

平成16年度  
研究開発成果報告書

全方位高解像リアルタイム動画入力と  
その配信システムに関する研究開発

委託先：(株)映蔵

平成17年5月

情報通信研究機構

# 平成16年度 研究開発成果報告書

「全方位高解像度リアルタイム動画像入力とその配信システムに関する研究開発」

## 目 次

1	研究開発課題の背景	2
2	研究開発の全体計画	2
2-1	研究開発課題の概要	2
2-2	研究開発目標	7
2-2-1	最終目標	7
2-2-2	中間目標	7
2-3	研究開発の年度別計画	8
3	研究開発体制	9
3-1	研究開発実施体制	9
4	研究開発実施状況	
4-1	複合センサカメラの開発	10
4-1-1	はじめに	10
4-1-2	複合センサカメラの試作	10
4-1-3	複合センサカメラのキャリブレーション	11
4-1-4	評価実験	14
4-2	リアルタイム高解像度動画像の作成	17
4-2-1	周波数空間アプローチによる高解像度画像の作成	17
4-2-2	モーフィングアプローチによる高解像度画像の作成	25
4-3	高解像度全方位カメラ用光学系の設計	31
4-3-1	双曲面型全方位視覚センサ	32
4-3-2	二枚反射全方位視覚センサ	33
4-4	全方位高解像度リアルタイム動画像入力記録方式の開発	36
4-4-1	複合センサカメラシステムを利用したリアルタイム動画像記録	36
4-4-2	書き込み速度の最適化	37
4-4-3	パルスジェネレータの作成	37
4-5	高解像度動画配信用ソフトウェアの開発	38
4-5-1	視点情報に基づく動画像配信	38
4-5-2	MPEG4 規格の調査	39
4-5-3	受信クライアントの構築	42
4-5-4	配信サーバーの構築	44
4-5-5	システム評価実験	47
4-6	ライブ入力, 配信システムの開発	48
4-7	総括	49
5	参考資料・参考文献	
5-1	研究発表・講演等一覧	



を作成する技術を開発する。これにより、遠隔からのモニタリング、セキュリティ、遠隔会議、遠隔医療、遠隔教育、コンサートやスポーツなどのインタラクティブ放送などへの応用が実現可能となる。図1に全体システムのイメージ図を示す。

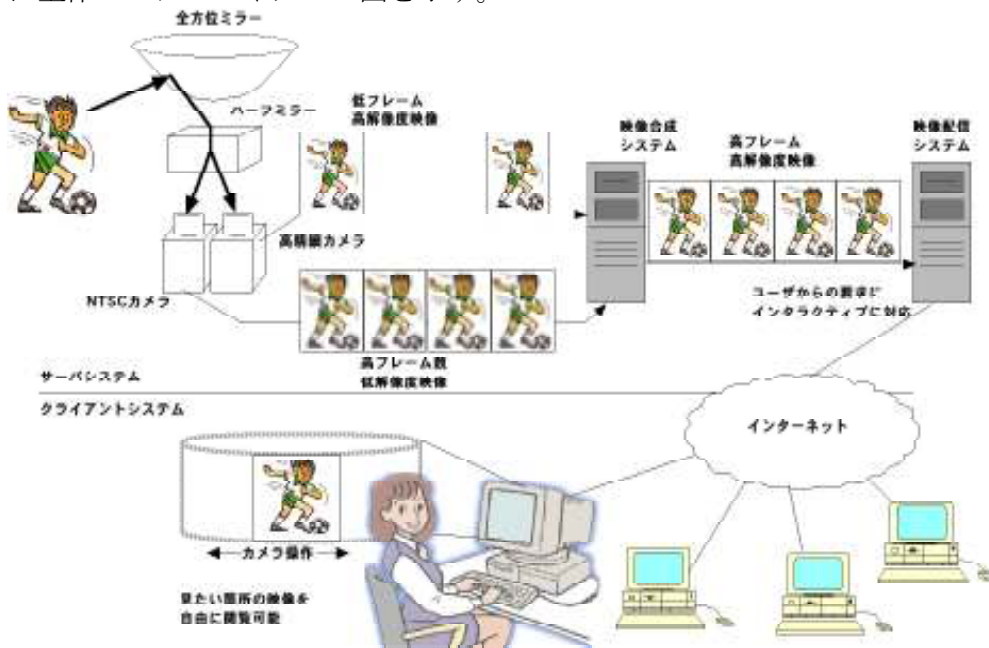


図1 全体システムのイメージ図

### (1) 複合センサカメラの試作：

従来、動画入力のできるビデオカメラとしては、NTSCクラス（640x480画素、30フレーム/秒）のカメラが長年使われてきた。しかし、NTSCクラスの画像サイズ（640x480画素）では、広視野で画像を撮像すると解像度が不足するという欠点があった。一方、近年のデジタルカメラの急速な発展と高機能化により、高解像度のカメラが次々と現れ、現状で4000x4000画素を持つ高解像度のカメラも市販されている。しかし、高解像度になると、フレームレートは低下し、リアルタイムで撮像することはできないという欠点がある。たとえば、4000x4000画素のセンサの場合、1秒間に1フレームの速度である。実時間入力（30フレーム/秒）できるものはNTSC（640x480画素）クラスの画像サイズがほとんどである。本研究では、焦点（レンズ中心）を合わせることにより、同一の視野を持った2種類の動画像；a)高解像度（4000 x 4000画素）だが時間的には粗い動画像（1フレーム/秒）と、b)通常の解像度（640x480画素）だが時間的に密な動画像（30フレーム/秒）から、空間的にも高解像度で、しかも時間的にも密な動画像（4000x4000画素、30フレーム/秒）を作成する技術を開発する。このように、高解像度の実時間全方位動画像を撮影できるカメラは他に例がなく本研究独自のものである。まず、その第一歩として、図2に示すように、焦点（レンズ中心）を合わせることにより、同一の視野の動画像を2種類の異なった解像度で撮影し蓄積するカメラを試作する。

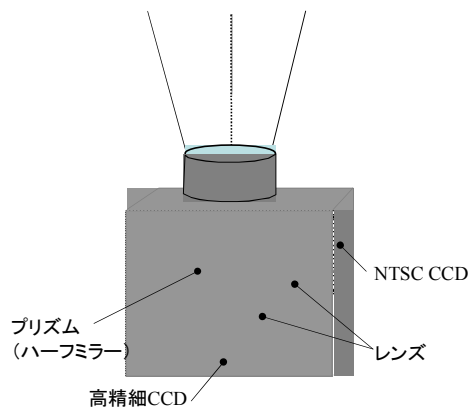


図2. 複合センサカメラ

## (2) リアルタイム高解像度動画の作成 :

上で試作した複合センサカメラから得られる2つの動画像から、空間的にも高解像度で、しかも時間的にも密な動画像(4000x4000画素、30フレーム/秒)を作成する技術を確認する。すでに、理論的な解析は終わっているので、そのソフトウェアを試作し、実際に、高解像度だが時間的には粗い動画像と、通常の解像度だが時間的に密な動画像の2つの動画像から空間的にも時間的にも密な動画像を作成できることを実験的に確かめる。

高解像度動画像の作成には、2台のセンサからの動画像を2つの3次元時空間データとして扱い、3次元時空間で統合することにより高解像度化を図る。具体的な時空間高解像度処理ブロックを図3に示す。低解像度用カメラと高解像度用カメラの2つの映像信号を、空間オーバーサンプリング、ディレイ、高解像度化処理を行うことにより時空間周波数の広帯域化を行う。図4に示すように、低解像度カメラは空間周波数は低いが、フレームレートは高いので時間周波数は高いという特徴がある。また、高解像度カメラは空間周波数は高いがフレームレートは低いので時間周波数は低いという特徴がある。高帯域化を行うには、図4における右上領域まで有効信号成分を含ませることである。通常、これらの領域には信号の折り返し成分(エイリアス成分、ノイズ)が含まれているので、オーバーサンプリングによりエイリアス成分をより高周波に移動させ、有効信号成分を含むことができるようにする。この処理は、図3における空間オーバーサンプリングとディレイ(時間方向のオーバーサンプリング)で実現する。図3における高解像度化処理は、図5のような周波数成分を持つ映像信号を混成する処理となる。左上と右下の周波数領域は、それぞれのカメラの信号成分を利用する。左下の信号成分は、低解像度カメラと高解像度カメラの2つの有効信号成分を本来含んでいるので、この信号にノイズ低減処理を施し、高品質映像信号を生成する。右上の領域はオーバーサンプリングによる疑似信号成分であるが、この部分にもノイズ低減処理を行うことで、最終的に時空間高周波映像を生成する。

