未だコンセンサスが取れる段階に達していない。アプリケーションインタフェースとして Socket モデルではなく、全く別のモデルを用いる提案 [SharedAPI] もなされているが、上記は提案段階で実績はまだなく、また従来の Socket モデルを前提に開発されたアプリケーションにとっては改造量が非常に多くなるという問題点があった。

我々はアプリケーションの改造量と得られる効果のバランスを考え、今回の DCCP とアプリケーションの連携では、アプリケーションインタフェースモデルとしては Socket モデルを採用し、送信レート制御を行うためにやり取りする情報は個々のパケットに関する情報ではなく、大域的な情報を用いることとした。具体的には、DCCP ライブラリが持つ内部状態である想定送信レート/キューサイズをアプリケーションから参照可能とした。

想定送信レートとは、DCCP ライブラリ側で受信側から送られた ACK 情報を基に、TFRC でのアルゴリズムに基づいて算出した、現在のネットワーク状況で輻輳を起ささないであろうと想定される最大送信レートである。

また、キューサイズは、DCCP ライブラリが内部的に管理しているパケットキューの長さである。

これらの値は、DCCP ライブラリの `getsockopt` 関数によって獲得できる。以下にこの関数の引数と戻り値を示す。

表 15 dccp_getsockopt 関数の引数と戻り値

#	引数 / 戻り値	型	説明
1	第 1 引数	DCCP_SOCKET	接続済みの DCCP ソケット。
2	第 2 引数	int	DCCP ソケット・オプション名 (表 16 参照) をセットする。
3	第 3 引数	void *	オプション名に依存した、取得する値のためのバッファのアドレスを渡す。
4	第 4 引数	int *	第 3 引数のバッファのサイズを指定した変数のアドレスを渡す。正常終了した場合、第 3 引数に渡された値のサイズが渡される。
5	戻り値	int	正常終了すると 0 を返す。エラーの場合、負の値を返す。

DCCP ソケット・オプションは、表 16 のものを用意する。

表 16 DCCP ソケット・オプションの一覧

#	ソケット・オプション	内容
1	SOCKOPT_PACKET_RATE	センダにおいて、レシーバの受信レート、センダの送信レートの実値と推定値、送信キューに格納できず破棄されたデータのドロップ・レ

		ートを取得する。このオプションの値を取得するには sockopt_packet_rate_t 型の構造体を用いる。
2	SOCKOPT_QUEUE	送信キュー、受信キューに格納されたパケット数と、格納できるパケットの最大個数を取得する。このオプションの値を取得するには、sockopt_queue_packet_t 型の構造体を用いる。

今回の検証用コンテンツであるビデオアバタでは、動画送信レートは動画サイズとフレームレートという二つのパラメータにより決定される。動画サイズは 640×480、320×240、160×120 の 3 種類であり、サイズを一段小さくすることで送信レートが 1/4 となる、大きなステップでのレート変更となる。一方、フレームレートは 1 fps(frame per sec)~15 fps の間で連続的に変更可能な値である。

送信レートの制御はフレームレートの制御だけで行うほうが方式として簡単であるが、アプリケーションの性質上、フレームレートはなるべく高い方がユーザに与える印象が高く、また、アバタが VR 空間内で遠方にある場合は動画サイズが小さくても品質に問題がないという特徴を考慮して、以下に示すようにフレームレートと動画サイズの変更を組み合わせさせて送信レートの制御を行った。

■DCCP のバッファ残量が少なくなった場合（送りすぎの状態）

カメラの上限フレームレートを減らす。

この値がある一定以下になった場合には、動画サイズを縮小する

■DCCP のバッファ内容が少なくなった場合（余裕のある状態）

カメラの上限フレームレートを増やす

この値がある程度以上になった場合には、動画サイズを拡大する。

制御用データとしては、想定送信レートを用いる場合と、DCCP バッファサイズを用いる場合とを試作したが、DCCP バッファサイズを用いた場合の方が見た目がスムーズな制御を行えたため、DCCP バッファサイズを用いた制御を採用した。

5 実証実験

実験環境のシステム構成を、図 21 に示す。

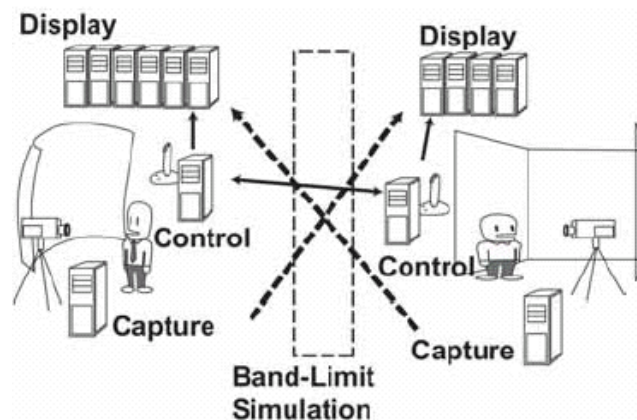


図 21 実験システム構成図

ビデオアバタの1つサイト内には撮影部、表示部、操作部の3台のマシンが置かれ、2つのサイトが互いに撮影した映像を相手側のディスプレイに表示するという構成となっている。2つのサイト間はLAN(100Base-T)で結ばれ、その間に帯域制御機能をもつブリッジであるL2Shaperを設置し、両サイト間の通信速度を動的に変更可能とした。

実験環境では、L2Shaperにより、ビデオアバタの駆動制御データトラフィックに関しては常に1Mbpsの帯域を確保し、映像トラフィックに関しては、帯域幅を20Mbps、10Mbps、5Mbpsの三通りに変化させて、アプリケーションの動作を確認した。

実験結果を以下に示す。

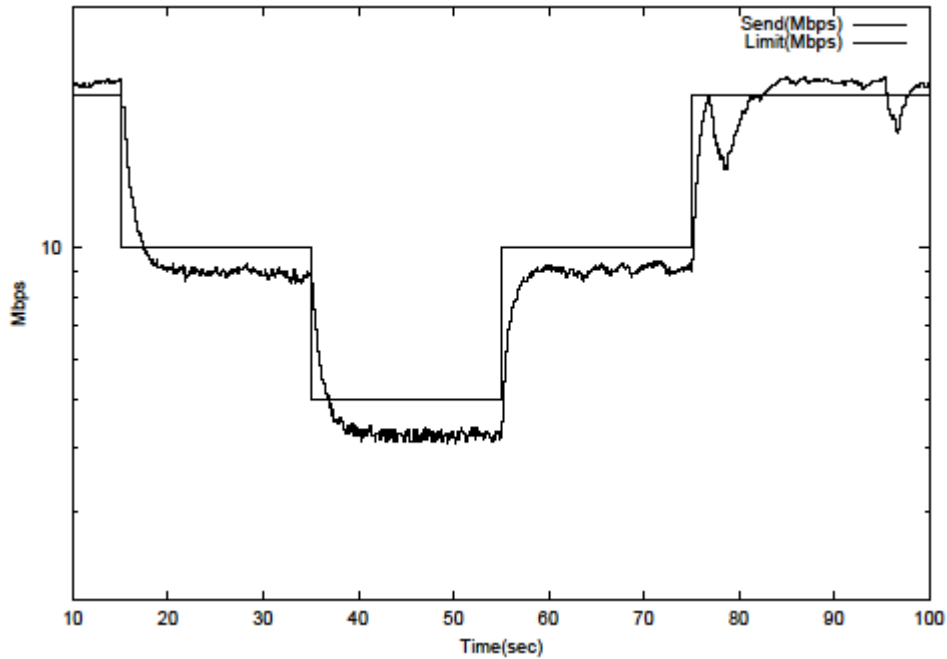


図 22 DCCP ライブラリの送信レート

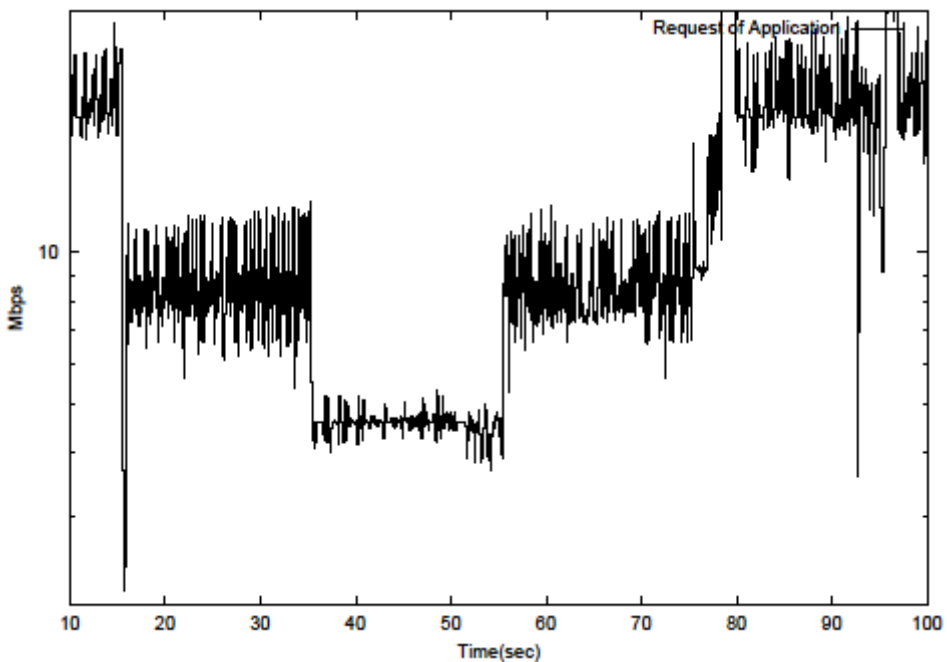


図 23 アプリケーションの送信リクエストレート

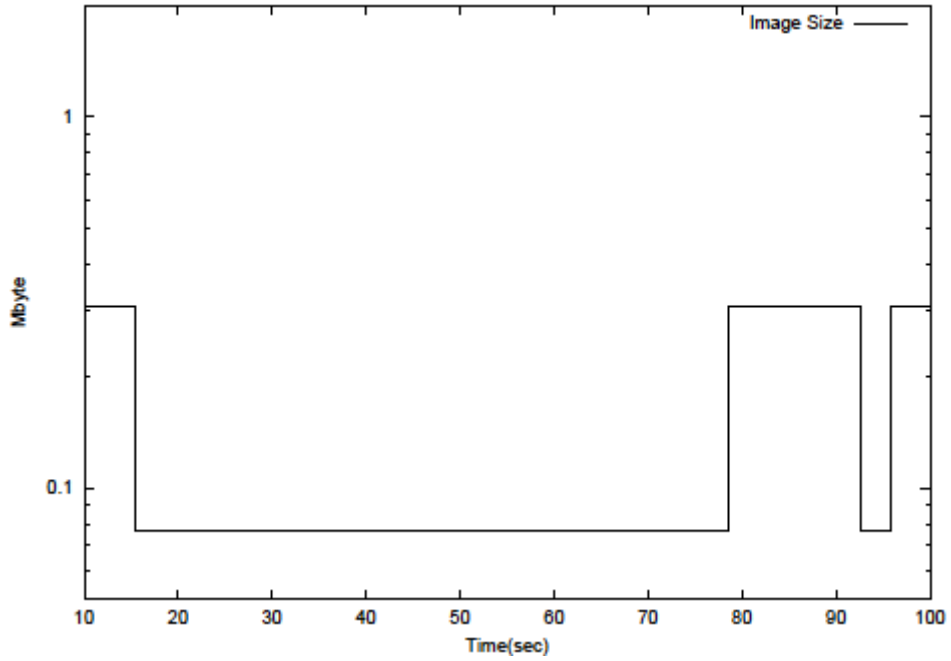


図 24 画像サイズ

帯域幅の変動に合わせて、送信レートが動的に対応、それに合わせて画像サイズとフレームレートの切り替えが行われていることが確認できる。

グラフからも分かるように、ネットワークの帯域幅が変化した瞬間に一瞬転送レートが大幅に変動する点はあるが、表示はほとんど乱れずにスムーズに切り替わり、ユーザに違和感を与えずに切り替えを行うことができた。

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

1 管理対象ネットワーク

1.1 シングルドメイン VLAN

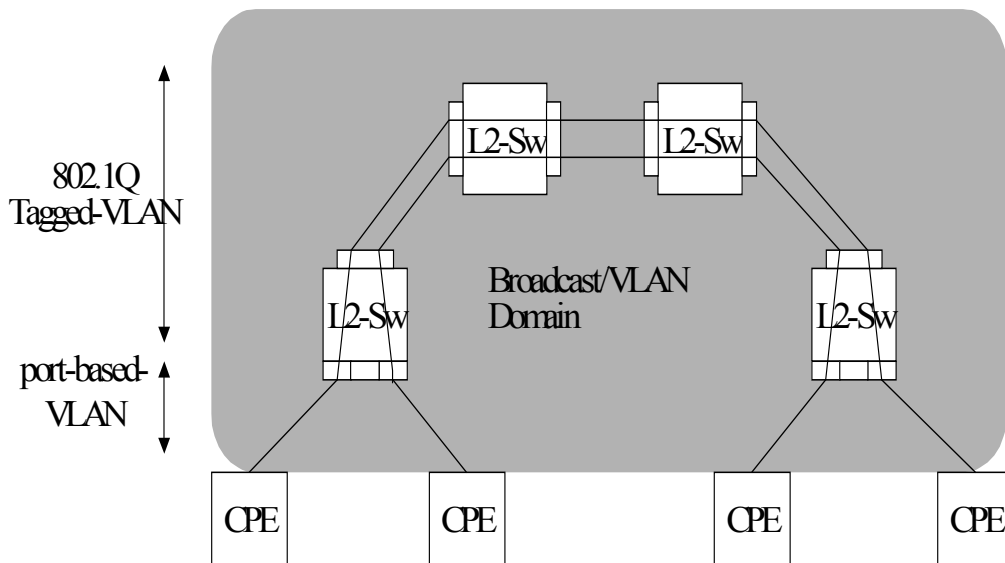


図 1 管理対象ネットワーク図

- ・ブロードキャストドメイン全体で、VLAN_ID は一意である必要がある為、収容できる VLAN は最大で 4094。
- ・原理的には、ブロードキャストドメイン中の全てのノードは、全ての CPE の MAC アドレスを FDB に保持する必要がある。
- ・GVRP を使えば緩和可能だが、その効果は網トポロジに依存。
- ・VLAN 構成の管理は容易。