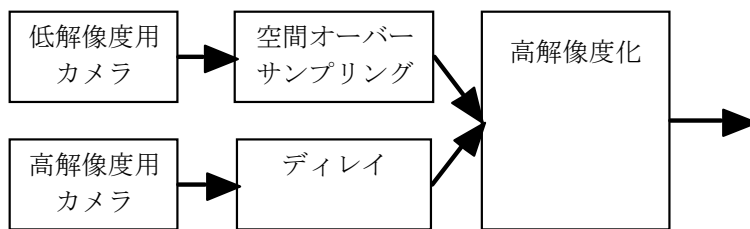


図3 時空間高解像度化処理

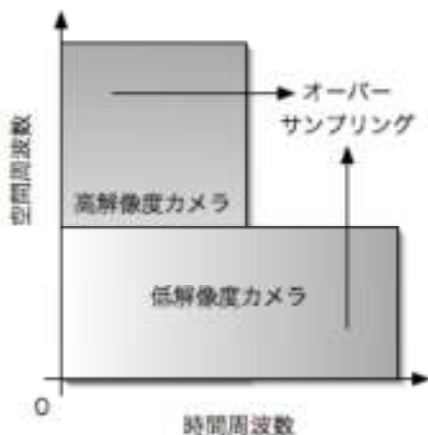


図4 各カメラの周波数帯域

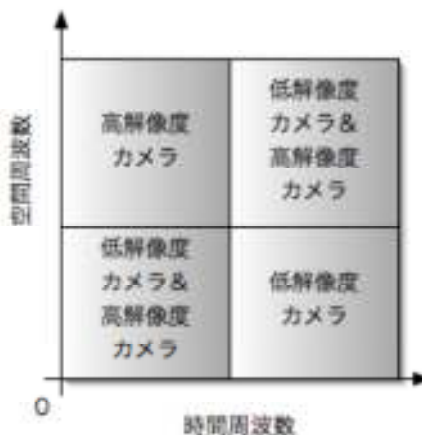


図5 時空間高解像度化後の周波数帯域

## (3) 高解像度全方位カメラ用光学系の設計と全方位高解像度リアルタイム動画入力記録方式の開発 :

上で作成した複合センサカメラ用に全方位ミラーを設計し、全方位複合センサカメラを試作する。また、このカメラから得られる2つの全方位動画像から空間的にも高解像度で、しかも時間的にも密な全方位動画像(4000x4000画素、30フレーム/秒)を作成する技術を開発する。研究担当者が世界に先駆け開発したリアルタイム全方位カメラを用いると、周囲360°のシームレスな動画像がリアルタイムで撮影でき、臨場感の高い映像を得ることができる。また、この全方位画像入力に研究担当者らの提案した双曲面ミラーを用いると、双曲面は焦点を持つことから、パノラマ画像や

透視変換画像に精度良くかつ容易に変換できる。このため、世界的にこれを用いた応用研究がなされ、すでにNTSC (640x480 画素) クラスの画像を対象とした全方位カメラは市販されている。しかし、全方位カメラの欠点は、ある部分に着目したときの解像度が劣るという点である。すなわち、通常のカメラが水平画角 40° 前後であるのに対し、全方位視では 1 台のカメラで周囲 360° を撮影していることから、ある部分に着目したときの空間分解能は 1/9 程度しかないため、通常のカメラではその用途に限られてくる。たとえば、遠隔監視を例にとってみても、人がいることは分かってもその人の顔を判別できるまでの解像度がない。この解像度の低さが実用化の妨げとなっていた。この問題を解決するための高解像度化には高解像度のセンサを用いることが当然考えられる。しかし、すでに述べたように、非常に高価格なものを除いては高解像度になるとフレームレートは低下し、実時間入力できるものはNTSC (640x480 画素) クラスの画像サイズがほとんどである。このため、本研究では、上で開発した複合センサカメラを入力カメラとして用いることにより、全方位の高解像度リアルタイム動画入力記録システムを開発する。

全方位画像の解像度を上げるには、センサの解像度を上げると同時に全方位画像を撮影するための光学系（主にミラー）の解像度も上げる必要がある。しかし、我々の提案した双曲面で代表される単一のミラーを用いた全方位カメラの場合、非点収差、コマ収差、球面収差などの影響を受け、像にぼけが生じることが原因で、解像度を上げるにはミラーを大型化する必要があり、カメラとしての携帯性に問題が生じる。この問題を研究担当者らが新たに考案した二回反射全方位光学系（特許）を用いることで解決する。二回反射光学系の例を図 6-b ~ 6-d に示す。図 6-b は双曲面鏡と平面鏡を用いた構造で、平面鏡により光路を折り曲げることで、図 6-a の一枚反射光学系に対して双曲面鏡とカメラ間の距離を約 1/2 にできるため小型化に有利である。また、図 6-c, d はそれぞれ双曲面-楕円面と放物面-放物面の組み合わせの凹凸面鏡ペアによる光学系である。二枚の曲面を用いることで設計自由度が上がるため、更なる小型化が期待できる。図 6-b ~ d の二回反射光学系は、いずれも図 6-a 同様に単一視点を保持しており、従来の双曲面鏡を用いた全方位センサと同様に、自由視点の画像やパノラマ画像に無ひずみで変換可能である。これら二回反射光学系は、これまでの試作ならびにシミュレーション実験から、単一ミラーを用いた場合の数分の 1 の大きさでほぼ同程度の光学系が実現できている。これにより、実用性という点では重要となる、小型軽量で携帯性のある高解像度の全方位カメラが初めて実現可能となる。

また、今回対象とする CCD センサは、9 $\mu$ m のセルサイズの 4000 $\times$ 4000 画素を想定しているため、従来設計してきたものに比べ、スペックが 1 ランク上となる。また、ミラーが 2 枚になることで設計自由度が上がるので、最適解の計算には、専用のシミュレーションプログラムを作成し、並列計算による高速化が必要である。このシミュレーションは、研究担当者らの所属機関で開発したシミュレータを利用し行う。また試作した全方位カメラの解像度分布などの特性評価もこのシミュレータを用いて行う。

こうして試作された複合双曲面ミラーと複合センサカメラを用いて全方位高解像度リアルタイム動画入力記録カメラを作成する。また、このカメラから得られる 2 つの全方位動画画像から、空間的にも高解像度で、しかも時間的にも密な全方位動画画像 (4000 $\times$ 4000 画素、30 フレーム/秒) を作成する技術を確認する。このように、高解像度の実時間全方位動画画像を撮影できるカメラは世界的に見て他に例がない。なお、開発当初は、現状で入手できる 4000 $\times$ 4000 画素の解像度を持つセンサで開発を行い中間目標とするが、2、3 年後には 8000 $\times$ 8000 画素 (6400 万画素) のセンサが開発されると思われるので、最終年度までには 8000 $\times$ 8000 画素の解像度を持つ全方位のリアルタイム動画入力記録装置を開発する予定である。

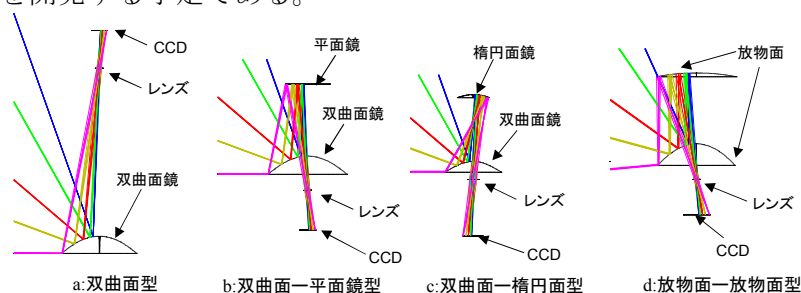


図 6：全方位視覚センサの光学系

#### (4) 高解像度動画配信ソフトウェアの開発：



複数人に同時配信するときに、複数人が複数の部分画像を同時要求した場合の性能評価を行う。通常の映像配信の場合は、高優先度映像だけが送られることになるが、部分画像だけで良い場合は、部分画像だけを転送すれば良いので帯域圧縮が期待できる。そこで、部分画像を要求したクライアントには、低優先度高空間周波数画像データチャンネルを高解像度部分画像データに差し替えて伝送することにより、配信サーバは個人個人の要求にインタラクティブに応答することができる。ここでは、ネットワーク帯域と応答性能、CPU 能力、システムバス能力とを比較して、配信サーバが何台までのクライアントの要求に応えられるかを検証する。

## 2-2 研究開発目標

### 2-2-1 最終目標（平成19年3月末）

- 1) 焦点（レンズ中心）を合わせることにより、同一の視野を持った2種類の全方位動画、a) 高解像度（画像サイズ 8000x8000 画素）だが時間的には粗い（1 フレーム/秒）全方位動画と、b) 通常の解像度（640x480 画素）だが時間的に密な（30 フレーム/秒）全方位動画を同時に撮影し蓄積するカメラを開発する。
- 2) 上の2つの全方位動画から、空間的にも高解像度で、しかも時間的にも密な全方位動画（8000x8000 画素、30 フレーム/秒）を作成する技術を開発する。
- 3) 蓄積された2種類の全方位動画を圧縮してインターネットにより複数の閲覧装置（クライアント）に配信するサーバシステムと配信された2種類の全方位動画から閲覧者側で高解像度（8000x8000 画素）かつリアルタイム（30 フレーム/秒）の動画を作成する技術を開発する。また、閲覧者の要求する視野の部分画像のみを切り出しそれらを能率良く配信し、クライアント側でその部分画像のみを高解像度で動画表示するシステムも実現する。
- 4) 以上のシステムを蓄積された全方位動画に対してだけでなく、撮影しつつ配信するライブ配信についても同様の機能を実現する。
- 5) エジプトのピラミッドや日本の古墳等の高精細なデジタルアーカイブ化とその配信による実証実験を行い、本提案技術の有効性の検証とその周知を行う。