2 イーサネットプロビジョニングサーバ

2.1 ソフトウェア構成

本研究開発にて開発したイーサネットワーク管理装置のソフトウェア構成図を図 2 に、構成機能一覧を表 1 に示す。

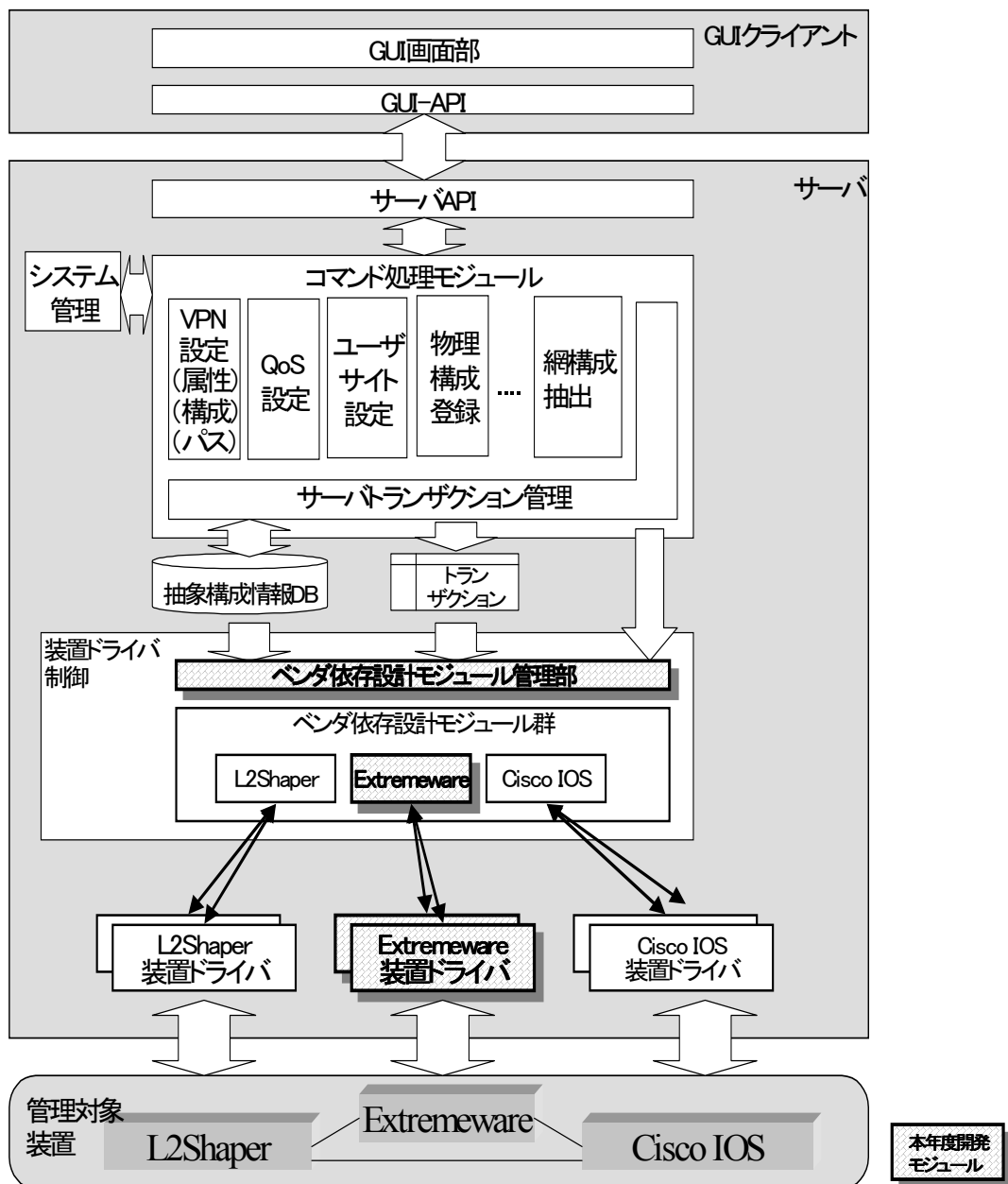


図 2 ソフトウェア構成図

表1 ソフトウェア構成機能一覧

#	機能名称	機能概要
1	サーバ API	GUI 向けのアプリケーションインタフェースを提供する。
2	コマンド処理 モジュール	XML 形式データ(コマンド+パラメータ)を解釈し、各コマンドを実行する。
3	サーバトランザクション 管理モジュール	サーバ内のトランザクションの管理を行う。
4	抽象構成情報 DB	設定情報を保持しておく DB を構築する。
5	ベンダ依存設計 モジュール管理部	サーバトランザクションから、制御対象となるルータ・スイッチを決定し、適切な機種に対応したベンダ依存装置ドライバ制御モジュールに、ルータトランザクションへの展開を指示する。
6-1	IOS 装置ドライバ制御 モジュール	コマンド処理モジュールからの指示に従い、設定情報 DB の変更前と変更後の差分を抽出し、IOS の装置コマンド列に変換する。
6-2	Extremeware 装置 ドライバ 制御モジュール	コマンド処理モジュールからの指示に従い、設定情報 DB の変更前と変更後の差分を抽出し、Extremeware の装置コマンド列に変換する。
7-1	IOS 装置ドライバ モジュール	IOS 装置ドライバ制御モジュールからの指示に従い、IOS 搭載装置に対してログインし、コマンドの投入を行う。
7-2	Extremeware 装置 ドライバモジュール	Extremeware 装置ドライバ制御モジュールからの指示に従い、Extremeware 搭載装置に対してログインし、コマンドの投入を行う。
8	システム管理 モジュール	サーバ初期設定や GUI のログイン権限を管理する。

2.2 ベンダ依存設計モジュール管理部

ベンダ依存設計モジュール管理部は、サーバトランザクションを元に、制御対象となるルータ・スイッチを決定し、適切な機種に対応したベンダ依存装置ドライバ制御モジュールに、ルータトランザクションへの展開を指示する。更新対象ルータの判定手順を図 3 に示す。

```

for (指定されたトランザクション中の更新中レコード){
    if(レコードが Sap テーブルに属す){
        Sap に関連付けられたルータを更新対象ルータリストに追加; continue;
    }
    if(レコードが router テーブルに属す){
        その router を更新対象ルータリストに追加; continue;
    }
    if(レコードが line テーブルに属す){
        その line を収容している router を更新対象ルータリストに追加; continue;
    }
    :

```

図3 更新対象ルータ判定手順

2.3 Extremeware 装置ドライバ制御モジュール及び装置ドライバ

2.3.1 機能概要

Extremeware 装置ドライバ制御モジュールで実現する機能を以下に示す。

- ・ 初期化時にそれ以前に作成されていた DB を基に仮想ルータを生成する機能
- ・ checkSetting 受信時に対象装置を決定し、該当装置の仮想ルータを変更する機能
- ・ tryCommit 受信時に対象装置を決定し、仮想ルータの差分を抽出した上で設定コマンドを生成した後、該当装置に対するコマンド送信を装置ドライバに対して要求する機能
- ・ Commit 受信時に対象装置を決定し、保持していたコマンドを削除する機能
- ・ Commit 受信時に仮想ルータの更新を行う機能

上記機能を実現するため、Extremeware 装置ドライバ制御モジュール及び、Extremeware 装置ドライバモジュールの機能を、図 4 に示す構成にて構築した。各モジュールの機能概要を、表 2 及び表 3 に示す。

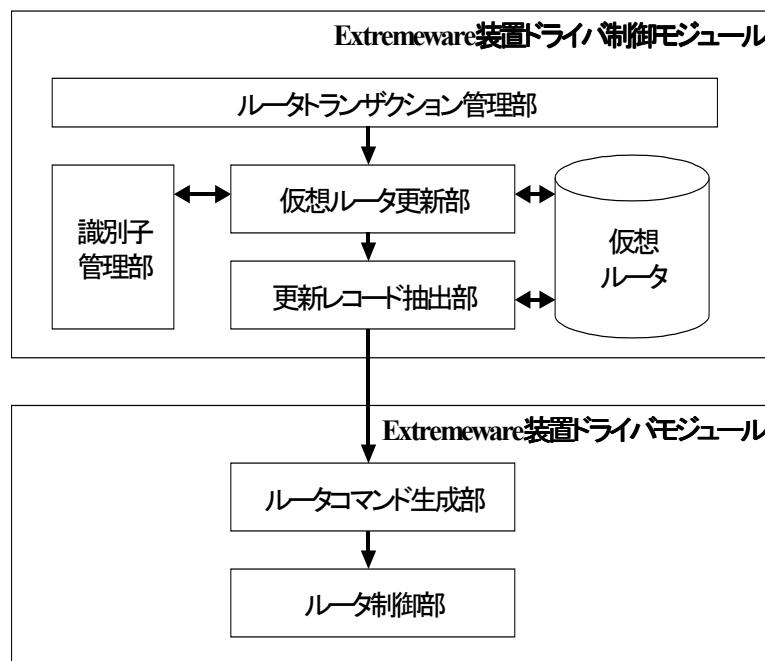


図4 Extremeware 依存機能の機能ブロック概念図

表2 Extremeware 装置ドライバ制御モジュールの構成要素

	サブモジュール名称	機能概要
1	ルータトランザクション管理部	サーバトランザクション(装置非依存の抽象構成情報 DB の更新差分情報)とルータトランザクション(機種依存の具象構成情報である仮想ルータの更新差分情報)との間の入れ子従属関係を管理する。
2	仮想ルータ更新部	サーバトランザクションを元に、仮想ルータの更新すべきレコードの一覧を生成する。

3	更新レコード抽出部	仮想ルータの更新レコードを元に、ルータコマンドの投入制約順に従って、ルータ更新コマンド列及びロールバックコマンド列の元となる、更新レコード列を抽出する。
4	識別子管理部	ルータに設定する際の各種識別子を管理する。

表 3 Extremeware 装置ドライバモジュールの構成要素

	サブモジュール名称	機能概要
1	ルータコマンド生成部	仮想ルータの更新レコード列を元に、ルータ更新コマンド列及びロールバックコマンド列を生成する。
2	ルータ制御部	ルータにリモートログインし、ルータトランザクションに従って構成定義コマンドを実行する。コマンド実行にエラーが発生したか、ルータトランザクション管理部からトランザクションのアボートを要求された際には、ロールバックコマンドを実行する。

2.3.2 仮想ルータモデル

2.3.2.1 仮想ルータのクラス依存性

あるクラスのオブジェクトが生成／削除／属性変更されると、これを契機に他のクラスのオブジェクト属性値を更新する必要となる事がある。例えば、port オブジェクトは port_policy オブジェクトの識別子を保持する事により、port_policy との関連を保持しているため、port_policy オブジェクトを生成/削除する際には、port オブジェクトの port_policy 属性を変更する必要がある。

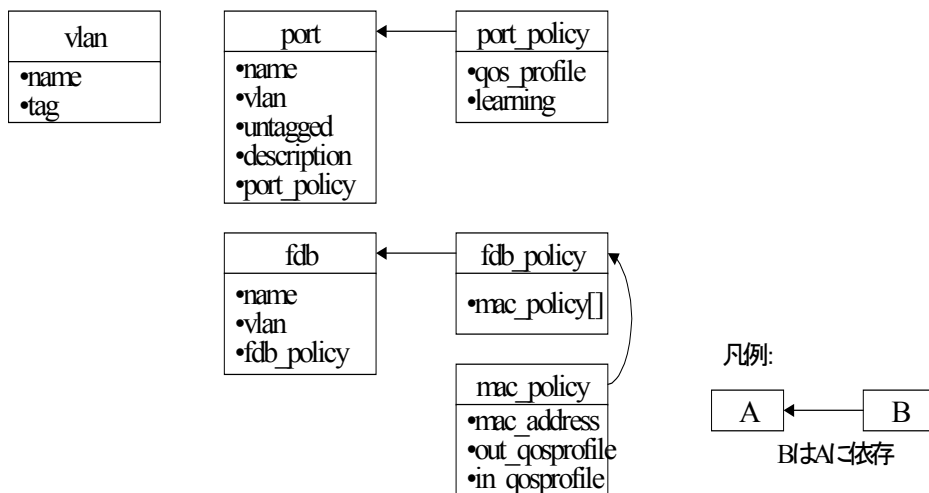


図 5 仮想ルータ依存性グラフ

2.3.2.2 仮想ルータのクラス構成

仮想ルータは、ツリー状に管理される。抽象網構成 DB には、実ルータと一対一対応した複数のルータオブジェクトが配列状に保持され、その各々には表 4 に示す形式にて vlans, ports, fdbentry の各ツリーが保持される。これらは表 5 に示すように実ルータ内のコンフィギュレーションオブジェクトと、一対一に対応しており、容易にルータコマンドを生成する事が出来る。

表 4 仮想ルータモデル

tag	value	tag	value	tag	value	tag	value	tag	value	tag	value	ツリー再構築*1 (◎/○/×)	備考
configuration	""											×	1 element の構成
		vlan	""									×	(vlaninf 単位)
				vlan(複)	""							◎	
						name	<vlanName>					○	
						vlan_object						○	
		ports	""									×	(logicakinf 単位)
				port(複)	""							◎	
						name	<portName>					○	
						port_object						○	
						port_policy	""					○	
						port_policy_object						○	
		fdbentry	""									×	(logicalinf 単位)
				fdb(複)	""							◎	
						name	<portName>					○	
						fdbentry_object						○	
						fdb_policy	""					○	
								mac_policy(複)	""			○	
								mac_address	<macaddress>			○	
						mac_policy_object						○	

*1 ツリー再構築

◎ : 再構築の際に、ツリーの頂点になる、再構築の結果フラグが立つ

○ : 再構築の結果変更フラグが立つ

× : ツリー再構築の対象外であり、変更フラグは立たない

表 5 仮想ルータとデータベースの対応

仮想 ルータ クラス名	属性名	対応 DB テーブル名	対応 DB フィールド 名	対応管理対象 オブジェクト名	備考
vlan					
	<vlanName>	SAP	vpnName		SAPテーブルからVPNテーブルを参照する
	tag <vlanId>	VPN	vpnId		SAPテーブルからVPNテーブルを参照する
port					
	<portName>	Line	lineName	logical(subInf=1)	
	vlan <vlanName>	SAP	vpnName		Line の sapId から SAP テーブルを参照し、SAP テーブルから Line テーブルを参照する。
	untagged	Line	lineType	logical(subInf=1)	
	description <string>	Line	description	logical(subInf=1)	
	qosprofile <qosprofile>	policy(QoS)	policyRule		Line の policyId から policy テーブルを参照する
	learning	policy(QoS)	policyRule		Line の policyId から policy テーブルを参照する
fdbentry					

	port <portName>	Line	lineName	logical(subInf=1)	
	vlan <vlanName>	SAP	vpnName		Line の sapId から SAP テーブルを参照し、SAP テーブルから Line テーブルを参照する。
	mac_address <macAddress>	policy(QoS)	policyRule		Line の policyId から policy テーブルを参照する
	out_qosprofile <outQosprofile>	policy(QoS)	policyRule		Line の policyId から policy テーブルを参照する
	in_qosprofile <inQosprofile>	policy(QoS)	policyRule		Line の policyId から policy テーブルを参照する

2.3.3 生成するコマンド

2.3.3.1 vlan クラス

- 設定

```
create vlan <name>
```

```
config vlan <vlan name> tag <vlan tag>
```

- 変更

```
config vlan <name> name <vlan name>
```

```
config vlan <vlan name> tag <vlan tag>
```

- 削除

```
delete vlan <vlan name>
```

2.3.3.2 port クラス

- 設定

```
config vlan <vlanName> add ports <portNumber> [untagged | tagged]
```

```
config ports <portNumber> vlan <vlanName> [lock-learning | limit-learning <number>]
```

```
config ports <portNumber> display-string <string>
```

```
config ports <portNumber> qosprofile <qosprofile>
```

- 変更

```
config vlan <vlanName> add ports <portNumber> [untagged | tagged]
```

```
config ports <portNumber> vlan <vlanName> [lock-learning | limit-learning <number>]
```

```
config ports <portNumber> display-string <string>
```

```
config ports <portNumber> qosprofile <qosprofile>
```

- 削除

```
config vlan <vlanName> delete ports <portNumber>
config ports <portNumber> vlan <vlanName> [unlock-learning | unlimited-learning]
unconfig ports <portNumber> display-string
config ports <portNumber> qosprofile none
```

2.3.3.3 fdbentry クラス

- 設定

```
create fdbentry <macaddress> vlan <vlanName> ports <portNumber> qosprofile <out_qosprofile> ingress-qosprofile <in_qosprofile>
```

- 変更

```
create fdbentry <macaddress> vlan <vlanName> ports <portNumber> qosprofile <out_qosprofile> ingress-qosprofile <in_qosprofile>
```

- 削除

```
delete fdbentry <macaddress> vlan <vlanName>
```

2.4 評価

2.4.1 性能

図 6 に示す構成にて、GUI クライアントを操作し、GUI クライアントが要求メッセージをイーサネットプロビジョニングサーバに送信後、制御対象イーサネットスイッチ Alpine の設定を終え、GUI クライアントに応答が返されるまでの時間を計測した。

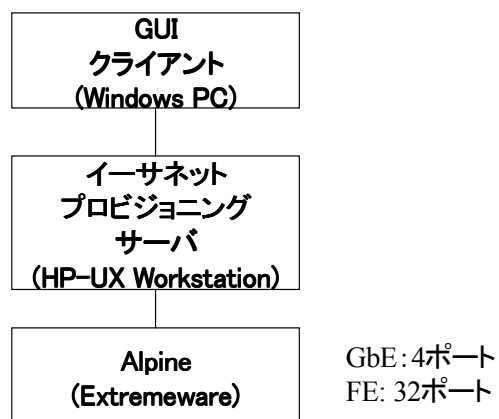


図 6 性能評価試験システム構成

結果を図 7 に示す。殆どが 4 秒前後となっており、運用上十分な処理性能が得られた。

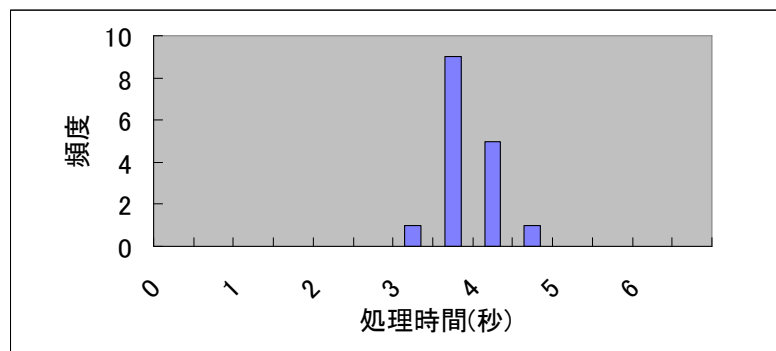


図 7 処理時間

2.4.2 スケーラビリティ

大規模網を管理可能であるか否かを評価するため、図 8 のシステム構成で実験を行った。多数のイーサネットスイッチを実際に用意することは、事実上不可能であるため、装置ドライバ部のルータ制御部において、イーサネットスイッチへのリモートログインを行わないように設定した上で、プロビジョニングサーバ部に対して、多数の装置及び VLAN を設定した。尚、本実験の際には、1 プロセスが使用可能な最大メモリサイズが 1 GB となるように、OS のカーネルパラメータを設定した。