### 2-2-2 中間目標（平成17年1月末）

- 1) 焦点（レンズ中心）を合わせることにより、同一の視野を持った2種類の全方位動画、a) 高解像度（画像サイズ 4000x4000 画素）だが時間的には粗い（1 フレーム/秒）全方位動画と、b) 通常の解像度（640x480 画素）だが時間的に密な（30 フレーム/秒）全方位動画を同時に撮影し蓄積するカメラを開発する。
- 2) 上の2つの全方位動画から、空間的にも高解像度で、しかも時間的にも密な全方位動画（4000x4000 画素、30 フレーム/秒）を作成する技術を確立する。
- 3) 蓄積された2種類の全方位動画を圧縮してインターネットにより複数の閲覧装置（クライアント）に配信するサーバシステムと配信された2種類の全方位動画から閲覧者側で高解像度（4000x4000 画素）でかつリアルタイム（30 フレーム/秒）の動画を作成する技術を開発する。また、閲覧者の要求する視野の部分画像のみを切り出しそれらを能率良く配信し、クライアント側でその部分画像のみを高解像度で動画表示するシステムも実現する。



## 2-3 研究開発の年度別計画

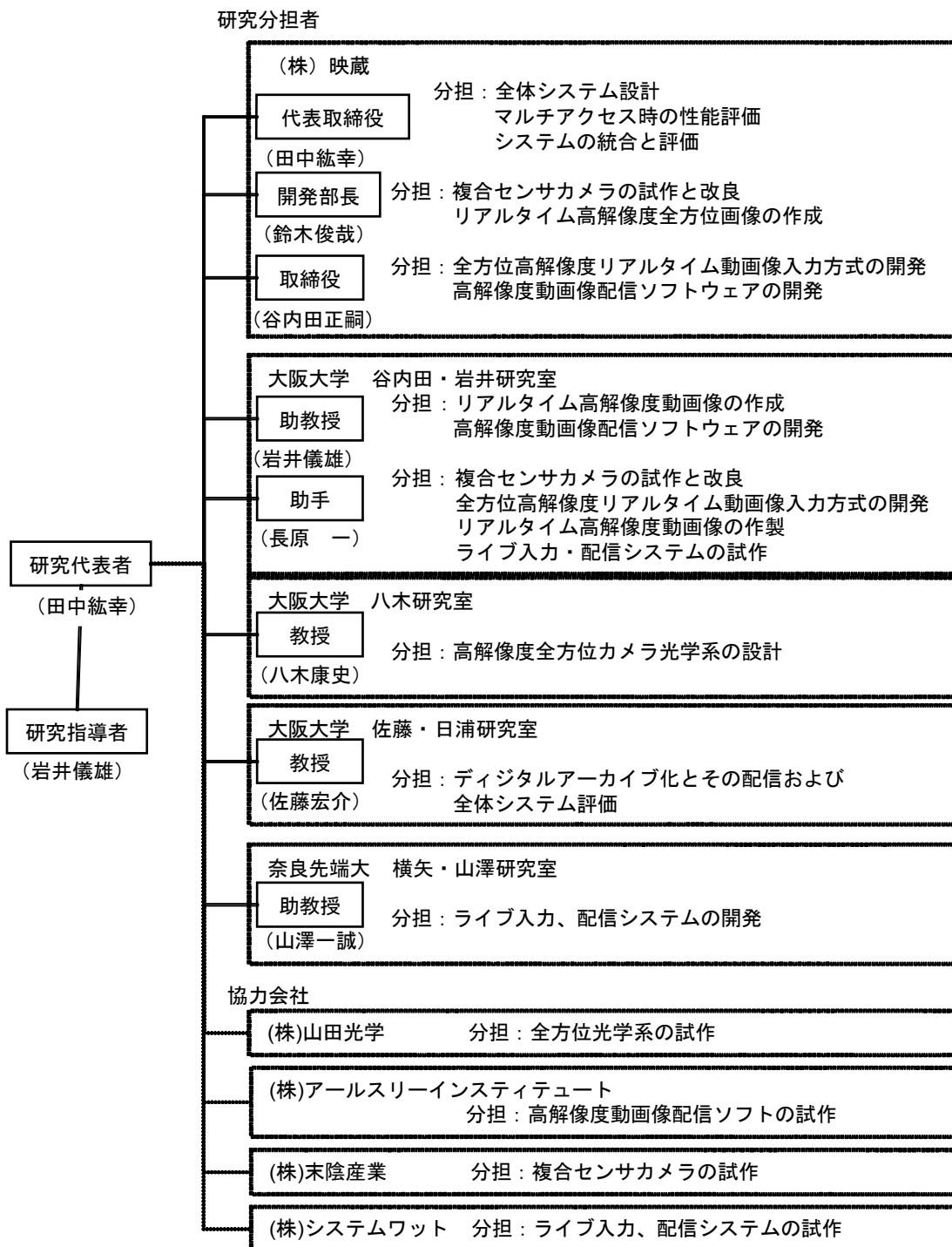
(金額は非公表)

研究開発項目	15年度	16年度	17年度	18年度	年度	計	備考
全方位高解像度動画入力とその配信システムに関する研究開発							
1. 複合センサカメラの試作 ・複合センサ設計・試作, センサ特性評価		→					
2. リアルタイム高解像度動画像の作成 ・高解像度動画像生成アルゴリズムの開発 ・シミュレーション評価		→					
3. 高解像度全方位カメラ用光学系の設計 ・光学系設計・試作, センサ特性評価			→	→			
4. 全方位高解像度リアルタイム動画像入力記録方式の開発 ・動画入力フォーマットの設計・試作			→	→			
5. 高解像度動画配信用ソフトウェアの開発 ・動画配信アルゴリズムの開発 ・シミュレーション評価			→				
6. ライブ入力, 配信システムの開発 ・ライブ配信システムの試作				→			
7. マルチアクセス時の性能評価 ・マルチアクセスプロトコル開発				→			
8. 総合システムの試作と評価 ・データ収集, 総合評価				→			
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 年度の欄は研究開発期間の当初年度から記載。

### 3 研究開発体制

#### 3-1 研究開発実施体制



## 4 研究開発実施状況

### 4-1 複合センサカメラの開発

#### 4-1-1 はじめに

カメラ技術の発達により様々なカメラが市場に出回っている。また、それに伴って高品質な映像に対する世の中のニーズも高まっている。高品質な映像の条件としては高臨場感を与える高解像度や、スムーズな動画再生を実現する高フレームレートが挙げられる。そのような高品質な映像を実現するために多くの研究が行われてきた。その例として、日本放送協会(NHK)が開発した超高精細カメラが挙げられる。暫定的な仕様ではあるがこのカメラは走査線 4000 本、フレームレートが 60fps とかなり高品質な映像を撮像できる。また、映画の撮影から上映までをデジタルで行うデジタルシネマが提案されている。これはフィルムに匹敵する品質の高精細映像を電子的手法により撮影、編集、保存することができ、またデジタルであるため画質の劣化がない。現在、4K フォーマットと呼ばれる  $4096 \times 2160$  pixel, 24fps の超高精細の映像規格が裁定され、これに対応する機器の開発や、配信実験が行われている。しかし、これらのシステムは、高価な機材を利用することで高品質な映像を実現しており、一般的な利用に結びつけるのはコストの面からも困難である。一般的なテレビ放送映像方式である NTSC 方式では、フレームレートはスムーズな動画を表現できる 30fps であるが、解像度は  $640 \times 480$  pixel で、高解像度であるとはいえない。

一方で、デジタルスチルカメラには  $4000 \times 4000$  pixel という高解像度のものが存在するが、フレームレートは低く滑らかな動画を撮影するには無理がある。このように、単位時間あたりの画像データレートは CCD (Charge Coupled Device) の画素数とフレームレートの積で表され、カメラの画像掃き出し速度の制限のために解像度とフレームレートはトレードオフの関係にある。つまり、一般的なカメラを用いての高解像度と高フレームレートを両立した撮影を行うのは困難である。

そこで、本研究では高解像度と高フレームレートを両立した撮影を実現するために特性の異なる 2 台のカメラを複合して用いることを提案する。この複合センサカメラでは解像度を重視した高解像度低フレームレートのカメラと、フレームレートを重視した低解像度高フレームレートのカメラを複合して用いる。シーンからの入射光をハーフミラーで分光し、時空間周波数の異なる 2 台のカメラでそれぞれ撮像することにより同一視点、同一視野で高解像度と高フレームレートの撮像を行うことができるカメラシステムを構築した。さらに、2 台のカメラの特性が各々異なるため、レンズ歪み等の幾何学的特性を一致させる幾何学的キャリブレーションと、画素値の光学的特性を一致させる光学的キャリブレーションを行う必要がある。このような 2 枚の画像を一致させるためのキャリブレーション手法を提案する。