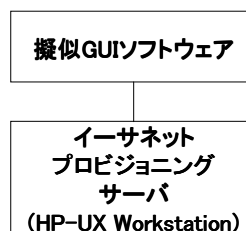


図 8 スケーラビリティ検証実験構成

2.4.2.1 イーサネットスイッチ及び I/F の登録

1 台当り 100 I/F のイーサネットスイッチを 100 台登録し、正常に登録されることを確認した。また、その際のサーバプロセスが占有するメモリ空間のサイズを、ps コマンドを用いて確認した。その結果を図 9 に示す。

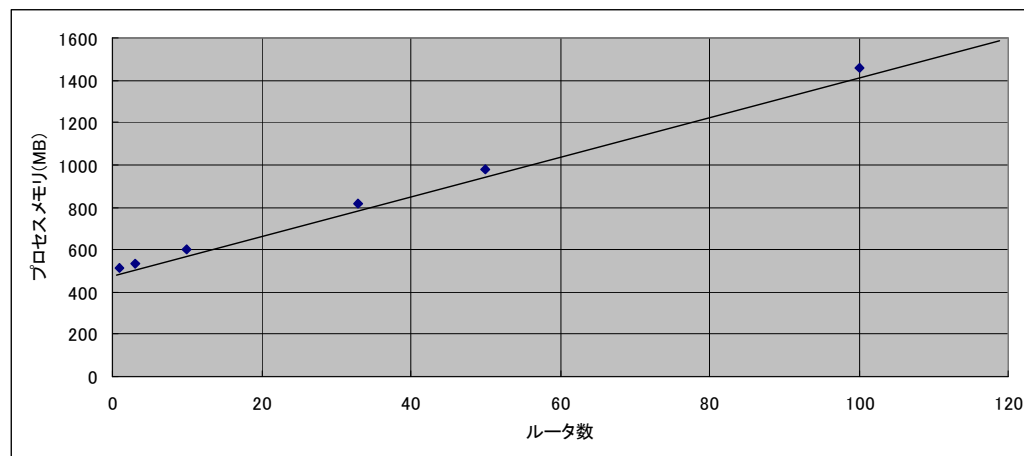


図 9 ルータ数とプロセスメモリの関係

2.4.2.2 VLAN 及びフローの登録

次に、前述の仕様のイーサネットスイッチを登録した後に、多数の VLAN 及びフロー制御パラメータを設定し、その動作を確認した。イーサネットスイッチ数は 50 とした。設定した VLAN の仕様を図 10 に示す。本実験により、目標とした 1000 フローの制御が可能であることを確認した。

インタフェース数：イーサネットスイッチ 1 台当り 100
顧客あたりの VLAN 数：1
VLAN 当りの収容 I/F 数：2
I/F 当りの設定フロー数：4

図 10 設定した VLAN の仕様

また、フロー数の増加に伴って、サーバプロセスが閉めるメモリ空間のサイズを、ps コマンドにより計測したところ、図 11 のようになった。フロー数の増加に従って線形に増加している様子が確認された。

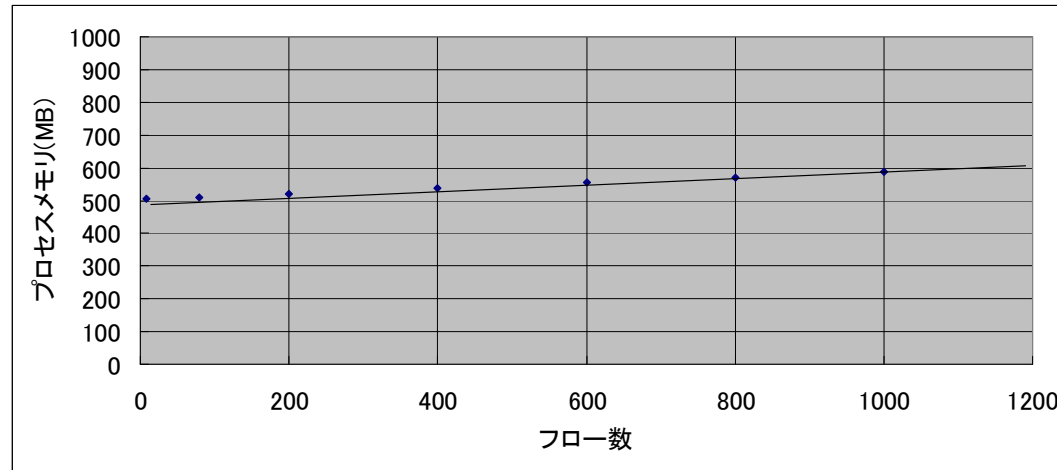


図 11 フロー数とプロセスメモリの関係

3 L2 メータ

3.1 システム概要

システム概要を図 12 に示す。本ソフトウェアは、ギガビットネットワークにおける L2/L3 通信品の計測が、ネットワークプロセッサベースにて可能となることを検証することを目的とする。

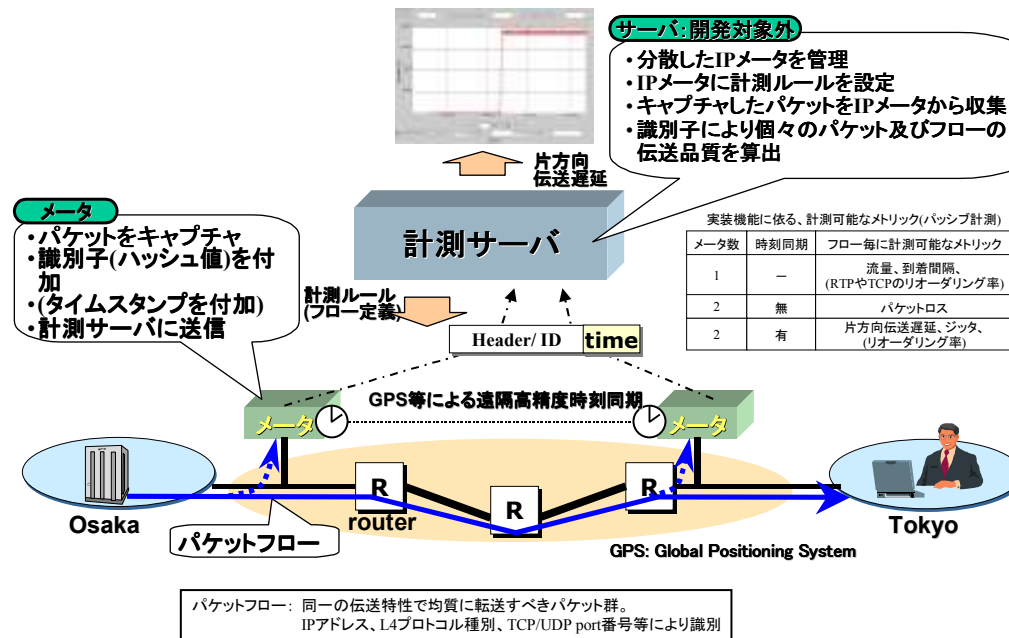


図 12 システム概要

3.2 ソフトウェア構成

本システムのソフトウェア構成を表 6、図 13 に示す。

表6 ソフトウェア構成

#	項目	仕様	備考
1	計測サーバ	<ul style="list-style-type: none"> ・L2 メータに計測対象フロー使用を設定する ・L2 メータが取得した計測対象フローに属するパケット群の特徴情報を受信し、それらの相関を元にフローの伝送品質を算出する 	開発対象外
2	フロー計測制御ソフトウェア	<ul style="list-style-type: none"> ・計測サーバから計測対象フロー仕様を受信しフロー制御ソフトウェアに対して設定 ・フロー計測ソフトウェアから、フローに属する個々のパケットの特徴情報を受信し、集計する 	汎用 CPU 上で動作
3	フロー計測ソフトウェア	<ul style="list-style-type: none"> ・フロー計測制御ソフトウェアから受信した計測対象フロー仕様に基づき、NW-I/F で受信したパケットから当該フローに属するものを抽出し、その特徴情報を抽出し、フロー計測制御ソフトウェアに送信する 	ネットワークプロセッサの μ エンジン上で動作

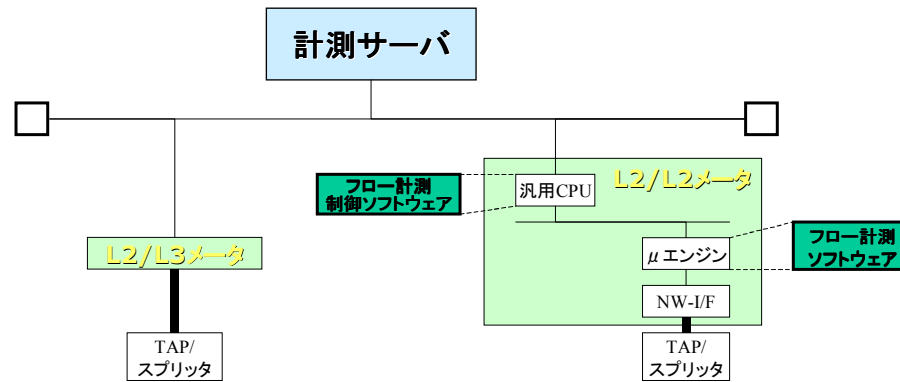


図 13 ソフトウェア構成

3.3 機能構成

本ソフトウェアの機能ブロック構成を図 14 に示す。また各機能ブロックの処理概要を表 7 に示す。

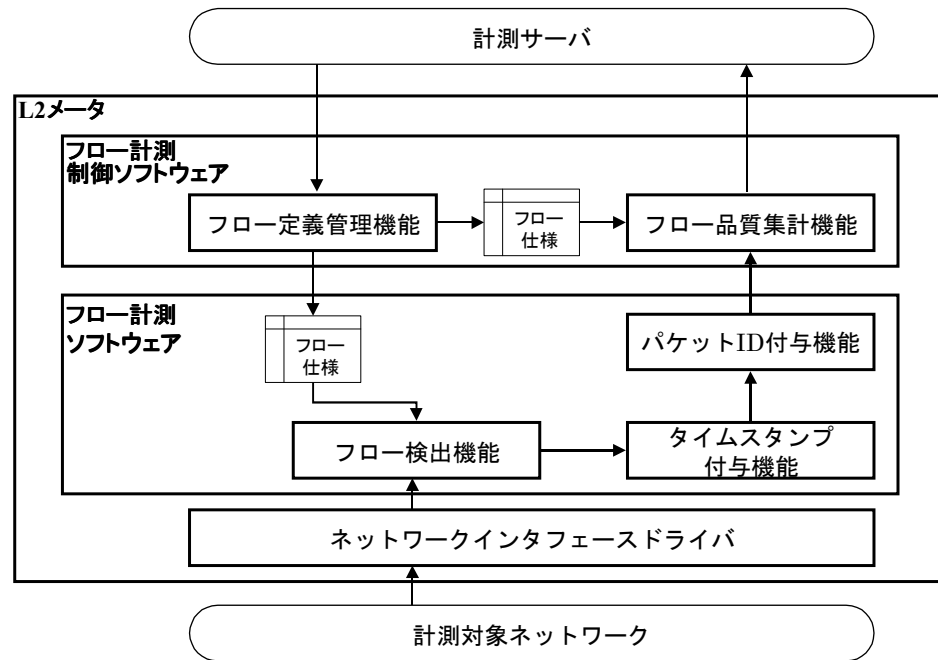


図 14 機能ブロック構成

表 7 機能一覧

#	大項目	中項目	概要
1	フロー計測制御ソフトウェア	フロー定義管理機能	品質管理サーバから、計測対象フロー定義を受取り、フロー計測ソフトウェアに設定する

		フロー品質集計機能	フロー計測ソフトウェアから、キャプチャパケットの各々の特徴情報を受け取って一つのメッセージに纏める
2	フロー計測ソフトウェア	フロー検出機能	フロー計測ソフトウェアから指定されたフローのパケットをキャプチャする
		タイムスタンプ付与機能	キャプチャしたパケットのキャプチャ時刻をパケットに付与する。
		パケット ID 付与機能	キャプチャしたパケットの特徴情報を抽出する。

3.4 評価

図 15 に L2/L3 メータの計測限界性能の試験環境を示す。

本構成にて、パケットあたり 512 Byte での 1 Gbit/sec の計測対象パケットを、損失無く処理することができることを確認した。

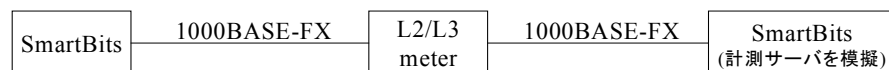


図 15 L2/L3 メータ性能評価実験構成

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

本研究課題については、平成14年度に実施した弊社エッジルータ AG8000 における L2 Shaper 相当機能の基本仕様の検討を元に、AG シリーズに対して IP ネットワークとイーサネットワークを接続し、相互の品質制御情報をマッピングし、品質保証を実施するといった連携機能を実際に適用することを行った。

1 AG の位置づけ

弊社の AG シリーズは、ネットワークプロセッサを採用することにより、柔軟なサービス提供及び高速なパケット処理性能を実現した IP ネットワーク機器である。ブロードバンドアクセスサーバモデルの AG シリーズでは、アクセスサービスにおいて、加入者トラフィックを収容する IPv4/IPv6 デュアルスタックを搭載した機器である。

AG では、PPP あるいは PPPoE を終端することが可能であり、また L2TP のトンネルを使用することも可能である。また、ユーザ管理方法として RADIUS サーバと連携し、各種認証及びアカウント制御を行うことが可能である。

図1に AG が適用されるネットワーク構成例を示す。

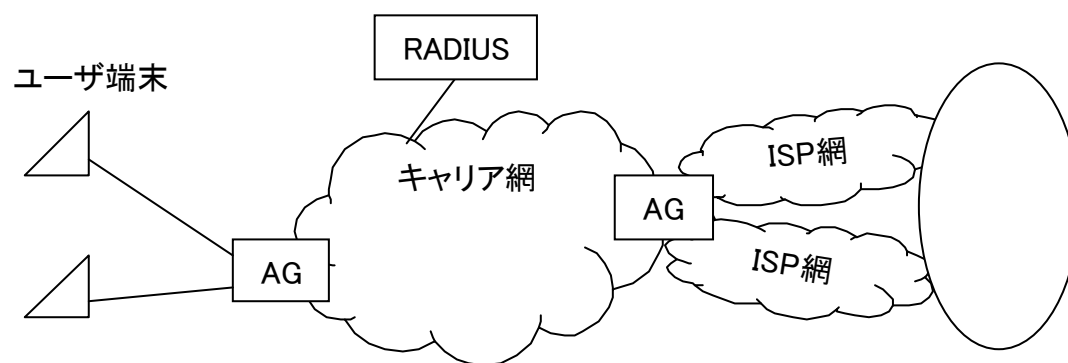


図1 AG シリーズが適用されるネットワーク例

AG シリーズのシステム構成の概略図を図 2 に示す。AG のシステムは、装置制御部、回線インタフェース部、電源部、ファン部から構成されている。また、装置制御部においては、装置制御用の CPU の他に、パケット送受信の専用プロセッサとして、前述の通り、ネットワークプロセッサを搭載している。ネットワークプロセッサ内には、パケット処理専用エンジン（マイクロエンジン：ME）が搭載されており、パケット処理をカスタマイズすることが可能である。