#### 4-1-2 複合センサカメラの試作

本システムは複合センサカメラと、画像を取り込むための PC、画像記録用の RAID (Redundant Arrays of Inexpensive Disks) システム、画像表示用のディスプレイ、フレームレートの異なる 2 台のカメラを同期させて撮像するためのパルスジェネレータから構成される (図 8)。複合センサカメラで撮像シーンの入射光をハーフミラーを用いて分光して透過光を一方のカメラで、反射光をもう一方のカメラで撮像する。このようにして得られた画像を PC 上に取り込んで ディスプレイに表示させつつ、RAID システムに保存していく。

ここで、実際のシステムを図 9 に示し、試作した複合センサカメラで撮像される 2 種類の画像の仕様を表 1 に示す。また、本システムで取り込んだ画像を図 11, 12 に示す。

図 10 に示すように、複合センサカメラは高解像度低フレームレートカメラと低解像度高フレームレートカメラの 2 台とシーン情報を分光するためのハーフミラーから構成される。2 台のカメラは 3 軸ステージに固定されており位置の調整ができる。2 台のカメラ間で視点が一致するように、3 軸ステージで位置を調整して配置する。また、ハーフミラーの反射光と透過光の比は 4:6 である。高解像度低フレームレートカメラの方が感度が低いために、光量の多い透過光を高解像度低フレームレートカメラで、反射光を低解像度高フレームレートカメラで撮像する。

なお、低解像度高フレームレートカメラとして BASLER 社の A201bc を、高解像度低フレームレートカメラとして REDLAKE 社の ES11000 を用いる。

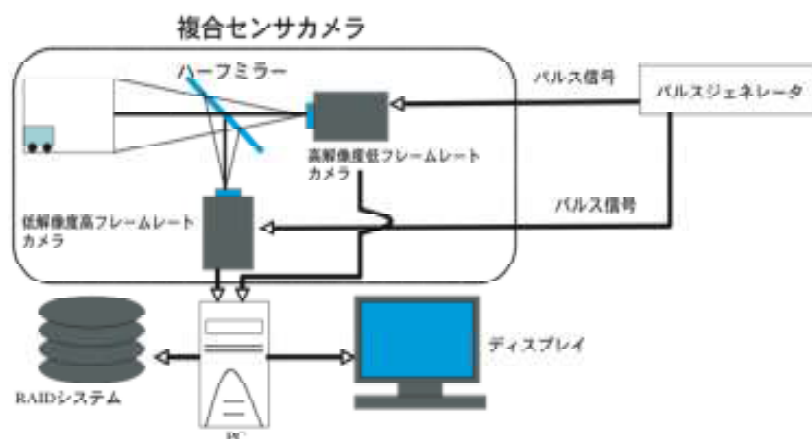


図8 複合センサカメラシステムの構成

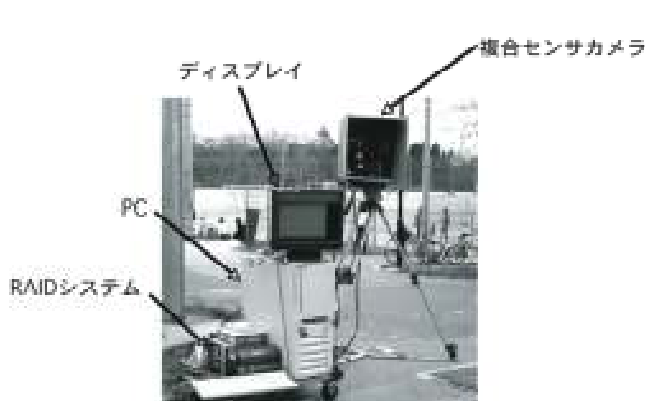


図9 試作した複合センサカメラシステム

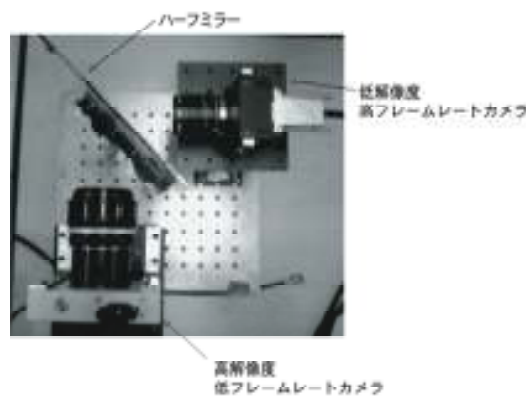


図10 複合センサカメラ

表1 複合センサカメラの仕様

	高解像度 低フレームレート	低解像度 高フレームレート
解像度	4008×2672 pixels	1008×1018 pixels
フレームレート	4.29 fps	30 fps
出力画像	Bayer Color	Bayer Color
出力フォーマット	8 bits/pixel	8 bits/pixel

#### 4-1-3 複合センサカメラのキャリブレーション

複合センサカメラには2つのカメラ間での CCD 特性やレンズ特性の違いにより画像に幾何学的、光学的な差が生じる。複合センサカメラにより得られる2つの画像列情報を統合するためには、キーフレームでの画像一致を前提としている。そのため、幾何学的特性および光学的特性の差を2つのカメラ間で補正する必要がある。ここではこのような複合センサカメラを対象としたキャリブレーション手法を提案する。

##### (1) カメラモデル

本研究では最も単純なピンホールカメラモデルを使用する。図13に示すように基準となる世界座標  $O_w - x_w, y_w, z_w$  における点 P の座標を  $(x_w, y_w, z_w)$  と表し、カメラ座標  $O - xyz$  における点 P の座標を  $(x, y, z)$  と表す。ただし、0 はレンズ中心を表し、z 軸はレンズの光軸に一致するように設定する。



図 11 高解像度低フレームレート画像 (4008×2672 pixels, 4.29 fps)



図 12 低解像度高フレームレート画像 (1008×1018 pixels, 30 fps)

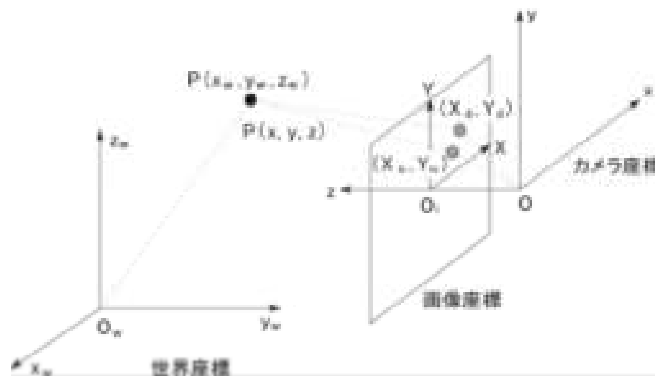


図 13 世界座標と画像座標の関係

世界座標の点 $(x_w, y_w, z_w)$ は回転行列 $\mathbf{R}$ と平行移動ベクトル $\mathbf{T}$ を用いてカメラ座標の点 $(x, y, z)$ へ変換され、さらに透視投影によって焦点距離 $f$ の画像面上の点 $(X_u, Y_u)$ に投影される。

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \mathbf{T}, \mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (1)$$

$$X_u = f \frac{x}{z}, Y_u = f \frac{y}{z}$$

画面上の $(X_u, Y_u)$ はレンズ歪みにより $(X_d, Y_d)$ に移り、さらに、撮像素子によってピクセル単位の画像座標 $(X_f, Y_f)$ に変換される。ここで $\delta(\cdot)$ はレンズ歪みを表す関数である。

$$X_u = X_d + X_d \delta(R_d) \tag{2}$$

$$Y_u = Y_d + Y_d \delta(R_d) \tag{3}$$

$$R_d = \sqrt{X_d^2 + Y_d^2} \tag{4}$$

$$X_f = d_x^{-1} X_d + C_x \tag{5}$$

$$Y_f = d_y^{-1} Y_d + C_y \tag{6}$$

ここで、 $d_x, d_y$ は受光素子の水平・垂直方向の中心間距離、 $(C_x, C_y)$ は画像中心の画像座標である。本研究では式(2)における歪みのモデルとして radial distortion のみを考える。

$$\delta(R_d) = \kappa_1 R_d^2 + \kappa_2 R_d^4 + \dots \tag{7}$$

実際には radial distortion は第1項のみでも十分であるため、

$$X_u = X_d + X_d \kappa_1 R_d^2 \tag{8}$$

$$Y_u = Y_d + Y_d \kappa_1 R_d^2 \tag{9}$$

とする。

## (2) 処理の流れ

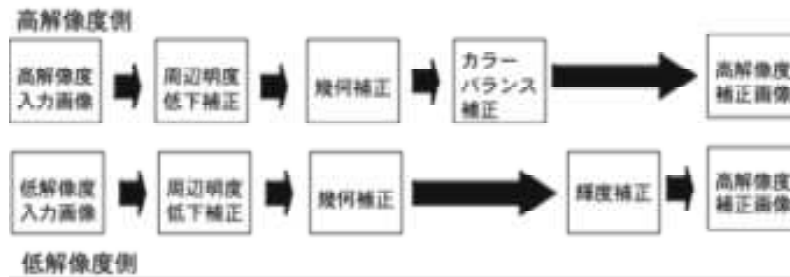


図 14 処理の流れ

複合センサカメラを対象としたキャリブレーション処理の流れを図 14 に示す。高解像度カメラ、低解像度カメラにおいてそれぞれ様な明度を持つ白色ターゲットを撮影し、周辺明度低下の補正を行う。さらにキャリブレーションパターンを撮影することで幾何学的キャリブレーションを行い、画像の幾何特性の違いを補正する。高解像度画像に対して Gray World Assumption に基づくアルゴリズムを用いてカラーバランスの補正を行う。最後に高解像度、低解像度の 2 枚の画像の輝度ヒストグラムが一致するように低解像度側に対して輝度補正を行う。

幾何補正については幾何学的キャリブレーションで、周辺明度低下補正、輝度補正については光学的キャリブレーションで詳しく述べる。

## (3) 幾何学的キャリブレーション

高解像度カメラと低解像度カメラそれぞれに(1)節のカメラモデルを想定し、パラメータ推定を行う。パラメータ推定法として Tsai の手法を用いた。Tsai の手法では、多数の点の世界座標 $(x_w, y_w, z_w)$ とそれらに対応する画像座標 $(X_f, Y_f)$ の組が与えられたときに、内部パラメータとして $f, \kappa_1, (C_x, C_y)$ 、外部パラメータとして $\mathbf{R}$ と $\mathbf{T}$ を求めることができる。ただし、実験の容易さから平面のキャリブレーションパターンを使用する。

複合センサカメラの高解像カメラと度低解像度カメラの 2 つの各画像空間と、世界座標の関係およびカメラパラメータの関係を図 15 に示す。ここで、 $x^{high}, x^{low}$  はそれぞれ高解像度カメラと低解像度カメラの各々カメラパラメータを示す。

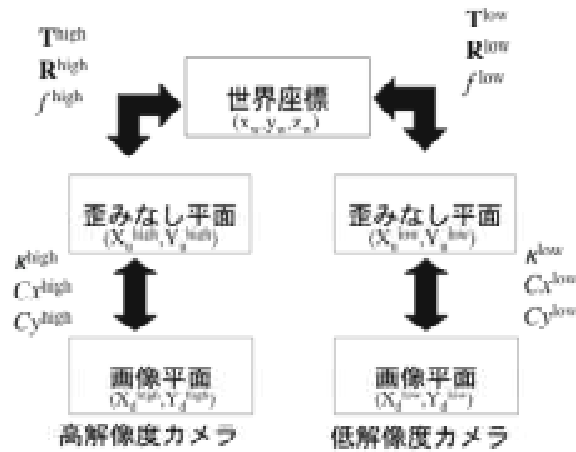


図 15 パラメータと座標の関係

高解像度カメラは画像座標  $(X_f^{high}, Y_f^{high})$  が与えられると、高解像度カメラパラメータにより高解像度歪みなし平面  $(X_u^{high}, Y_u^{high})$  に画像を変換できる。低解像度カメラも同様に画像座標  $(X_f^{low}, Y_f^{low})$  が与えられると、低解像度カメラパラメータと高解像度カメラパラメータを用いて高解像度歪みなし平面  $(X_u^{high}, Y_u^{high})$  に変換できる。つまり、高解像度歪みなし平面上に 2 枚の画像を投影することで幾何学的特性の異なる画像の一致をはかる。以後この高解像度歪みなし平面を基準平面と呼ぶ。

#### (4) 光学的キャリブレーション

##### a. 周辺明度低下の補正

レンズを用いた撮像系では、輝度が一様なシーンを撮影しても画像中心から離れるに従って明度が低下することが知られている。2 台のカメラでこの明度低下の度合いが異なるため、2 枚の画像の輝度差を正確に補正するためには、事前に各カメラで周辺明度低下の補正を行う必要がある。画像周辺の明度低下の原因としては  $\cos^4 \theta$  現象と口径蝕現象がある。

ここでは、 $\cos^4 \theta$  現象や口径蝕現象による明度低下に加え、その他の要因による明度低下や画素間の感度のバラツキによる固定パターンノイズも考慮に入れるため、そのようなモデルをもたずに単純な補正法を利用して明度低下を補正する。具体的には、放射輝度一様の白色シーンを撮影し、画像全体で一様な輝度を持つと仮定し補正マップを作成することで周辺明度低下を補正する。

##### b. 画像間の輝度差の補正

周辺明度低下の補正後、画像間で異なる濃淡や色調を一致させるため、一方の画像に対して輝度変換を行う。2 枚の画像間の輝度変換を行う際には、対応する点の輝度値を比較して最小自乗法により変換式を求めたり、変換テーブルを作成するのが一般的である。ここでは、そのような位置の制約を用いず、任意の非線形な変換に対応するために、2 枚の画像の輝度ヒストグラムが一致するように変換を行う。

ヒストグラムの変換には Yang らの手法を用いた。2 枚の画像の輝度ヒストグラムを正規化し、その累積ヒストグラムをそれぞれ  $H_i^1, H_j^2$  で表す。このとき、カメラ 1 の輝度  $i$  に対応するカメラ 2 の輝度  $j$  を次式のように求める。

$$j = \arg \min_j |H_i^1 - H_j^2| \quad (7)$$

この変換を RGB すべての輝度レベルに対して行う。一度変換テーブルを作成しておく、以後は異なる画像に対しても、同一の変換テーブルを用いて補正が行える。

#### 4-1-4 評価実験

シミュレーション実験と実際の画像に対してキャリブレーションを行い本手法の評価を行なった。

## (1) シミュレーション実験

ここではキャリブレーション精度についての評価実験を行う。まず始めに高解像度カメラと低解像度カメラのパラメータを設定し、世界座標から画像上へ特徴点を変換する。画像上の特徴点に平均0標準偏差0.2のガウスノイズを付加し、幾何学的キャリブレーションを行い幾何補正する。ただし、画像サイズは高解像度側4000×2600画素、低解像度側1000×1000画素とし、特徴点は世界座標で(0, 0, 0)から(225, 150, 0)まで25mm間隔で10×7=70個配置する。

設定したカメラパラメータを表2に示し、推定結果を表3に示す。誤差が含まれていても安定してパラメータが求められることがわかる。特徴点の誤差は基準平面上で平均1.055 pixelであった。

表2 設定カメラパラメータ

	高解像度カメラ	低解像度カメラ
$f$ [mm]	55.0	10.0
$\kappa \times 10^{-4}$ [1/mm <sup>2</sup> ]	1.0	8.0
$T_x$ [mm]	-100.0	-100.0
$T_y$ [mm]	-100.0	-100.0
$T_z$ [mm]	600.0	600.0
$R_x$ [deg]	0.0	0.0
$R_y$ [deg]	30.0	30.0
$R_z$ [deg]	0.0	0.0
$C_x$ [pixel]	2000.0	500.0
$C_y$ [pixel]	1300.0	500.0

表3 カメラパラメータ推定結果

	高解像度カメラ	低解像度カメラ
$f$ [mm]	54.9	10.6
$\kappa \times 10^{-4}$ [1/mm <sup>2</sup> ]	0.999	7.37
$T_x$ [mm]	-100.5	-75.19
$T_y$ [mm]	-101.2	-101.2
$T_z$ [mm]	598.1	635.4
$R_x$ [deg]	0.986	0.889
$R_y$ [deg]	29.90	32.22
$R_z$ [deg]	-0.026	-0.047
$C_x$ [pixel]	1998.9	452.9
$C_y$ [pixel]	1309.1	501.8

## (2) 画像に対するキャリブレーション

キャリブレーションパターンとして半径10mmの円を中心間距離25mm間隔で10×7=70個平面に配置したパターンを用いる。ただし撮影に使用した高解像度カメラの画像サイズは4006×2670画素、低解像度カメラの画像サイズは1006×1016画素である。特徴点として円の重心を検出し、世界座標とそれに対応する画像座標の組を得る。さらに、幾何学的キャリブレーションを行いパラメータを推定する。

高解像度入力画像と低解像度入力画像をそれぞれ図16, 図17に、それぞれのヒストグラムを図18, 図19に示す。

基準平面に投影された特徴点の誤差は平均0.624pixel, 標準偏差0.410, 最大2.342pixelであった。誤差の平均はサブピクセルとなり十分な精度で幾何補正されたことがわかる。

変換後の画像を図20, 図21に、それぞれのヒストグラムを図22, 図23に示す。また、その差分画像を図24に、差分画像のヒストグラムを図25に示す。ただし差分画像はログスケールで示す。