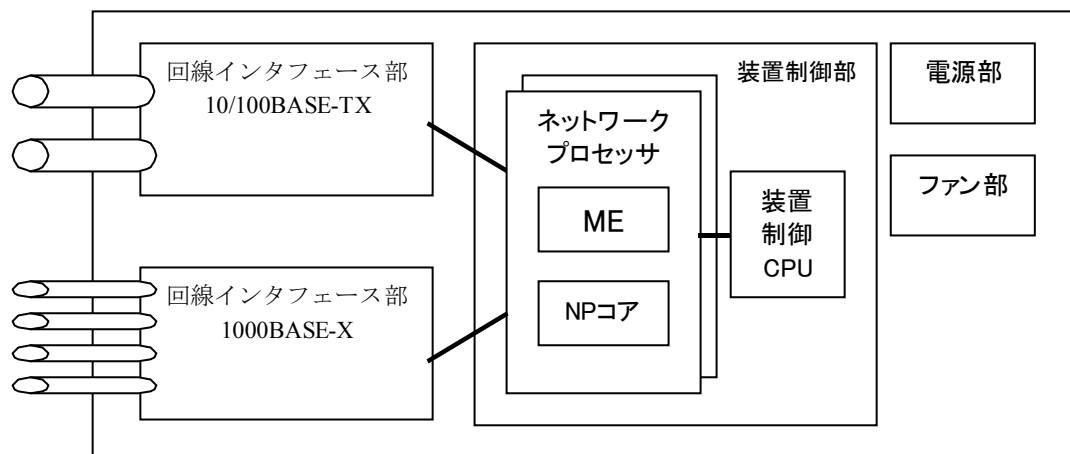


図 2 AG シリーズのシステム構成概要図

2 AG における QoS 機能

AG シリーズに次に示す QoS 機能を追加することとした。

- ① フロー識別機能
- ② キュー制御機能
- ③ ToS-CoS 変換機能

以下に、各機能についての詳細を示す。

2.1 フロー識別機能

フロー識別機能は、ネットワークプロセッサ内のマイクロエンジンにて、フロー識別テーブルを設けることで実現した。図3にフロー識別テーブルの概略図を示す。下図に示すように、フローを識別するための条件を示すフロー識別ルールはフロー識別ルールリストにまとめられ、フロー識別ルールリストはルールに合致した際の動作を示すアクションと関連付けられる。また、そのフロー識別ルールリストとアクションの結びつけられたものを複数まとめてポリシーリストを構成し、1つのポリシーリストが1つのサブインタフェースに対して割り当てられる。

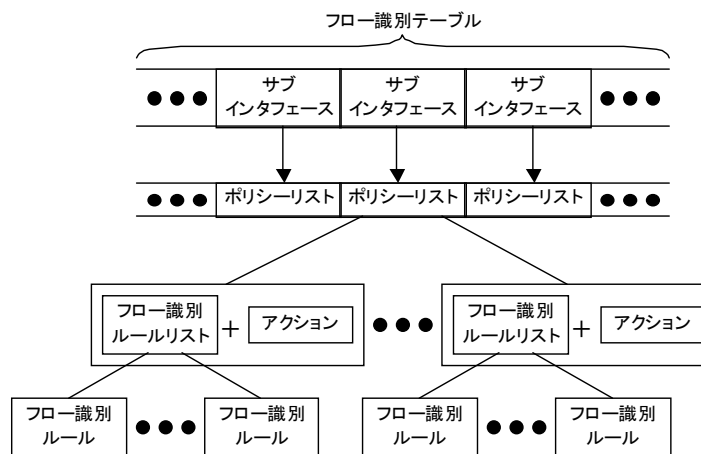


図3 サブインタフェースとフロー識別機能の関連

リスト内の優先順位の高いフロー識別ルールから判定を行い、あるルールに合致した場合は、そのルールが属するリストに対して設定されたアクション（フロー制御）を行う。全てのリストに合致しなかった場合は、予め定義してあるデフォルトのアクション（フロー制御）を行う。フロー識別は、以前の送受信状態などの状態と関連付けたフロー識別は行わず、個々のパケット毎に行う。

表1にフロー識別に利用することが可能な識別ルールの一覧を示す。

表1 フロー識別ルール一覧

#	レイヤ	制御ルール構成要素	備考
1	レイヤ 2	送信元 MAC アドレス	
2		送信先 MAC アドレス	
3		VLAN ID	
4		CoS(Class of Service)値	
5		Ethernet 種別	
6	レイヤ 3	送信元 IP アドレス	ネットワーク指定可能
7		送信先 IP アドレス	ネットワーク指定可能
8		プロトコル種別(レイヤ 4)	IP ヘッダ内のため
9		ToS(Type of service)値	
10	レイヤ 4	TCP 制御フラグ	L4 プロトコル種別が TCP のみ
11		ICMP 種別	L4 プロトコル種別が UDP/TCP 以外
12		ICMP コード	L4 プロトコル種別が UDP/TCP 以外
13		送信元ポート番号	L4 プロトコル種別が UDP/TCP のみ 範囲指定も可能
14		送信先ポート番号	L4 プロトコル種別が UDP/TCP のみ 範囲指定も可能

上記 14 項目の制御ルール構成要素は、レイヤ 4 の構成要素については、上表の備考欄に示したレイヤ 3 のプロトコル種別との組合せで一部制限があるものの、それ以外についてはレイヤ間を含めて任意の組合せを選択して設定することが可能であり、同一ルール内の条件は、“AND 条件”となる。同一フロー制御ルールリスト内の他のフロー制御ルールとは“OR 条件”となる。

2.2 フロー制御機能

フロー制御機能については、ネットワークプロセッサに搭載されている機能を用いて実現することとした。ネットワークプロセッサでは、各物理ポ

ポートに対して優先キューと非優先キューの2つのキューが割り当てられており、さらに Egress 側に1つあたり最大 2,048 個のキューを作成でき、特定の物理ポートのキューと関連付けることが可能である。また、Ingress 側では最大 1,024 フローに対して流量の測定を行い、マーキング制御を行うことが可能である。

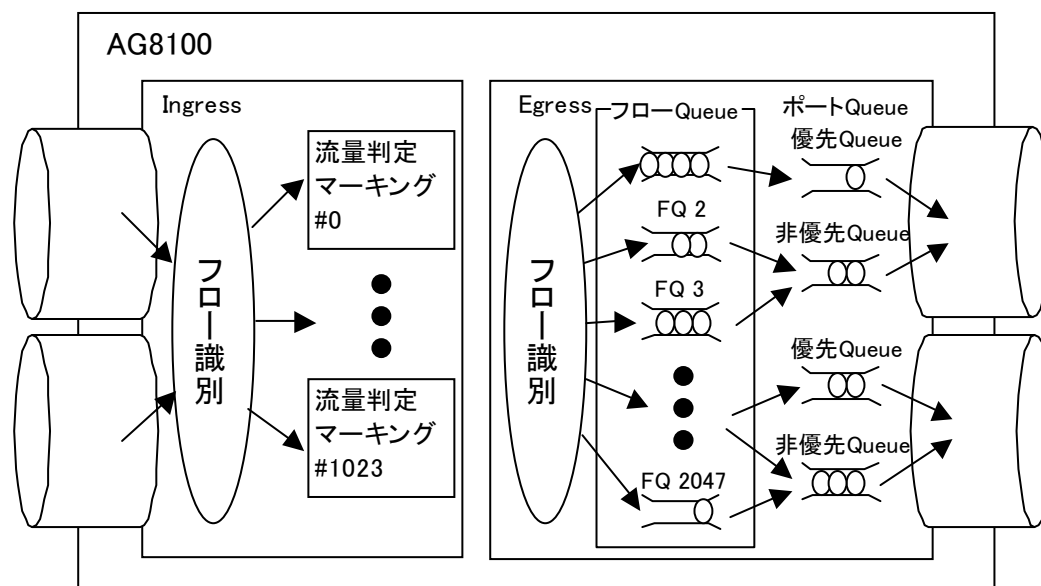


図4 フロー制御機能の概略図

ネットワークプロセッサに搭載されているスケジューリング機能を表2に示す。

表2 ネットワークプロセッサに搭載されているスケジューリング機能一覧

#	適用箇所	機能	備考
1	Ingress	Single Rate Three Color Marker(srTCM)	
2		Two Rate Three Color Marker(trTCM)	
3	Egress	Low Latency Sustainable bandwidth(LLS)	1 キュー／ 物理ポート
4		Normal Latency Sustainable bandwidth(NLS)	
5		Peak Bandwidth Service(PBS)	
6		Queue Weight	
7		Random Early Detection(RED)	

以下に各機能について簡単に説明する。

[1] Single Rate Three Color Marker(srTCM)

Single Rate Three Color Marker(srTCM)とは、RFC2697 に定義されているパケットの流量監視及び流量に応じたマーキングを行う装置または機能を指す。流量判定のために、1つのレート Committed Information Rate (CIR)と、2つのバーストサイズ Committed Burst Size (CBS)、Excess Burst Size (EBS)のパラメータを使用する。ピークレートではなく、バースト長がサービスに影響する場合に用いられる。マーキングについては、IP ヘッダ内の ToS フィールドを書き換えることを行い、CBS を超えていない場合は緑色、CBS を超えているが EBS を超えていない場合は黄色、EBS も超えている場合は赤色にそれぞれマーキングされる。主に、Diff-Serv 内のトラフィック調整に用いられる。

[2] Two Rate Three Color Marker(trTCM)

Two Rate Three Color Marker(trTCM)とは、RFC2698 に定義されているパケットの流量監視及び流量に応じたマーキングを行う装置または機能を指す。srTCM との違いは、流量測定のために、2つのレート Committed Information Rate (CIR)、Peak Information Rate (PIR)と、2つのバーストサイズ Committed Burst Size (CBS)、Peak Burst Size (PBS)のパラメータを使用する。2つのピークレートとそれぞれに関連付けられたバースト長で流量を測定する。マーキング

については、srTCM 同様に、IP ヘッダ内の ToS フィールドを書き換えることを行い、CIR を超えていなければ緑色、CIR を超えているが PIR を超えていなければ黄色、PIR を超えている場合は赤色にマーキングされる。主に、srTCM 同様、Diff-Serv 内のトラフィック調整などに用いられる。

[3] Low Latency Sustainable bandwidth(LLS)

Low Latency Sustainable bandwidth(LLS)は、キュー制御パラメータであり、キュー単位に低遅延の保証帯域シェーピングを行う機能である。本設定を行えるキューは、1つのポートに対して1つのキューに限定され、このキューに積まれたパケットのみが出力ポートの優先キューに載せかえられる。また、この設定のみの場合、シェーピング機能となるため、帯域が余っている状態で該当キューに保証された帯域以上のパケットが入ってきたとしても、設定された保証帯域以上のパケットを流すことはしない。本設定により保証された帯域に加えてさらに余った帯域を利用する（いわゆる最低帯域保証に相当する）ためには、weight 機能などと組み合わせる必要がある。

[4] Normal Latency Sustainable bandwidth(NLS)

Normal Latency Sustainable bandwidth(NLS)は、キュー制御パラメータであり、キュー単位に通常遅延の保証帯域シェーピングを行う機能である。本設定を行えるキューは、同一ポートに対して複数作成することが可能であり、このキューに積まれたパケットは出力ポートの非優先キューに載せかえられる。LLS 同様、この設定のみの場合、シェーピング機能となるため、帯域が余っている状態で該当キューに保証された帯域以上のパケットが入ってきたとしても、設定された保証帯域以上のパケットを流すことはしない。本設定により保証された帯域に加えてさらに余った帯域を利用する（いわゆる最低帯域保証に相当する）ためには、weight 機能などと組み合わせる必要がある。1つのキューで、LLS と NLS の両方を同時に設定することはできない。

[5] Peak Bandwidth Service(PBS)

Peak Bandwidth Service(PBS)は、キュー制御パラメータであり、キュー単位にピークレート制御を行う機能である。本設定を行えるキューは、同一ポートに対して複数作成することが可能である。この設定のみの場合、あくまでピークレートの制限を行うだけであり、帯域保証は全くされない。また、後ほど表3に示すように、LLS や NLS と組み合わせ設定することも可能であるが、その場合、LLS や NLS の設定値からの値を設定する(例えば、500 Mbps に制限したい場合で LLS が 200 Mbps と設定されている場合は、PBS の設定値は 300 Mbps となる)。

[6] Weight

Weight は、キュー制御のパラメータであり、キュー単位にキューからパケットを取り出す重み付けを行う機能である。本設定を行えるキューは、同一ポートに対して複数作成することが可能である。1/X の X を指定する形式で設定するが、同一ポート内で合計が 1 以下でもよいし、1 以上になっても Weight で設定された割合で空き帯域を分割して送信する。

[7] Random Early Detection(RED)

Random Early Detection(RED)は、キュー制御のパラメータであり、パケットの廃棄をキューがいっぱいになる以前に統計的にランダムに廃棄する機能である。本機能を行えるキューは、同一ポートに対して複数作成することが可能である。本機能が設定されていない場合、キューがいっぱいになるとパケットが廃棄される（一般に Tail Drop 方式と呼ばれる）。Tail Drop 方式のルータにてトラフィックの輻輳が発生した場合には、キュー長が短いと TCP パケットなどの廃棄により各プロトコルの再送が頻繁に発生してしまったり、キュー長が長すぎて遅延が発生してしまったりといった問題が生じる。これに対し、RED 方式の場合、それらの問題を軽減することが可能である。

Egress 側で使用可能な機能は表 3 に示す組合せにて用いることが可能である。

表 3 Egress 側にて使用可能なフロー制御機能の組合せ

QoS 制御	LLS	NLS	Weight	PBS
保証帯域シェーピング(低遅延)	○			
保証帯域シェーピング(通常遅延)		○		
重み付けベストエフォート			○	
ピーク制御 + 重み付けベストエフォート			○	○
保証帯域シェーピング(低遅延) + 重み付けベストエフォート	○		○	
保証帯域シェーピング(通常遅延)		○	○	

+ 重み付けベストエフォート				
保証帯域シェーピング(低遅延) + ピーク制御 + 重み付けベストエフォート	○		○	○
保証帯域シェーピング(通常遅延) + ピーク制御 + 重み付けベストエフォート		○	○	○

2.3 ToS-CoS 変換機能

ToS-CoS 変換機能としては、以下に示す2つの方式を実施する。

自動変換方式 … 平成14年度の報告書にて報告した IEEE802.1p が推奨する優先度と CoS 対応付けに従って変換する方式

指定変換方式 … 「2.1 フロー識別機能」の ToS 値（または CoS 値）を指定して識別したパケットに対して、指定した CoS 値（または ToS 値）を設定する方式

また、ToS 値または CoS 値付加（更新）機能として、「2.1 フロー識別機能」の ToS 値以外の条件で識別したフローに対して、特定の ToS 値または CoS 値をつけることも可能とする。

3 性能評価

今回、AG に対して搭載した QoS 機能に対する性能評価試験を行った。評価方法としては、AG 単体としての性能評価と AG をレイヤ2とレイヤ3の間に配置したネットワーク構成での性能評価を行った。

3.1 AG 単体としての性能評価

AG 単体の性能評価として、測定した評価項目について、表4に示す。

表4 AG 単体としての性能評価項目

#	評価項目	備考
A-1	スループット	

I-1	trTCM + 伝送確率	
I-2	CoS-ToS 変換	
I-3	ToS 付与	
E-1	保証帯域シェーピング(通常遅延)	NLS
E-2	重み付けベストエフォート	Weight
E-3	ToS-CoS 変換	
E-4	CoS 付与	

以下に、上記評価項目の実施結果について報告する。

[A-1] スループット測定

本項では、AGにおけるスループットの性能評価を行った結果について示す。本評価試験を行った際のネットワーク構成図を図5に示す。下図のように、AG-AX4000間及びAG-SmartBit間をGigabit Ethernet回線で接続し、それぞれから1 Gbpsのトラフィックを送信しそれぞれに戻すことで測定を行った。また、1つのパケット長を64 Byte, 128 Byte, 256 Byte, 512 Byte, 1024 Byte, 1518 Byteの6パターンに変更してそれぞれのスループット測定を実施した。Ingress側およびEgress側のどちらも、QoSに関する設定は行わない状態で実施した。但し、Egress側ではポート毎に設定されているフロー制御キューのデフォルトキューを使用した状態となる。