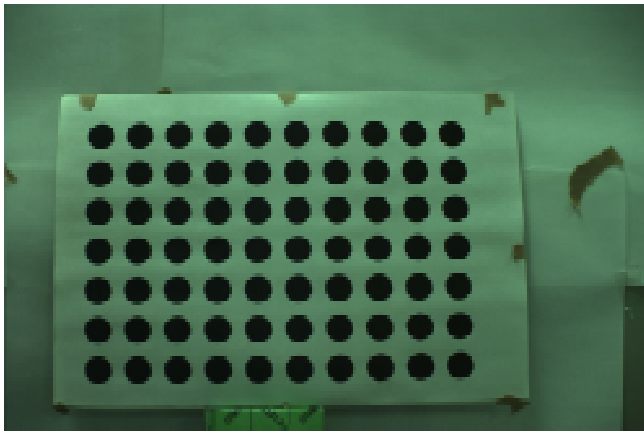


図 16 高解像度入力画像



図 17 低解像度入力画像

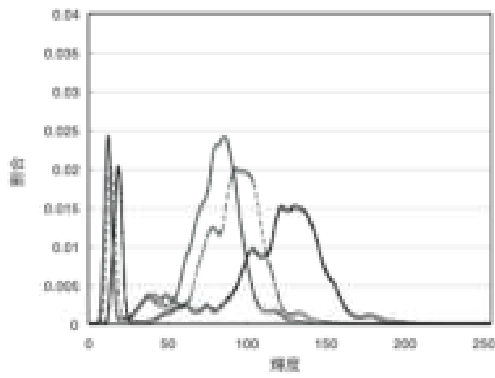


図 18 高解像度入力画像ヒストグラム

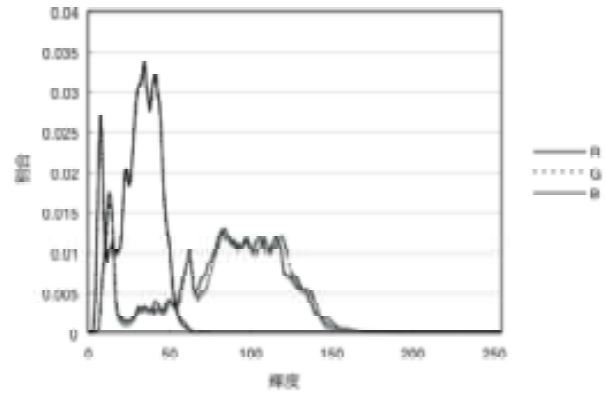


図 19 低解像度入力画像ヒストグラム

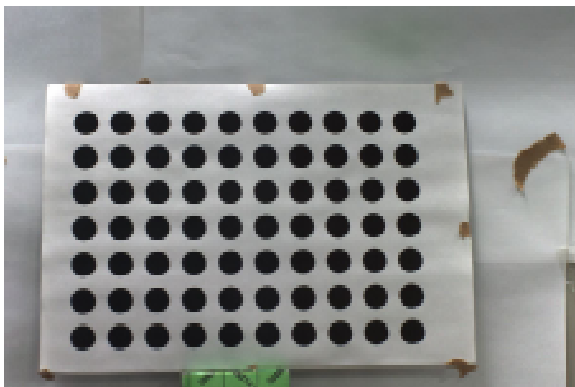


図 20 高解像度補正画像

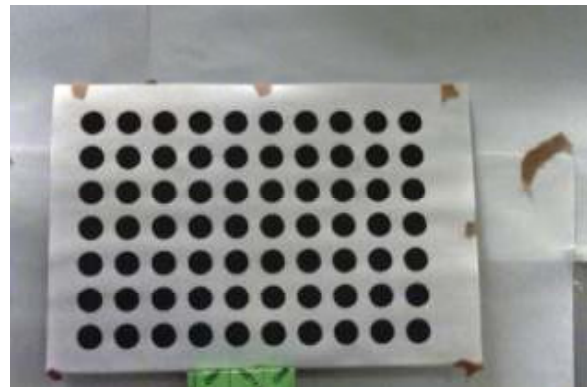


図 21 低解像度補正画像

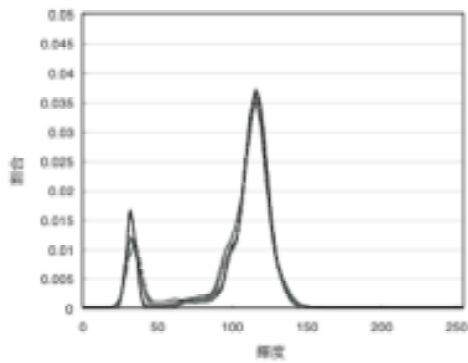


図 22 高解像度補正画像ヒストグラム

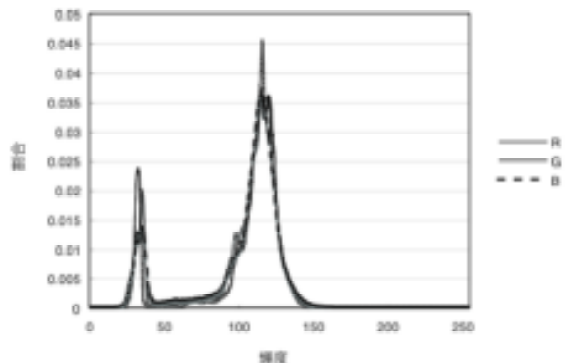


図 23 低解像度補正画像ヒストグラム

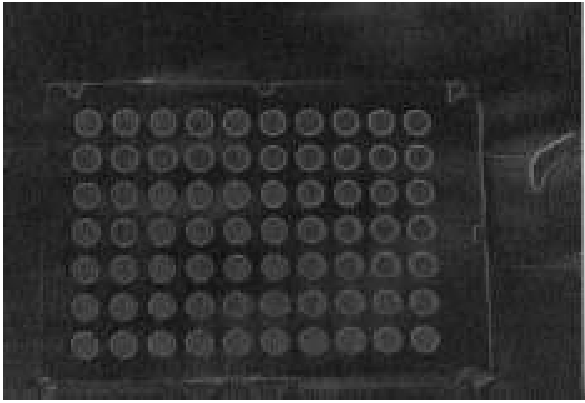


図 24 差分画像（ログスケール）

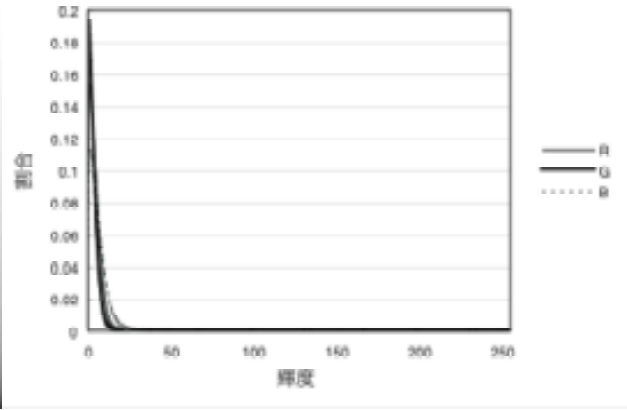


図 25 差分画像ヒストグラム

図 24 よりエッジの部分で輝度に差があるのがわかる．しかし，差分画像のヒストグラムを見てみるとその差は小さいものであった．以上の結果より本手法によって複合センサカメラのキャリブレーションがうまく行われたのを確認した．

## 4-2 リアルタイム高解像度動画の作成

時空間分解能の異なる 2 つの画像系列（高解像度低フレームレート動画，低解像度高フレームレート動画）の情報を統合する手法として，周波数空間的アプローチとモーフィングにより欠落している高解像度画像を生成するアプローチを提案し，アルゴリズムの検討を行った．

### 4-2-1 周波数空間アプローチによる高解像度画像の作成

複合センサカメラから得られる 2 種類の動画から，画像処理によって高解像度・高フレームレート動画を生成するアルゴリズムを開発した．開発手法の有効性を確かめるため，MPEG テストシーケンスに対して各手法を適用し，高解像度動画を生成するシミュレーション実験を行った．また，生成画像のピーク信号対雑音比 (PSNR) を測定し手法の性能を比較した．

#### (1) 高解像度動画の生成

本研究では複合センサカメラから得られた動画をインターネット配信することを想定している．動画のもつデータ量は非常に大きいため，通常は動画をインターネット配信する場合に符号化するが，各フレームは符号化処理の過程で何らかの周波数変換により周波数領域へ移される．したがって，動画をインターネット配信するという前提の下では，高解像度画像の生成処理を画像空間領域と周波数領域の両方で行うことが可能である．画像を周波数空間に移す周波数変換はこれまでに様々な方法が提案され，実用化されているが，本研究では DCT を用いる方法を提案する．DCT を用いた画像生成は次節以降で詳しく述べる．

はじめに，準備として位相相関法による動き推定の手順を説明する．提案手法では，位相相関法を用いて低解像度画像中の動きを推定し，得られた情報を後の処理で利用する．

#### a. 位相相関法による動き推定

連続する 2 つのフレーム  $I_k$ ,  $I_{k+1}$  の間で単純な並進移動 ( $d_x$ ,  $d_y$ ) が生じていると仮定する．すなわち

$$I_k(x,y) = I_{k+1}(x + d_x, y + d_y) \quad (8)$$

が成り立つとする．上式を 2 次元 Fourier 変換すると

$$F_k(u,v) = F_{k+1}(u,v) e^{j2\pi(ud_x + vd_y)} \quad (9)$$

となる．ここで  $F_k = F[I_k]$  とする．すなわち，画像空間領域の並進移動は，Fourier 領域において位相差として現れる．位相相関法はこの性質を利用して動き推定を行う．連続するフレーム  $I_k$  と  $I_{k+1}$  の間の正規化相互パワースペクトルは次式で定義される．

$$\Phi_{k,k+1}(u,v) = \frac{F_{k+1}(u,v)F_k^*(u,v)}{|F_{k+1}(u,v)F_k^*(u,v)|} \quad (10)$$

ここで\*は複素共役を表す. 式 (9) と式 (10) から

$$\Phi_{k,k+1}(u,v) = e^{-j2\pi(ud_x+vd_y)} \quad (11)$$

が成り立つ. 上式の2次元逆 Fourier 変換は

$$c_{k,k+1}(x,y) = F^{-1}[\Phi_{k,k+1}(u,v)] = \delta(x-d_x, y-d_y) \quad (12)$$

となる. ここで  $\delta$  は Dirac のデルタ関数を表す.  $c_{k,k+1}$  は位相相関関数と呼ばれ, 座標  $(d_x, d_y)$  にインパルスが現れる.  $c_{k,k+1}$  に現れるインパルスの座標を調べるにより, 画像中に含まれる並進移動量と方向, すなわち動きベクトルを知ることができる. 図 26 に位相相関関数の例を示す. なお, 方向や移動量の異なる複数の動きが2フレーム間に含まれている場合, 位相相関関数のインパルスはそれらの動きの方向・移動量に対応する座標に複数現れる. しかも, 位相相関関数  $c_{k,k+1}$  の点  $(x, y)$  における値は, ブロック中で  $(x, y)$  だけ移動した物体の面積(画素数)を近似する.

位相相関関数のインパルスの座標を調べるにより, 画像中に含まれる動きベクトルを知ることができる. しかし, 動きベクトルが画像中のどこに対応するかは, 位相相関関数からはわからない. そこで, 位相相関法による動き推定は図 27 に示すような処理の流れで画像の各ピクセルに動きベクトルを割り当てる. 以下, 図の各ブロックの処理を説明する.

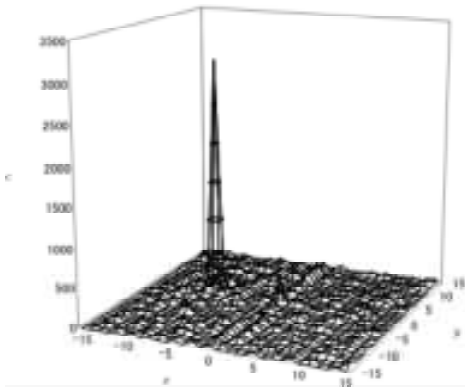


図 26 位相相関関数

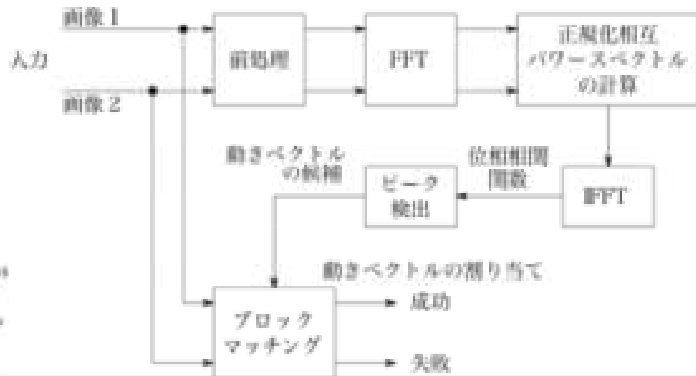


図 27 位相相関法による動き推定

### 前処理

まず, 画像を  $24 \times 24$  画素のブロックに分割する (図 28(a)の波線). 次に,  $24 \times 24$  ブロックを縦に2つ, 横に2つ並べた  $48 \times 48$  ブロックを取る (図 28(a)の灰色のブロック). これにより, 例えば図 28(a)の点Pを含むような  $48 \times 48$  ブロックは4つとなり, これらのブロックの少なくとも1つで点Pの動きが検出できる. その後, ブロックの境界で連続になるようにするため,  $48 \times 48$  ブロックを  $64 \times 64$  ブロックに拡張する (図 28(b)). 以降の処理はすべてこの  $64 \times 64$  ブロックごとに行う. そして, ブロックを拡張した部分に対して, ブロックの周囲で白と黒の中間色 (mid-grey) になるようなぼかし処理を施す. ぼかし処理は次のように行う. ぼかしの重み関数を

$$W(\theta) = \frac{1}{2} \left( 1 + \cos \frac{\pi\theta}{8} \right) \quad (13)$$

と定義する. まず, サイズ  $64 \times 64$  のブロック  $I(x,y)$  の左右に対してぼかし処理を施し,  $I'(x,y)$  を求める.

$$I'(x,y) = \begin{cases} (1-W(x))I(x,y) + W(x) \times 128 & 0 \leq x < 8 \\ I(x,y) & 8 \leq x < 56 \\ (1-W(63-x))I(x,y) + W(63-x) \times 128 & 56 \leq x < 64 \end{cases} \quad (14)$$

次に, ブロック  $I'(x,y)$  の上下に対してぼかし処理を施し,  $I''(x,y)$  を求める.

$$I''(x,y) = \begin{cases} (1-W(y))I'(x,y) + W(y) \times 128 & 0 \leq y < 8 \\ I'(x,y) & 8 \leq y < 56 \\ (1-W(63-y))I'(x,y) + W(63-y) \times 128 & 56 \leq y < 64 \end{cases} \quad (15)$$

ぼかし処理を行った画像ブロックの例を図 29 に示す。

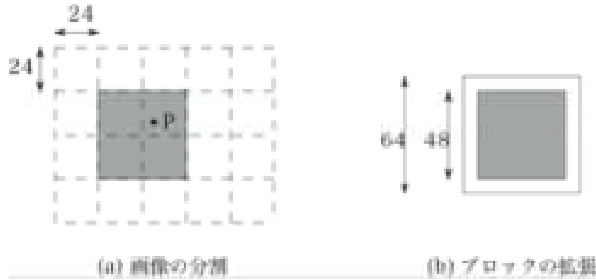


図 28 前処理における分割

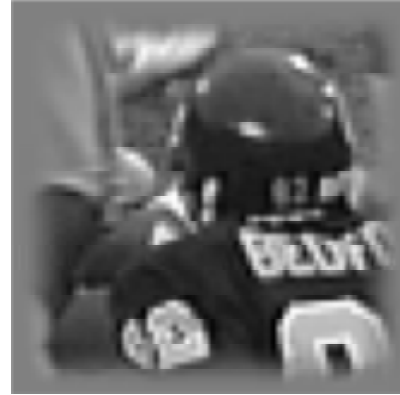


図 29 画像ぼかしの例

## FFT

前処理で得られた  $64 \times 64$  ブロックに対して Fourier 変換を実行する。

## 正規化相互パワースペクトルの計算

2つの画像の対応する位置の  $64 \times 64$  ブロック同士で正規化相互パワースペクトル  $\Phi_{k,k+1}$  (式(10)) を計算する。

## IFFT

正規化相互パワースペクトル  $\Phi_{k,k+1}$  に対して逆 Fourier 変換を実行し、位相相関関数  $c_{k,k+1}$  を求める。

## ピーク検出

次の式で表される集合  $\Lambda$  の各要素(座標)について、位相相関関数の値を調べる。

$$\Lambda = \left\{ (x,y) \mid -32 \leq x < 32, -32 \leq y < 32, x = \frac{i}{\sigma}, y = \frac{j}{\sigma}, i, j \in Z \right\} \quad (16)$$

ここで、高解像度画像と低解像度画像の画像サイズ比を  $\sigma:1$  ( $\sigma \in N$ ) とする。点  $P(x,y) \in \Lambda$  が  $x \notin Z$  または  $y \notin Z$  の場合、双一次補間法により点 P の近傍の点から位相相関関数の値を補間する。そして、位相相関関数の値があるしきい値 T よりも大きくなるような座標  $P \in \Lambda$  を 10 点選択し、それらを動きベクトルの候補としてリストにする。すなわちベクトル  $v_r$  の集合

$$\begin{aligned} V_{k,k+1} &= \{ v_r = (v_{rx}, v_{ry})^T \mid (v_{rx}, v_{ry}) \in \Lambda, c_{k,k+1}(v_{rx}, v_{ry}) \geq T \} \\ &= \{ v_1, v_2, \dots, v_{10} \} \end{aligned} \quad (17)$$

を作成する。

## ブロックマッチング

画像の各画素に対して動きベクトル  $v$  を割り当てる。動きベクトルの候補リスト  $V$  の各要素について、動きベクトルを求めたい画素を中心とした  $3 \times 3$  ブロックによるブロックマッチングを行う。 $V$  の要素の中で輝度値の絶対値差分が最も小さいものを、その画素の動きベクトルとする。すなわち、

$$D = \{ w = (w_x, w_y)^T \mid w_x, w_y \in \{-1, 0, 1\} \} \quad (18)$$

とすると、点  $P(x,y)$  に割り当てる動きベクトル  $v(x,y)$  を次式により求める。

$$v(x,y) = \arg \min_{v_r, v_y} \left\{ \sum_{w_x, w_y} \left| I_k(x+w_x, y+w_y) - I_{k+1}(x+w_x+v_{rx}, y+w_y+v_{ry}) \right| \right\} \quad (19)$$