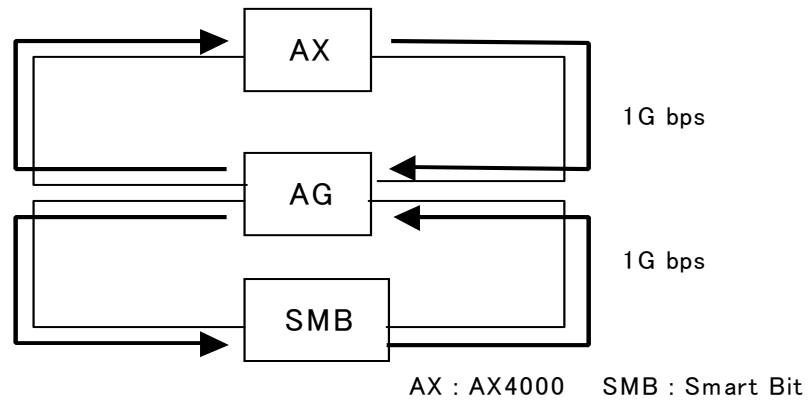


図5 AG単体のスループット測定試験の環境

AG単体としてのスループット測定を実施した結果を表5に示す。

表5 AG単体のスループット測定実施結果

#	パケット長 (Byte/packet)	スループット (Mbps)
1	64	1027
2	128	1803
3	256	2000
4	512	2000
5	1024	2000
6	1518	2000

上記結果により、入力パケット長が 64 Byte/packet のショートパケットの場合でも、1 Gbps のスループットを実現することができ、パケット長が 256 Byte/packet 以上であれば、2 Gbps 以上の性能を発揮することが可能であることが判明した。

[I-1] trTCM + 伝送確率

本項では、Ingress 側の機能である Two Rate Three Color Marker(trTCM)に関する性能評価を実施した結果について示す。1 フローのみを対象とした場合と、複数フローを対象とした場合についての結果を示す。

① フローのみの評価結果

本試験を行った際のネットワーク構成を図 6 に示す。本試験の評価手順は、下図のように AG-AX4000 間を Gigabit Ethernet 回線で接続し、AX4000 より AG に対して制御対象となるフローのパケットを 1 Gbps で送信し、AG 内では受信したパケットを trTCM 機能によるカラーリング処理を実施すると共に、カラーリングされたパケットのカラー毎に設定された伝送確率による伝送または廃棄処理を行わせ、その AG の制御結果に基づいたパケット出力レートを AX4000 に戻ってきた流量から判定する評価を行った。

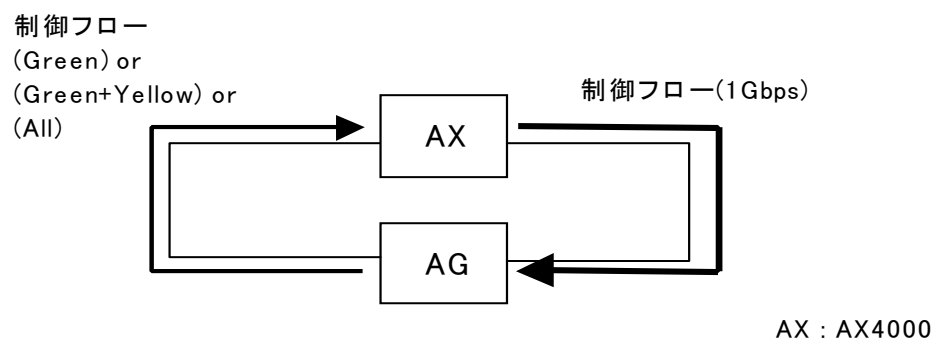


図 6 trTCM 性能評価 (1 フロー) 試験の構成図

以下に評価結果を示す。具体的な設定値としては、trTCMのパラメータを表6のように設定し、カラーリングされた後のパケットの伝送確率を変化させることで性能確認を行った。伝送確率の設定値と性能評価結果を表7に示す。

表6 trTCM 性能評価 (1 フロー) 試験設定パラメータ

QoS パラメータ	設定値
Committed Information Rate(CIR)	100 (Mbps)
Peak Information Rate(PIR)	200 (Mbps)
Committed Burst Size (CBS)	3,000
Peak Burst Size (PBS)	10,000

表7 trTCM 性能評価 (1 フロー) 試験結果

設定 (color 毎パケット伝送確率%)	出力レート (Mbps)
Greenのみ通過許可 (Red : 0%,Yellow : 0%,Green : 100%)	107
Green + Yellowのみ通過許可 (Red : 0%,Yellow : 100%,Green : 100%)	210
全ての色を通過許可 (Red : 100%,Yellow : 100%,Green : 100%)	1,000

また、上記試験の際に、AX4000で受信したパケットをキャプチャーすることで、それぞれのパケットに対してカラーリング (ToS フィールドの該当ビットが更新) が行われていることを確認した。

以上により、trTCMが1フローに対して設定値通りに動作していることが確認できた。

② 複数フローの評価結果

本試験を行った際のネットワーク構成図を図7に示す。本試験の評価手順としては、下図に示すように、AG-AX4000間を Gigabit Ethernet 回線で接続し、AX4000よりAGに対してアドレスなどの異なる複数のフローを同時に送信し、AGではフロー毎に受信したパケットに対してtrTCM機能によるカラーリング処理を実施すると共に、カラーリングされたパケットのカラー毎に設定された伝送確率による伝送または廃棄処理を行わせ、そのAGの制御結果に基づいたパケット出力レートをAX4000に戻ってきた流量から判定する試験を行った。

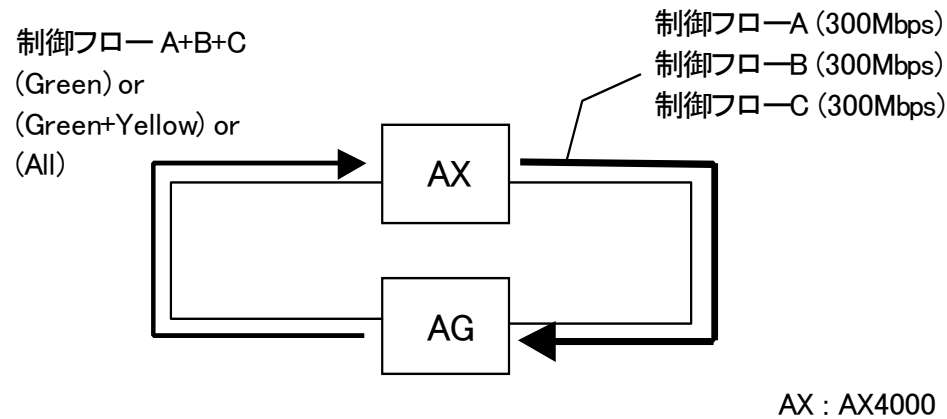


図7 trTCM性能評価（複数フロー）試験の構成図

以下に評価結果を示す。制御対象フローは3フロー（フローA/B/C）とし、AX4000からそれぞれ300Mbpsずつ送信した。具体的な設定値としては、trTCMのパラメータを表8のように設定し、カラーリングされた後のパケットの伝送確率を変化させることで性能確認を行った。伝送確率の設定値とフロー毎の入力レートおよび性能評価結果を表9に示す。

表8 trTCM性能評価（複数フロー）試験設定パラメータ

制御フロー	QoS パラメータ	設定値
フローA	Committed Information Rate(CIR)	100 (Mbps)
	Peak Information Rate(PIR)	200 (Mbps)
	Committed Burst Size (CBS)	3,000
	Peak Burst Size (PBS)	10,000
フローB	Committed Information Rate(CIR)	150 (Mbps)
	Peak Information Rate(PIR)	250 (Mbps)
	Committed Burst Size (CBS)	3,000
	Peak Burst Size (PBS)	10,000
フローC	Committed Information Rate(CIR)	50 (Mbps)
	Peak Information Rate(PIR)	150 (Mbps)
	Committed Burst Size (CBS)	3,000
	Peak Burst Size (PBS)	10,000

表 9 srTCM 性能評価(複数フロー)試験結果

設定 (color 毎パケット伝送確率%)	フロー	出力レート (Mbps)
Green のみ通過許可 (Red : 0%,Yellow : 0%,Green : 100%)	A	104.3
	B	155.5
	C	52.9
Green + Yellow のみ通過許可 (Red : 0%,Yellow : 100%,Green : 100%)	A	206.4
	B	257.5
	C	155.5
全ての色を通過許可 (Red : 100%, Yellow : 100%, Green : 100%)	A	300.0
	B	300.0
	C	300.0

また、上記試験の際に、AX4000 で受信したパケットをキャプチャーすることで、それぞれのパケットに対してカラーリング (ToS フィールドの該当ビットが更新) が行われていることを確認した。

以上により、trTCM が複数のフローに対して設定値通りに動作していることが確認できた。

[I-2] CoS-ToS 変換

CoS-ToS 変換については、AG にて CoS 値をフローの識別に利用し、その CoS に対する変換後の ToS 値をルールとして定義した状態で、AX から AG に対して CoS を設定したパケットを送信し、AG を経由したパケットを AX にてキャプチャーして確認するという手法で行った。

その結果、変換設定がされている CoS 値である全てのパケットが設定通りに ToS 値に変換されること、及び、変換設定がされていない CoS 値である全てのパケットの ToS 値が 0 のままであることが確認できた。

[I-3] ToS 付与

ToS 付与については、AG にて IP アドレス情報などをフローの識別に利用し、その対象パケットに対して、ToS フィールドに値を設定する定義をした状態で、AX から AG に対してフロー識別ルールに該当するパケットを含む複数のパケットを送信し、AG を経由したパケットを AX にてキャプチャして確認するといった手法で行った。

その結果、フロー識別ルールに該当するパケットの ToS フィールドが設定値に更新されていること、及び、その他のパケットの ToS フィールドの値が更新されていないことが確認できた。

[E-1] 保証帯域シェーピング（通常遅延）：NLS

本項では、保証帯域シェーピング（通常遅延）：NLS の設定を行った場合の性能評価結果について示す。1 フローのみを対象とした場合と、複数フローを対象とした場合についての結果を示す。

① 1 フローのみの評価結果

本試験を行った際のネットワーク構成図を図 8 に示す。本試験では、制御フローの他に非制御フロー（いわゆる JAM パケット）を発生させる必要があるため、制御対象のフローを発生させる AX4000 および JAM パケットを発生させる SmartBits 装置と AG 間をそれぞれ Gigabit Ethernet 回線で接続した。試験手法としては、常に 1 Gbps のレートで JAM パケットを流した状態で、制御フローの流量を変化させる方法で実施した。

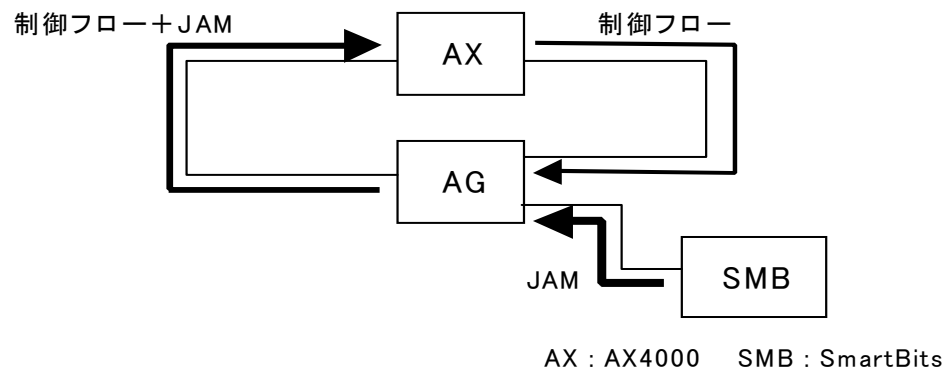


図8 保証帯域シェーピング（通常遅延）フロー評価試験の構成図

以下に試験結果を示す。具体的な設定値としては、フローに対して 600 Mbps の帯域を NLS として設定した。フローの識別条件としては、パケットの送信元 IP アドレスおよび送信先 IP アドレスを用いた。制御対象フローのパケットを 100 Mbps ずつ増加させていった場合の評価結果を表 10 および図 9 に示す。

表 10 帯域保証シェーピング（通常遅延：1 フロー）の性能評価結果

フローA 入力	100	200	300	400	500	600	700	800	900	1000
フローA 出力	98.8	197.6	296.4	395.2	494.0	593.3	597.4	596.7	596.3	595.6
JAM 出力	900.6	801.3	703.3	596.2	505.2	406.0	401.6	402.5	402.6	403.3

(単位：Mbps)

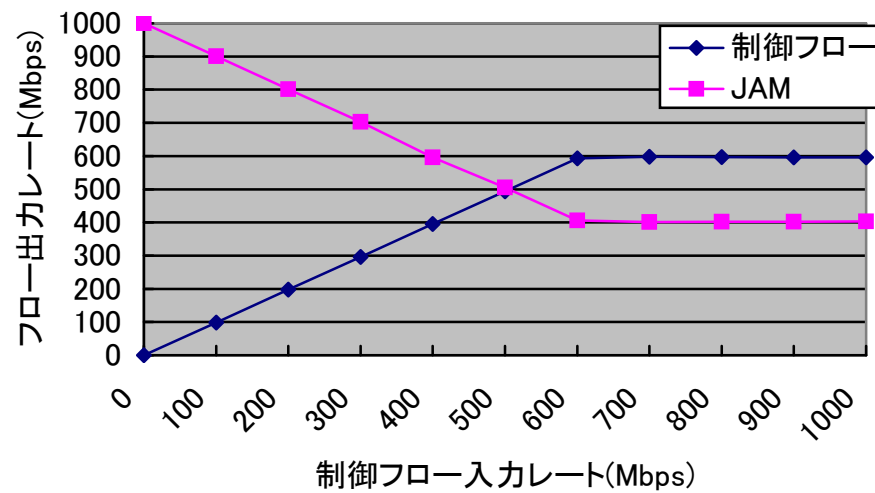


図9 帯域保証シェーピング（通常遅延：1フロー）の性能評価結果グラフ

上記結果の通り、入力レートが保証帯域以下の場合、AGが輻轉した状態でも出力レートは保証されるが、保証帯域を超えると保証帯域以上のレートについてはシェーピングされる動作となり、設定通りの動作をすることが確認できた。

② 複数フローの性能評価

本試験を行った際のネットワーク構成図を図10に示す。本試験では、1フローの時と同様に、制御対象のフローを発生させるAX4000およびJAMパケットを発生させるSmartBits装置とAG間をそれぞれGigabit Ethernet回線で接続した。試験手法としては、常に1GbpsのレートでJAMパケットを流した状態で、複数の制御フローを保証帯域以上の流量で送信することで確認を行った。

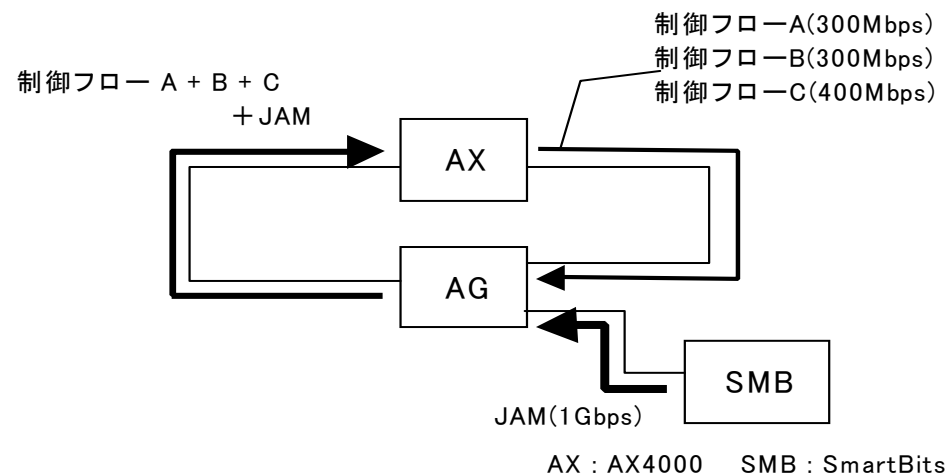


図 10 保証帯域シェーピング（通常遅延：複数フロー）評価試験の構成図

以下に試験結果を示す。各フローに対する AG における入力レートおよび保証帯域は表 11 に示すように設定した。各フローの識別条件としては、パケットの送信元 IP アドレスおよび送信先 IP アドレスを用いた。

表 11 保証帯域シェーピング（通常遅延：複数フロー）評価試験の設定パラメータ

フロー	入力レート (Mbps)	保証帯域設定(Mbps)
制御フローA	300	200
制御フローB	300	250
制御フローC	400	300
JAM	1,000	なし

測定結果は表 12 に示すようになった。

表 12 保証帯域シェーピング（通常遅延：複数フロー）評価試験の測定結果

フロー	出力レート(Mbps)	遅延時間(μ sec)
制御フローA	197.7	
制御フローB	248.8	
制御フローC	299.0	
Jam	253.6	—

[E-2] 重み付けベストエフォート：Weight

本項では、重み付けベストエフォート：Weight の設定を行った場合の性能評価結果について示す。重み付けによるベストエフォートのため、1 フローでは意味をもたないため、複数フローを対象とした場合についての結果を示す。

本試験を行った際のネットワーク構成図を図 11 に示す。本試験では、1 Gbps の 2 つの入力フローを作成するために、AX4000 と SmartBits を AG にそれぞれ Gigabit Ethernet 回線で接続した。試験手順としては、フローA,B に対する重み付けの組合せを変更させて測定を行った。

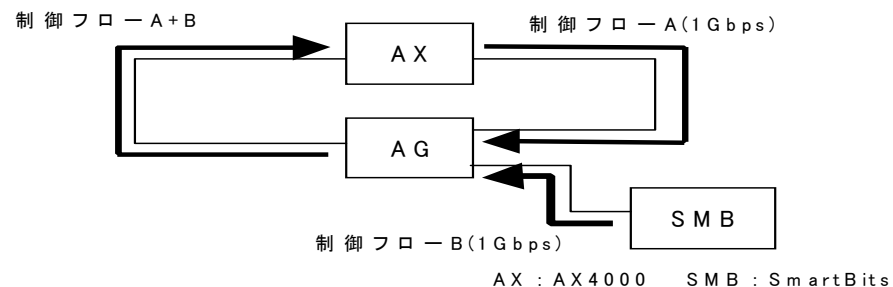


図 11 重み付けベストエフォート評価試験の構成図

以下に試験結果を示す。フローAの重み付けとフローBの重み付けを組み合わせたそれぞれの設定値における試験結果を表13に示す。左端がフローAに対して設定した重み付けであり、上端がフローBに対して設定した重み付けである。交差した柵の上段の数字がフローAの出力レートを、下段の数字がフローBの出力レートをそれぞれ示す。

表 13 重み付けベストエフォートの評価試験結果

A \ B	1/1	1/2	1/3	1/4	1/7	1/15	1/31
1/1	500.0	666.7	750.0	800.0	875.0	937.5	968.8
	500.0	333.3	250.0	200.0	125.0	62.5	31.2
1/2		500.0	600.0	666.7	777.8	882.4	939.4
		500.0	400.0	333.3	222.2	117.6	60.6
1/3			500.0	571.4	700.0	833.3	911.8
			500.0	428.6	300.0	166.7	88.2
1/4				500.0	636.4	789.5	885.7
				500.0	363.6	210.5	114.3
1/7					500.0	681.8	815.8
					500.0	318.2	184.2
1/15						500.0	673.3
						500.0	325.8
1/31							500.0
							500.0

フローAを1/1に設定した状態で、フローBの重み付けを変化させた場合の各フローの出力レートを図12に示す。

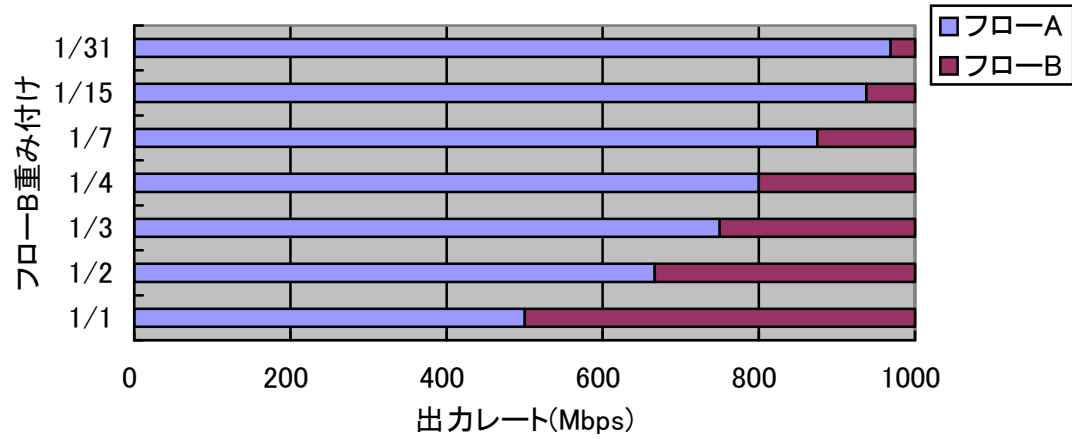


図12 重み付け出力制御 (A:1/1 固定, B:縦軸の値) の評価結果

グラフ

また、4フローを対象にした場合の設定値を表14に測定結果を図13に示す。

表 14 重み付け出力制御(4 フロー)の評価試験設定値

フロー	入力レート(Mbps)	重み付け
フローA	500	1/1
フローB	500	1/2
フローC	500	1/3
フローD	500	1/4

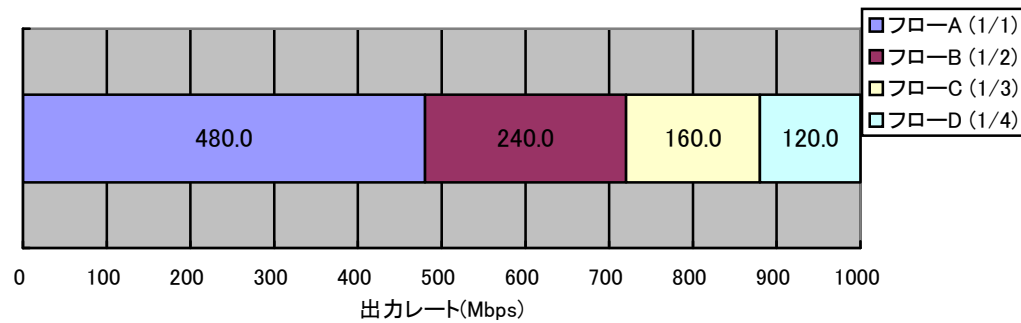


図 13 重み付け出力制御 (4 フロー) の評価試験結果グラフ

上記結果により、設定値通りの動作をすることが確認できた。

[E-3] ToS-CoS 変換

ToS-CoS 変換については、AG にて ToS 値をフローの識別に利用し、その ToS に対する変換後の CoS 値をルールとして定義した状態で、AX から AG

に対して ToS を設定したパケットを送信し、AG を経由したパケットを AX にてキャプチャして確認するという手法で行った。

その結果、変換設定がされている ToS 値である全てのパケットが設定通りに CoS 値が変換されること、及び、変換設定がされていない ToS 値である全てのパケットの CoS 値が 0 のままであることが確認できた。

[E-4] CoS 付与

CoS 付与については、AG にて IP アドレス情報などをフローの識別に利用し、その対象パケットに対して、CoS フィールドに値を設定する定義をした状態で、AX から AG に対してフロー識別ルールに該当するパケットを含む複数のパケットを送信し、AG を経由したパケット AX にてキャプチャして確認するといった手法で行った。

その結果、フロー識別ルールに該当するパケットの CoS フィールドが設定値に更新されていること、及び、その他のパケットの CoS フィールド値が更新されていないことが確認できた。

3.2 Ethernet ネットワーク-IP ネットワークとしての性能評価

Ethernet ネットワーク及び IP ネットワークからなるネットワークに本研究による QoS 制御機能を搭載した AG を適用し連携させた場合の性能評価項目について、表 15 に示す。

表 15 Ethernet ネットワーク-IP ネットワークとしての性能評価

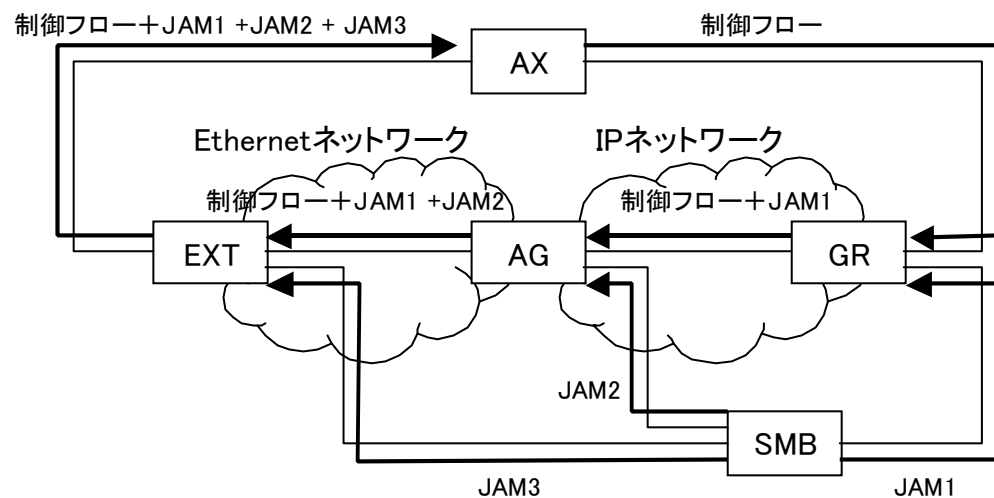
#	評価項目	備考
1	帯域保証	

[1] 帯域保証

本項では、Ethernet ネットワーク-IP ネットワークにおいて帯域保証を行った場合の性能評価結果について示す。IP ネットワーク→Ethernet ネットワーク方向（下り）および Ethernet ネットワーク→IP ネットワーク方向（上り）において、1 フローおよび複数フローの動作を確認した。

<IP ネットワーク→Ethernet ネットワーク方向（下り）>

IP ネットワーク→Ethernet ネットワーク方向（下り）の連携ネットワークとしての性能評価環境を図 14 に示す。下図に示すように Ethernet ネットワーク側のネットワーク装置としては Extreme を、IP ネットワーク側のネットワーク装置としては GR2000 を使用し、Ethernet ネットワークと IP ネットワークの間に AG を設置する構成とした。各装置間のインタフェースは Gigabit Ethernet 回線で接続した。また、制御フローを発生させ各ネットワークを経て戻ってきた結果を測定する装置として AX4000 を、非制御フローを発生させる装置として SmartBits をそれぞれ Gigabit Ethernet 回線で接続した。



AX:AX4000, SMB:SmartBits, EXT : Extreme, GR : GR2000

図 14 Ethernet-IP ネットワークの性能評価環境（下り）

① 1 フローのみの性能評価

1 フローのみの性能評価をするために、IP ネットワークでは、ToS 値が 128 のフローに対しては、300Mbps を帯域保証する想定とし、Ethernet ネット

ワークでは、CoS 値が 1 のフローに対して 300Mbps を帯域保証する想定で評価を行った。

実際には、IP ネットワークにおいては、表 16 及び表 17 に示すように、ToS 値が 128 のフローに対して、GR2000 では Gigabit Ethernet 回線の帯域の 30%保証(300Mbps 保証)、AG では保証帯域シェーピング（通常遅延）として 300Mbps をそれぞれ設定し、ToS 値が 0 のフローに対しては、特に制限を設けない設定とした。また、AG において、ToS-CoS 変換を有効にした場合は、ToS 値が 128 で入ってきたパケットに対しては、CoS 値を 1 に設定し、ToS 値が 0 で入ってきたパケットに対しては、CoS 値を 0（設定しない）とする設定とした。

一方、Ethernet ネットワークについては、表 18 に示すように、CoS 値が 1 のフローに対して、Extreme にて Gigabit Ethernet 回線の帯域の 30%保証(300Mbps 保証)とし、CoS 値が 0 のフローに対しては、特に制限を設けない設定とした。

表 16 GR2000 における帯域保証（1 フロー）の QoS 制御パラメータの設定値

	帯域保証	
ToS 128	MIN	30 %
	MAX	—
ToS 0	MIN	0 %
	MAX	100 %

表 17 AG における帯域保証（1 フロー）の QoS 制御パラメータの設定値

	帯域保証		ToS-CoS 変換
ToS 128	NLS	300 (Mbps)	CoS 1
	PBS	—	
ToS 0	NLS	—	CoS 0
	PBS	1000 (Mbps)	

表 18 Extreme における帯域制御（1 フロー）QoS 制御パラメータの設定値

	帯域保証	
CoS 1	MIN	30 %
	MAX	—
CoS 0	MIN	0 %
	MAX	100 %

上記のような設定に対して、表 19 に示すようなフローA、B 及び Jam1～3 のパケットをそれぞれ流して、パケットの流量および遅延時間の測定を行った。但し、遅延時間の測定については、測定機器の制限にてロスパケットが多い場合には、正確な情報を取得するのが困難なため、ロスパケットがほとんどないフローのみ測定した。また、Jam 3 のフローについては、Ethernet ネットワークを輻輳させる場合にのみ、500 Mbps を入力した。

表 19 各フローの評価ネットワークへの投入レート

フロー	入力レート	ToS 値
フローA	300 (Mbps)	128
フローB	700 (Mbps)	0
Jam 1	500 (Mbps)	0
Jam 2	1,000 (Mbps)	0
Jam 3	0 or 500 (Mbps)	0

上記条件における試験結果を表 20 に示す。

表 20 Ethernet-IP ネットワーク連携帯域保証 (1 フロー) の測定結果

条件	フロー	出力レート(Mbps)	平均遅延時間 (μ sec)
----	-----	-------------	------------------------

ToS-CoS 変換なし Ethernet ネットワーク 輻輳なし (Jam 3 : 0 Mbps)	フローA	296.0	254.1
	フローB	149.6	—
	Jam 1	121.7	—
	Jam 2	432.7	—
	Jam 3	—	—
ToS-CoS 変換なし Ethernet ネットワーク 輻輳あり (Jam 3 : 500 Mbps)	フローA	196.4	—
	フローB	104.1	—
	Jam 1	78.8	—
	Jam 2	288.3	—
	Jam 3	331.6	—
ToS-CoS 変換あり Ethernet ネットワーク 輻輳なし (Jam 3 : 0 Mbps)	フローA	295.7	327.5
	フローB	161.3	—
	Jam 1	124.2	—
	Jam 2	418.4	—
	Jam 3	—	—
ToS-CoS 変換あり Ethernet ネットワーク 輻輳あり (Jam 3 : 500 Mbps)	フローA	295.9	323.5
	フローB	94.1	—
	Jam 1	71.6	—
	Jam 2	244.3	—
	Jam 3	293.6	—

上記結果の通り、AGにおいてToS-CoS変換をしない場合、フローAは、Ethernetネットワークが輻輳していない場合は、IPネットワークが輻輳していても保証している300Mbpsの出力をほぼ利用することができるが、Ethernetネットワークが輻輳している場合、Ethernetネットワーク内でパケットロ

スが発生し、保証しなかった 300 Mbps を下回ってしまう。しかしながら、ToS-CoS 変換した場合は、変換後の CoS 値を使用して Ethernet ネットワーク内でも帯域保証が行われるために、Ethernet ネットワークが輻輳していた場合でも、保証しなかった 300 Mbps の帯域を確保することができている。

また、遅延時間についても、約 300 μ sec 程度であり、ToS-CoS 変換をすることによる増加分についても、70~80 μ sec 程度であることがわかった。

② 複数フローの性能評価

複数フローの性能評価をするために、IP ネットワークでは、ToS 値が 4 のフローに対しては 100 Mbps を、ToS 値が 128 のフローに対しては 200 Mbps を、ToS 値が 252 のフローに対しては 300 Mbps を帯域保証する想定とし、Ethernet ネットワークでは、CoS 値が 1 のフローに対して 100 Mbps を、CoS 値が 4 のフローに対して 200 Mbps を、CoS 値が 7 のフローに対しては 300 Mbps を帯域保証する想定で評価を行った。

実際には、IP ネットワークにおいては、表 21 及び表 22 に示すように、ToS 値が 4,128,252 のフローに対して、GR2000 では Gigabit Ethernet 回線の帯域のそれぞれ、10%、20%、30%保証(100 Mbps, 200 Mbps, 300 Mbps 保証)、AG では保証帯域シェーピング(通常遅延)としてそれぞれ 100 Mbps, 200 Mbps, 300 Mbps を設定し、ToS 値が 0 のフローに対しては、特に制限を設けない設定とした。また、AG において、ToS 値が 4 のパケットに対しては CoS 値を 1 に、ToS 値が 128 のパケットに対しては、CoS 値を 4 に、ToS 値が 252 のパケットに対しては、CoS 値を 7 に設定し、ToS 値が 0 で入ってきたパケットに対しては、CoS 値を 0 (設定しない)とする設定とした。

一方、Ethernet ネットワークについては、表 23 に示すように、CoS 値が 1,4,7 のフローに対して、Extreme にてそれぞれ Gigabit Ethernet 回線の帯域の 10%、20%、30%保証(100 Mbps, 200 Mbps, 300 Mbps 保証)とし、CoS 値が 0 のフローに対しては、特に制限を設けない設定とした。

表 21 GR2000 における帯域保証（複数フロー）の QoS 制御パラメータの設定値

	帯域保証	
	ToS 4	MIN
	MAX	—
ToS 128	MIN	20 %
	MAX	—
ToS 252	MIN	30 %

	MAX	—
ToS 0	MIN	0 %
	MAX	100 %

表 22 AG における帯域保証（複数フロー）の QoS 制御パラメータの設定値

	帯域保証		ToS-CoS 変換
ToS 4	NLS	100 (Mbps)	CoS 1
	PBS	—	
ToS 128	NLS	200 (Mbps)	CoS 4
	PBS	—	
ToS 252	NLS	300 (Mbps)	CoS 7
	PBS	—	
ToS 0	NLS	—	CoS 0
	PBS	1000 (Mbps)	

表 23 Extreme における帯域制御（複数フロー）QoS 制御パラメータの設定値

	帯域保証	
CoS 1	MIN	10 %
	MAX	—
CoS 4	MIN	20 %
	MAX	—
CoS 7	MIN	30 %
	MAX	—
CoS 0	MIN	0 %
	MAX	100 %

上記のような設定に対して、表 24 に示すようなフローA～D 及び Jam1～3 のパケットをそれぞれ流して、パケットの流量および遅延時間の測定を行った。但し、遅延時間の測定については、ロスパケットがほとんどないフローのみ測定した。

表 24 各フローの評価ネットワークへの投入レート

フロー	入力レート	ToS 値
フローA	100 (Mbps)	4
フローB	200 (Mbps)	128
フローC	300 (Mbps)	252
フローD	400 (Mbps)	0
Jam 1	500 (Mbps)	0
Jam 2	1,000 (Mbps)	0
Jam 3	500 (Mbps)	0

上記のような条件下での試験結果を表 25 に示す。

表 25 Ethernet-IP ネットワーク連携帯域保証（複数フロー）の測定結果

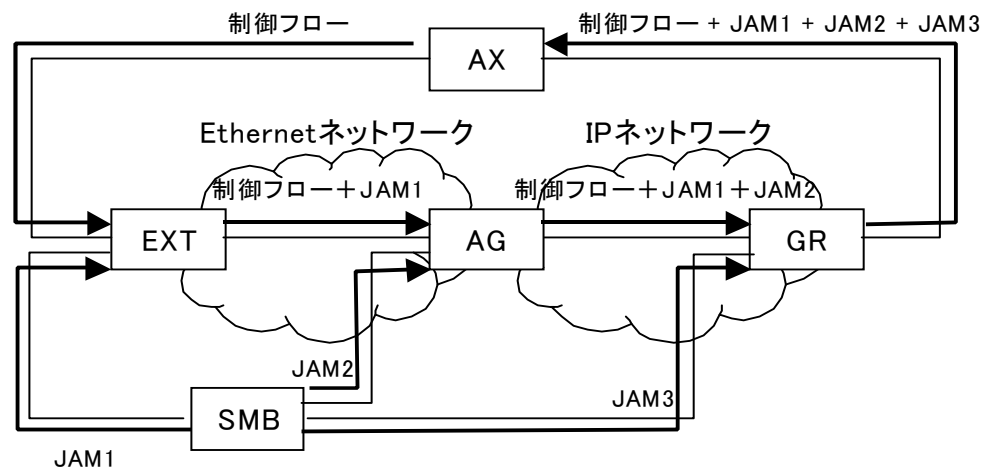
フロー	出力レート (Mbps)	平均遅延時間 (μ sec)
フローA	98.8	520.7
フローB	197.4	420.1
フローC	296.1	334.5
フローD	14.2	—
Jam 1	27.1	—
Jam 2	133.8	—
Jam 3	232.3	—

上記の結果により、複数のフローに対しても、ToS-CoS 変換を行うことで IP ネットワークおよび Ethernet ネットワークが輻輳している場合でも、保証したい帯域を確保することが可能であることが証明できた。

また、遅延時間についても 300~500 μ sec 程度であり、実用する上での問題はないと思われる。

<Ethernet ネットワーク→IP ネットワーク方向（上り）>

IP ネットワーク→Ethernet ネットワーク方向（上り）の連携ネットワークとしての性能評価環境を図 15 に示す。下図に示すように Ethernet ネットワーク側のネットワーク装置としては Extreme を、IP ネットワーク側のネットワーク装置としては GR2000 を使用し、Ethernet ネットワークと IP ネットワークの間に AG を設置する構成とした。各装置間のインターフェースは Gigabit Ethernet 回線で接続した。また、制御フローを発生させ各ネットワークを経て戻ってきた結果を測定する装置として AX4000 を、非制御フローを発生させる装置として SmartBits をそれぞれ Gigabit Ethernet 回線で接続した。



AX : AX4000, SMB : SmartBits, EXT : Extreme, GR : GR2000

図 15 Ethernet-IP ネットワークとしての性能評価環境（上り）

① 1 フローのみの性能評価

1 フローのみの性能評価をするために、下り方向と同様に、IP ネットワークでは、ToS 値が 128 のフローに対しては、300 Mbps を帯域保証する想定とし、Ethernet ネットワークでは、CoS 値が 1 のフローに対して 300 Mbps を帯域保証する想定で評価を行った。

実際には、Ethernet ネットワークでは、表 26 に示すように、Extreme にて Extreme-AG 間の出側のインタフェースに、CoS 値が 1 のフローに対して、Gigabit Ethernet 回線の帯域の 30%保証(300 Mbps 保証)とし、CoS 値が 0 のフローに対しては、特に制限を設けない設定とした。

IP ネットワークにおいては、AG では、Extreme-AG 間を接続する入り側のインタフェースにおいて、表 27 に示すような CoS 値を元にした ToS 値への変換を行い、GR-AG 間を接続する出側のインタフェースでは、表 28 に示すように、ToS 値が 128 のフローに対しては、保証帯域シェーピング（通常遅延）として 300 Mbps を設定し、ToS 値が 0 のフローに対しては、特に制限を設けない設定とした。さらに、GR2000 では GR-AX 間の出側のインタフ

ケースにて、表 29 に示すように、ToS 値が 128 のフローに対して、Gigabit Ethernet 回線の帯域の 30%保証(300 Mbps 保証)をする設定とした。

表 26 Extreme における帯域制御（1 フロー） QoS 制御パラメータの設定値

	帯域保証	
CoS 1	MIN	30 %
	MAX	—
CoS 0	MIN	0 %
	MAX	100 %

表 27 AG における帯域保証（1 フロー）の QoS 制御パラメータの設定値

	CoS-ToS 変換
CoS 1	ToS 128
ToS 0	CoS 0

表 28 AG における帯域保証（1 フロー）の QoS 制御パラメータの設定値

	帯域保証	
ToS 128	NLS	300 (Mbps)
	PBS	—
ToS 0	NLS	—
	PBS	1000 (Mbps)

表 29 GR2000 における帯域保証（1 フロー）の QoS 制御パラメータの設定値

	帯域保証	
ToS 128	MIN	30 %
	MAX	—
CoS 0	MIN	0 %
	MAX	100 %

上記のような設定に対して、表 30 に示すようなフローA、B 及び Jam1～3 のパケットをそれぞれ流して、パケットの流量および遅延時間の測定を行った。但し、遅延時間の測定については、ロスパケットがほとんどないフローのみ測定した。

表 30 各フローの評価ネットワークへの投入レート

フロー	入力レート	CoS 値
フローA	300 (Mbps)	1
フローB	700 (Mbps)	0
Jam 1	500 (Mbps)	0
Jam 2	1,000 (Mbps)	0
Jam 3	0 or 500 (Mbps)	0

上記のような条件下での試験結果を表 31 に示す。

表 31 Ethernet-IP ネットワーク連携帯域保証（複数フロー）の測定結果

フロー	出力レート (Mbps)	平均遅延時間 (μ sec)
フローA	296.3	318.4
フローB	91.5	—
Jam 1	300.8	—
Jam 2	245.4	—
Jam 3	66.6	—

上記の結果により、CoS-ToS 変換を行うことで Ethernet ネットワークおよび IP ネットワークが輻輳している場合でも、保証したい帯域を確保することが可能であることが証明できた。

② 複数フローのみ性能評価

複数フローの性能評価をするために、下り方向と同様に、IP ネットワークでは、ToS 値が 4 のフローに対しては 100 Mbps を、ToS 値が 128 のフローに対しては 200 Mbps を、ToS 値が 252 のフローに対しては 300 Mbps を帯域保証する想定とし、Ethernet ネットワークでは、CoS 値が 1 のフローに対して 100 Mbps を、CoS 値が 4 のフローに対して 200 Mbps を、CoS 値が 7 のフローに対しては 300 Mbps を帯域保証する想定で評価を行った。

実際には、Ethernet ネットワークについては、表 32 に示すように、CoS 値が 1,4,7 のフローに対して、Extreme にてそれぞれ Gigabit Ethernet 回線の帯域の 10%、20%、30%保証(100 Mbps, 200 Mbps, 300 Mbps 保証)とし、CoS 値が 0 のフローに対しては、特に制限を設けない設定とした。

IP ネットワークにおいては、AG では、Extreme-AG 間を接続する入り側のインタフェースにおいて、表 33 に示すように、CoS 値が 1 のパケットに対しては ToS 値を 4 に、CoS 値が 4 のパケットに対しては CoS 値を 128 に、CoS 値が 7 のパケットに対しては ToS 値を 252 に変更し、CoS 値が 0 で入ってきたパケットに対しては ToS 値を 0（設定しない）とする設定とした。また、AG-GR 間を接続する出側のインタフェースにおいては、既に CoS から

ToS への変換が完了しているため、表 34 に示すように、ToS 値が 4,128,252 のフローに対して、保証帯域シェーピング(通常遅延)としてそれぞれ 100 Mbps, 200 Mbps, 300 Mbps を設定した。さらに、表 35 に示すように、GR2000 では、ToS 値が 4,128,252 のフローに対して、Gigabit Ethernet 回線の帯域のそれぞれ、10%, 20%, 30%保証(100 Mbps, 200 Mbps, 300 Mbps 保証)を設定し、ToS 値が 0 のフローに対しては、特に制限を設けない設定とした。

表 32 Extreme における帯域制御（複数フロー）QoS 制御パラメータの設定値

	帯域保証	
	CoS 1	MIN
MAX		—
CoS 4	MIN	20 %
	MAX	—
CoS 7	MIN	30 %
	MAX	—
CoS 0	MIN	0 %
	MAX	100 %

表 33 AG における帯域保証（複数フロー）の QoS 制御パラメータの設定値

	ToS-CoS 変換
CoS 1	ToS 4
CoS 4	ToS 128
CoS 7	ToS 252
CoS 0	ToS 0

表 34 AG における帯域保証（複数フロー）の QoS 制御パラメータの設定値

	帯域保証	
ToS 4	NLS	100 (Mbps)
	PBS	—
ToS 128	NLS	200 (Mbps)
	PBS	—
ToS 252	NLS	300 (Mbps)
	PBS	—
ToS 0	NLS	—
	PBS	1000 (Mbps)

表 35 GR2000 における帯域保証（複数フロー）の QoS 制御パラメータの設定値

	帯域保証	
ToS 4	MIN	10 %
	MAX	—
ToS 128	MIN	20 %
	MAX	—
ToS 252	MIN	30 %
	MAX	—
ToS 0	MIN	0 %
	MAX	100 %

上記のような設定に対して、表 36 に示すようなフローA～D 及び Jam1～3 のパケットをそれぞれ流して、パケットの流量および遅延時間の測定を行った。但し、遅延時間の測定については、ロスパケットがほとんどないフローのみ測定した。

表 36 各フローの評価ネットワークへの投入レート

フロー	入力レート	CoS 値
フローA	100 (Mbps)	1
フローB	200 (Mbps)	4
フローC	300 (Mbps)	7
フローD	400 (Mbps)	0
Jam 1	500 (Mbps)	0
Jam 2	1,000 (Mbps)	0

Jam 3	500 (Mbps)	0
-------	------------	---

上記のような条件下での試験結果を表 37 に示す。

表 37 Ethernet-IP ネットワーク連携帯域保証（複数フロー）の測定結果

フロー	出力レート (Mbps)	平均遅延時間 (μ sec)
フローA	98.8	520.7
フローB	197.4	420.1
フローC	296.1	334.5
フローD	14.2	—
Jam 1	27.1	—
Jam 2	133.8	—
Jam 3	232.3	—

上記の結果により、複数のフローに対しても、CoS-ToS 変換を行うことで Ethernet ネットワークおよび IP ネットワークが輻輳している場合でも、上り方向の保証したい帯域を確保することが可能であることが証明できた。

また、遅延時間についても 300~500 μ sec 程度であり、実用する上での問題はないと思われる。

5-2-3 おわりに

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

本課題において本年度は、平成 14 年度に開発した L2Shaper プロトタイプのエンハンスを行い、以下（1）から（4）に示す機能、性能の実証を行

った。

(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

L2Shaper では、フローの識別機能、キュー制御機能をそれぞれ FreeBSD の NetGraph 機構を用いた独立したカーネルモジュールとして実装することで、フロー識別モジュール、キュー識別モジュールを組み合わせることで多段の識別および品質制御を行うことを実現した。

(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

L2Shaper のキュー制御機能に、キュー制御アルゴリズムとして優先キュー方式、及び重み付きラウンドロビン方式を実装してパラメータにより選択可能とし、優先制御および帯域制御を実現した。

(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

L2Shaper で設定可能なフロー数、キュー数はプログラムの設定ファイルで上限を規定されるが、設定上限数を 1000 として正常に動作することを確認した。

(4) フローの最大帯域は 100 Mbps が可能であること

L2Shaper で性能評価を行い、ギガビットイーサネット NIC 間のフロー制御で、500 Mbps 以上のフローをパケットロスなく転送可能であることを実証し、100 Mbps を大きく超えた性能を持つことを実証した。

また、アプリケーション対応通信品質制御技術に関し、検証用実験コンテンツとしてネットワーク品質に応じた適応制御を行う映像アプリケーションであるビデオアダプタシステムを開発し、DCCP(Datagram Congestion Control Protocol)を利用したアプリケーション対応通信品質制御技術の実験を行い、ネットワークの品質に応じてアプリケーションが適切な映像品質を保ちながら動作することを実証した。

但し、アプリケーションによって協調動作を行うために必要な情報は異なるため、汎用的なインタフェースを確立していくためには今後様々なアプリケーションでの実証実験が必要と考え、これからの課題は大きい。

今後は、階層フロー識別型通信品質制御技術の製品化に向けて検討を行っていく。

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

本課題において本年度は、下記(1)から(3)の仕様を満たす管理サーバのプロトタイプ開発を行い、機能、性能の検証を完了した。

(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

L2Shaper と Extreme 社製イーサスイッチ及び、開発した管理サーバプロトタイプから成る、システムを構築し、正常に制御可能であることを確認し

た。

(2) 1フローの設定が5秒以内に終了すること

L2Shaper と Extreme 社製イーサスイッチを管理対象とし、画面操作から装置コマンド投入までの時間を計測し、1フローあたりの設定時間が5秒以内であることを確認した。

(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

上記管理サーバ上に、1000フローの通信品質制御情報を格納し、この状態でフローの制御が可能であることを確認した。

また、アクセスゲートウェイプロトタイプ、ネットワークプロセッサと呼ばれる専用ハードウェア上に、フロー計測機能を実装し、ギガビットイーサネット回線上のフローの計測が可能であることを確認した。

今後は、L2管理サーバの製品化に向けて検討を行っていく。

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

本課題において本年度は、平成14年度までに検討した課題イ-1の通信品質制御方式及び課題イ-3のIPネットワークとイーサネットワークの連携方式を弊社ブロードバンドアクセスサーバAGに適用するための詳細検討を行い、その通信品質制御方式及び連携方式に準じた機能をAGに実装し評価試験を行った。

評価試験の結果、エッジノードにおいて、通信品質保証をするとともにToS-CoS変換を行い、IPネットワークとイーサネットワークの通信品質制御を連携させることで、End-Endでの品質保証ができることが確認できた。また、AG単体としての性能評価も実施した結果、ほぼ期待通りの動作をすることが確認できたため、製品版のAGに適用する目処がたった。

今後としては、AGの製品化に向けての必要機能についての精査を行い、QoSを実装したAGの市場への早期投入を目指す。

5-3 総括

非平面・大型スクリーンに高精細・広視野角な映像を投影することで、臨場感あふれる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、教育、医療、エンターテインメント、セキュリティ、文化・芸術といった様々な分野における映像情報の新たな表現手段として注目されている。一方、近年のネットワークの広帯域・高速化に伴い、極めて大量のデータ処理が必要とされる IPD アプリケーションについても、そのネットワーク対応化が現実的なものになりつつあり、これは従来のスタンドアロンタイプのものに対し、機能や適用範囲を大きく広げるものとして期待されている。

本研究では以上を鑑み、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、IPD 用映像制作技術ならびにネットワーク制御技術の開発を行ったものである。

映像制作技術については、超大容量映像データを PC クラスタで処理し大容量ストレージで管理するための技術、高精細広視野角映像をマルチカメラなどで撮影するための技術、そのような映像を高臨場感ディスプレイと撮影装置の幾何学的整合性を考慮して編集処理するための技術、また高臨場感ディスプレイにマルチカメラ映像をリアルタイムに伝送・表示するための技術を中心に開発した。ネットワーク制御に関する技術については、アクセス網に構築されたコンテンツ配信ネットワーク(CDN)において通信品質を階層的に保証する技術、イーサ網において通信品質を制御・管理する技術、およびこれらイーサ網と IP 網を連携するための技術を中心に開発した。

平成15年度は、平成13年度から始まった本研究の最終年度にあたり、平成16年3月末における最終目標に向けて、前年度までに開発した各課題の技術検討をベースとして、それらを実装するプロトシステムの開発と検証を主に実施した。PC クラスタによる IPD 用映像編集システムのプロトタイプ開発では、本研究で確立した画像処理並列実行技術、大容量ストレージ管理技術、マルチカメラ等による特殊映像の処理技術などを組み入れた。また遠隔監視アプリケーションを想定したマルチカメラとマルチプロジェクタによるリアルタイム撮影・通信・表示システムのプロトタイプ開発では、複数カメラの撮影映像をシームレスに接続するための画像処理技術やキャリブレーション技術、ならびに他のアプリケーションがネットワークを共有するときも安定な映像配信を実現するネットワーク制御技術などを組み込み、その有効性を確認した。

以上より、IPD 用映像コンテンツ制作に関する撮影から編集、配信、表示に至るまでの幅広い基盤技術を蓄積し、さらに IPD 映像データ等の大容量のデータを、遠隔地間で効率的に処理しスムーズに流通させるための通信品質保証技術を確立することができたので、本研究の当初の目標は達成できたものとする。

参考資料、参考文献

- [AEKL 2002] Alkis, Evlogimenos, K. H. Lim, and K. Lai, "On the implementation of datagram congestion control protocol,"
<http://www.cs.berkeley.edu/~laik/projects/dccp/AEKL02.DCCP.ps>, 2002.
- [Barnard 1980] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," IEEE Trans. PAMI, Vol. 2, No. 4, pp. 333-340, 1980.
- [Floyd 2000] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-based congestion control for unicast applications," ACM SIGCOMM, 2000.
- [Funkhouser 1993] T. A. Funkhouser and C. H. Sequin, "Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments," In Proc. CG93, pp. 247-254, Vol. 27, 1993.
- [Hartley 1992] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," Proc. ECCV, pp. 563-578, 1992.
- [Hirose 2001] K. Tamagawa, T. Yamada, T. Ogi and M. Hirose, "Developing a 2.5D video avatar," IEEE Signal Processing Magazine, pp. 35-42, Vol. 18, No. 3, 2001.
- [Kawanishi 1998] T. Kawanishi, K. Yamazawa, H. Iwasa, H. Takemura, and N. Yokoya, "Generation of high-resolution stereo panoramic images by omnidirectional imaging sensor using hexagonal pyramidal mirrors," Proc. IAPR, International Conference on Pattern Recognition, I, pp. 485-489, 1998.
- [Kim 2003] J. R. Kim, Y. Won and B. Kim, "Hierarchical QoS architecture for networked virtual dancing environment", Proc. ICAT2003, pp.46-53, 2003.

- [Majumder 1999] A. Majumder, G. Meenakshisundaram, W. B. Seales, and H. Fuchs, "Immersive teleconferencing: A new algorithm to generate seamless panoramic video imagery," ACM Proc. International Conference on Multimedia, 1999.
- [McManus 2002] P. R. McManus, "DCCP Linux Kernel Patches," <http://www.ducksong.com:81/dccp/>, May, 2002.
- [Moriya 2002] T. Moriya, K. Utsugi, F. Beniyama, and H. Takeda, "Method to modify images for IPD environment using pseudo-projection models," Proc. 8th International Conference on Virtual Systems and Multimedia, 2002.
- [Murotani 2002] K. Murotani and K. Sugihara, "G1 surface interpolation for irregularly located data," Geometric Modeling and Processing, pp. 187-196, 2002.
- [Nalwa 1996] V. Nalwa, "A true omnidirectional viewer," Bell Laboratories Technical Report, 1996.
- [Teler 2001] E. Teler and D. Lischinski, "Streaming of complex 3D scenes for remote walkthroughs," Proc. EG2001, Vol. 20(3), 2001.
- [Tsai 1984] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," IEEE Trans. PAMI, Vol. 6, 1984.
- [Tsai 1987] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the shelf TV cameras and lenses," IEEE Journal. RA, Vol. 3, No. 4, 1987.
- [Utsugi 2003] K. Utsugi, T. Moriya, N. Nonaka and H. Takada, "LOD of video avatar for walkthrough applications," ACM Proc. SIGGRAPH2003 Sketches and Applications, 2003.
- [Wolfe 1991] W. J. Wolfe, D. Mathis, C. W. Sklair, and M. Magee, "The perspective view of three points," IEEE Trans. PAMI, Vol. 13, No. 1, 1991.
- [Yamasaki 2002] M. Yamasaki, T. Minakawa, H. Takeda, S. Hasegawa, and M. Sato, "Technology for seamless multi-projection onto a hybrid screen

- composed of differently shaped surface elements," Proc. 7th Annual Immersive Projection Technology Symposium, 2002.
- [Zhang 1998] Z. Zhang, "A flexible new technique for camera calibration," Technical Report MSR-TR-98-71, Microsoft Research, 1998.
- [Zhang 2000] Z. Zhang, "A flexible new technique for camera calibration," IEEE Trans. PAMI, Vol. 22, No. 11, pp. 1330-1334, 2000.
- [Zhang 2002] Z. Zhang, "A flexible new technique for camera calibration," Microsoft Research Technical Report, MSR-TR-98-71, 1998, updated on 2002.
- [植芝 2002] 植芝俊夫, 富田文明, "平面パターンを用いた複数カメラシステムのキャリブレーション," 情報処理学会 CVIM 研究会, 2002-CVIM-135, pp. 47-54, 2002.
- [圓川 1996] 圓川隆夫, 伊藤謙治, シリーズ[現代人の数理] 10 生産マネジメントの手法, 朝倉書店, 1996.
- [杉原 1995] 杉原厚吉, グラフィックスの数理, 共立出版, 1995.
- [高木 1991] 高木幹雄, 下田陽久, 画像解析ハンドブック, 東京大学出版会, 1991.
- [続 2001] 続元宏, 岩田洋夫, "巡回式高解像度実画像ディスプレイ," 日本バーチャルリアリティ学会第6回大会論文集, 2001.
- [出口 2002] 出口光一郎, 岡谷貴之, "3次元環境理解のための多視点カメラのキャリブレーション," コンピュータビジョンとイメージメディア, CVIM, No. 131, pp. 1-8, 2002.
- [根岸 2003] 根岸善朗, 三浦純, 白井良明, "全方位ステレオとレーザレンジファインダの統合による移動ロボットの地図生成", 日本ロボット学会誌, Vol. 21, No. 6, pp. 110-116, 2003.
- [守屋 2003] 守屋俊夫, 武田晴夫, 新井民夫, "投影点観測による複数光線の照射方向推定方法," 電子情報通信学会論文誌, Vol. J86-D-II, No. 4, pp. 479-490, 2003.
- [八木 2001] 八木康史, 横矢直和, "全方位ビジョン: センサ開発と応用の最新動

- 向," 情報処理学会論文誌: コンピュータビジョンとイメージメディア, Vol. 42, No. SIG 13 (CVIM 3), pp. 1-18, 2001.
- [横矢 2000] 横矢直和, "3Dパノラマ画像合成技術," 映像情報メディア学会誌, Vol. 54, No. 3, pp. 338-342, 2000.
- [CCID3] S. Floyd et al., "Profile for DCCP Congestion Control ID 3:TFRC Congestion Control,"
<http://www.ietf.org/internet-drafts/draft-ietf-dccp-ccid3-04.txt>, 2003.
- [CCID3THIN] Eddie Kohler, "DCCP CCID 3-Thin,"
<http://www.ietf.org/internet-drafts/draft-ietf-dccp-ccid3-thin-00.txt>, 2003.
- [DCCP WG] The Internet Engineering Task Force - Datagram Congestion Control Protocol, <http://www.ietf.org/html.charters/dccp-charter.html>, 2003.
- [DCCPSPEC] E. Kohler et al., "Datagram Congestion Control Protocol (DCCP),"
<http://www.ietf.org/internet-drafts/draft-ietf-dccp-spec-05.txt>, 2003.
- [SharedAPI] Junwen Lai, "[dccp] shared buffer DCCP API,"
<http://www1.ietf.org/mail-archive/working-groups/dccp/current/msg00339.html>, 2003.

(添付資料)

1 研究発表、講演、文献等一覧

論文

(1) 発表済み

- [1] C. Trevai, T. Moriya, R. Ueda, and T. Arai, "Integration of uniform Monte-Carlo localization method for mobile robot with sonar array," Proc. IAS-8, The 8th Conference on Intelligent Autonomous Systems, 2004. (査読有)
- [2] R. Ueda, C. Trevai, T. Moriya, and T. Arai, "Mobile robot control for composition of seamless and high-resolution images in library," IS&T/SPIE Annual Symposium Electronic Imaging Science and Technology, 2004. (査読有)
- [3] 小林美保、宇都木契, "高臨場感ディスプレイ環境を用いた遠隔学習システム," 情報処理学会第66回全国大会, 2004. (査読無)
- [4] 宇都木契, 野中尚道, 小林美保, "多段階ネットワーク制御のアバタ通信への応用," 電子情報通信学会2004年総合大会, 2004. (査読無)
- [5] 野中尚道, 宇都木契, 草間一宏, "映像配信アプリケーションと協調したネットワーク品質制御方式の検討," 電子情報通信学会3月研究会, 2004. (査読無)
- [6] T. Moriya, F. Beniyama, K. Matsumoto, T. Minakawa, K. Utsugi, and H. Takeda, "Properties of three shadows on a plane," WSCG2004, Proc. The 12-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, 2004. (査読有)
- [7] T. Moriya, F. Beniyama, K. Utsugi, T. Minakawa, H. Takeda, and K. Ando, "Multi-camera and multi-projector based seamless live image display system," MMM2004, Proc. The 10-th International Multi-Media Modeling Conference, 2004. (査読有)
- [8] T. Moriya and H. Takeda, "Method to generate images for a motion-base in an immersive display environment," IEICE The Institute of Electronics, Information and Communication Engineers Transactions on Information and Systems, Vol. E86-D, No. 10, pp. 2231-2239, 2003. (査読有)
- [9] T. Moriya, F. Beniyama, K. Utsugi, and H. Takeda, "Image modification for immersive projection display based on pseudo-projection models," The Journal of

- Visualization & Computer Animation, Vol. 14, Issue 4, pp. 197-210, 2003. (査読有)
- [10] K. Utsugi, T. Moriya, N. Nonaka, and H. Takeda, "LOD of video avatar for workthrough applications," ACM SIGGRAPH, Sketches and Applications, 2003. (査読有)
- [11] 守屋俊夫, 紅山史子, 松本高斉, "マルチカメラの視点位置キャリブレーション方法," View2003, ビジョン技術の実利用ワークショップ, 2003. (査読無)
- [12] H. Namai, T. Moriya, H. Takeda, "Interactive educational content for immersive projection environment," SCI 2003, The 7th World Multi-conference on Systemics, Cybernetics and Informatics, 2003. (査読有) **[Best Paper Award 受賞]**
- [13] F. Beniyama and T. Moriya, "Creation of omnidirectional image eliminating moving object," SCI 2003, The 7th World Multi-conference on Systemics, Cybernetics and Informatics, 2003. (査読有)
- [14] C. Trivai, R. Ueda, T. Moriya, and T. Arai, "Mobile robot system for composition of seamless and high resolution images --Mobile robot localization and map building," IEEE SMC, International Conference on Systems, Man & Cybernetics, 2003. (査読有)
- [15] 守屋俊夫, 武田晴夫, 新井民夫, "投影点観測による複数光線の照射方向推定方法," 電子情報通信学会論文誌, Vol. J86-D-II, No. 4, pp. 479-490, 2003. (査読有)
- [16] 紅山史子, 守屋俊夫, 宇都木契, "ミラー付きマルチカメラシステムにおける画像接続誤差の低減," FIT2003 (情報科学技術フォーラム), 2003. (査読無)
- [17] 生井 仁, 及川 道雄, 守屋俊夫, 宇都木契, "高臨場感ディスプレイ環境における教育向け遠隔ガイドシステム," FIT2003 (情報科学技術フォーラム), 2003. (査読無)
- [18] 松本高斉, 石原利昭, 守屋俊夫, "局所画像特徴群の多重評価にもとづくロバストな物体認識," 第4回 S I C E システムインテグレーション部門講演会 (SI2003), 2003. (査読無)
- [19] 上田隆一, 守屋俊夫, トリワイ チョンチャナ, 新井民夫, "棚板上にある物体のモザイク映像作成のための移動ロボット制御," 第4回 S I C E システムインテグレーション部門講演会 (SI2003), 2003. (査読無)
- [20] 紅山史子, 守屋俊夫, 宇都木契, "ミラー式マルチカメラシステムにおける隣り合う画像間のエッジ除去," 2003年度 (社)映像情報メディア学会 年次大会, 2003. (査読無)

- [21] 小池崇文, 小林美保, 及川道雄, "複数台PC構成によるマルチプロジェクションシステム用Webブラウザの開発," 映像情報メディア学会2003年年次大会, 2003. (査読無)
- [22] K. Matsumoto, "Multiple Camera Image Interface for Operating Mobile Robot," 12th IEEE Workshop Robot and Human Interactive Communication RO-MAN2003, 2003. (査読有)
- [23] 小池崇文, 及川道雄, 山崎眞見, "複数台PCを用いたマルチプロジェクションシステム用動画表示ソフトウェアの開発," FIT2003 (情報科学技術フォーラム), 2003. (査読無)
- [24] M. Kobayashi, K. Utsugi, M. Yamasaki, and H Takeda, "Distance Learning about Chang'an in an Immersive Environment," 13th International Conference on Artificial Reality and Telexistance (ICAT2003), 2003. (査読有)

(2) 投稿中

- [25] T. Minakawa and M. Yamasaki, "Elliptic vs. rectangular blending for multi-projection displays," IEICE Trans. on Information and Systems.
- [26] 宇都木契, 小林美保, 坂井秀行, 山崎眞見, 武田晴夫, "異種サイバースペースのシームレス接続," バーチャルリアリティ学会論文誌.
- [27] 皆川剛, 山崎眞見, 宇都木契, 守屋俊夫, 武田晴夫, "An Immersive Environment with a Motion Base using Multi-Detailed Projection Technique," IEICE TRANSACTIONS on Information and Systems.
- [28] 石原利昭, 松本高斉, 守屋俊夫, 紅山史子, "ミラー式マルチカメラシステムにおける画像接続時の欠損領域補完方法," 映像情報メディア学会年次大会, 2004.

(3) その他の発表

- [29] "ブロードバンド環境に対応した、実写とCG映像をリアルタイムに融合する技術を開発," ニュースリリース, 2003.
- [30] "広域シームレス画像の自動撮影技術を開発," 日刊工業新聞記事, 2003.
- [31] "実写とCGがさらにリアルに融合するビデオアパタの新技术," CGWORLD, 2003.
- [32] "Distance Learning about Chang'an in an Immersive Environment," 日本経済新聞取材, 2004.