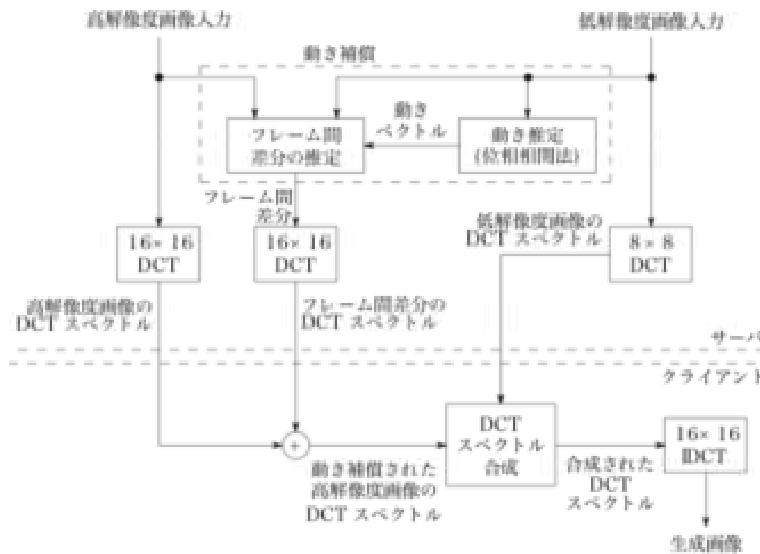


図 30 DCT を用いた高解像度画像生成処理の流れ

## b. DCT を用いた高解像度動画生成アルゴリズム

DCT を用いた高解像度画像生成手法では、生成する高解像度画像の高周波成分と低周波成分を異なる方法によって推定する。

- ・ 生成画像の高周波成分は、画像空間上で動き補償を行った高解像度画像のスペクトルをそのまま利用する。シーン中の動きは低解像度画像を用いて推定する。ただし、補償できない部分については低解像度画像から補間する。
- ・ 生成画像の低周波成分は、画像空間上で動き補償された高解像度画像のスペクトルに低解像度画像のスペクトルを合成することによって生成する。

DCT を用いた高解像度画像生成処理の流れを図 30 に示す。ただし、2 種類の動画の画像サイズ比を 2:1 とする。また、以下の処理は、1 は（輝度成分）のみを利用して行い、その他の処理は YCrCb の各成分ごとに行う。

高解像度画像の生成は次の手順で行う。

- 1) 位相相関法(a 節)により低解像度画像の各画素に対して動きベクトルを求める。
- 2) 1 で求めた動きベクトルを用いて高解像度画像のフレーム間差分を推定する。ただし、動きベクトルが推定できなかった部分は、双一次補間法により低解像度画像から画素値を補間する。
- 3) 高解像度画像とフレーム間差分に対してサイズ 16×16 のブロック DCT を実行する。また、低解像度画像に対してサイズ 8×8 のブロック DCT を実行する。
- 4) 高解像度画像とフレーム間差分のそれぞれの DCT スペクトルを DCT 領域で足し合わせることで、動き補償された高解像度画像の DCT スペクトルを求める。
- 5) 動き補償された高解像度画像の DCT スペクトルに対し、低解像度画像の対応する位置の DCT スペクトルを合成する。
- 6) 合成したスペクトルに対して IDCT(逆離散コサイン変換)を実行することにより、高解像度画像を生成する。

なお、動画のインターネット配信を考える場合は、上の 1 から 3 まだがサーバ側の処理、4 から 6 まだがクライアント側の処理となる。

以下、c 節では動き補償の原理とフレーム間差分の推定手順を示す。その後、d 節で DCT スペクトル合成について述べる。

### c. フレーム間差分を用いた動き補償

動画をインターネット配信する場合、動き補償処理をクライアントで行うとクライアント側の負担が大きくなる．そこで、フレーム間差分を用いて周波数領域で動き補償を行う．この原理について述べる．

今、 $k$  番目のフレームを  $I_k$ 、 $k$  番目と  $k+1$  番目のフレーム間差分を  $S_{k,k+1}$  で表すと

$$I_{k+1} = I_k + S_{k,k+1} \quad (20)$$

の関係がある．すると、DCT の線形性より

$$C[I_{k+1}] = C[I_k] + C[S_{k,k+1}] \quad (21)$$

が成り立つ．ここで  $C[\cdot]$  は DCT スペクトルを表す．すなわち、動き補償された画像の DCT スペクトルを得るためには、フレーム間差分の DCT スペクトルを前フレームの DCT スペクトルと足し合わせるだけでよいことになる．

次に、高解像度画像の 1 周期分のフレーム間差分を推定する処理の流れを説明する．ここで、高解像度画像と低解像度画像のフレームレート比は  $1:R$  ( $R \in N$ )、画像サイズ比は  $2:1$  ( $\sigma=2$ ) とする．

**Step 1:** 高解像度画像の 1 周期に対応する低解像度画像系列  $I_0, I_1, \dots, I_R$  について、隣接する低解像度フレーム間で位相相関法による後ろ向き動き推定 (ME) を行い、動きベクトルの候補リスト  $V_{1,0}, V_{2,1}, \dots, V_{R,R-1}$  を作成する．

**Step 2:**  $bk = 1, fw = R-1$  とする．

**Step 3:** ( $bk$  番目の高解像度画像の推定)

1) Step 1 で求めた後ろ向きの動きベクトルの候補リスト  $V_{bk,bk-1}$  を用いて、低解像度画像  $I_{bk}$  と  $I_{bk-1}$  の間でブロックマッチング (式 21) を行い、低解像度画像  $I_{bk}$  の各画素に対して後ろ向きの動きベクトルを割り当てる．今、 $bk$  番目の低解像度画像  $I_{bk}$  の点  $(x, y)$  に割り当てられた動きベクトルを  $\mathbf{v}(x, y) = (v_x, v_y)^T$  とすると、

$$\begin{aligned} \tilde{I}_{bk}(2x + \Delta_x, 2y + \Delta_y) &= \tilde{I}_{bk-1}(2(x + v_x) + \Delta_x, 2(y + v_y) + \Delta_y), \\ \Delta_x, \Delta_y &\in \{0, 1\} \end{aligned} \quad (22)$$

によって  $bk$  番目の高解像度画像  $\tilde{I}_{bk}$  の各点の画素値を求める．ただし、式 (19) の括弧内の値 (絶対値差分) がある値  $\tau$  より大きければ、動きベクトルの推定に失敗したとし、2 に進む．動きベクトルの推定に成功した場合は 4 に進む．

2)  $I_{bk}$  を起点として前向きに動きベクトルを追跡し、 $I_{fw+1}$  番目のフレームまでの動きの軌道を推定する．この処理は次のように行う．今、動きベクトルの推定に失敗した  $I_{bk}$  の点を  $(x', y')$  とおく．

- $\gamma = bk, \alpha' = x', \beta' = y'$  とする．
- ここで  $\gamma = fw+1$  ならば (e) へ進む．そうでなければ  $\alpha = \alpha', \beta = \beta'$  として (c) へ進む．
- 点  $(\alpha, \beta)$  について低解像度画像  $I_\gamma$  と  $I_{\gamma+1}$  の間のブロックマッチングを行う．Step 1 で求めた動きベクトルの候補リスト  $V_{\gamma+1,\gamma}$  の各要素の符号を反転したベクトルからなるリスト  $V_{\gamma,\gamma+1}^- = V_{\gamma+1,\gamma}^-$  を用いて、次式により  $I_\gamma$  の点  $(\alpha, \beta)$  の動きベクトルを求める．

$$\mathbf{v}(\alpha, \beta) = \arg \min_{\mathbf{v}_r \in V_{\gamma,\gamma+1}^-} \left\{ \sum_{w \in D} |I_\gamma(\alpha + w_x, \beta + w_y) - I_{\gamma+1}(\alpha + w_x + v_{rx}, \beta + w_y + v_{ry})| \right\} \quad (23)$$

ここで、 $D$  の定義は式 (18) と同様である．上式の括弧内の値を最小にする  $\mathbf{v}_r$  を  $\mathbf{v}^0 = (v_x^0, v_y^0)^T$  とおく．前と同様に、上式の括弧内の値 (絶対値差分) が  $\tau$  より大きければ、動きベクトルの割り当てに失敗したとして終了する．

- $\alpha' = \alpha + v_x^0, \beta' = \beta + v_y^0$  とする． $\gamma$  に 1 を加え、(b) へ戻る．

- e)  $f_w+1$ 番目の高解像度画像  $\tilde{I}_{f_w+1}$  から、次式によって  $bk$  番目の高解像度画像  $\tilde{I}_{bk}$  の画素値を求める。

$$\begin{aligned} \tilde{I}_{bk}(2x'+\Delta_x, 2y'+\Delta_y) &= \tilde{I}_{f_w+1}(2(x'+\alpha')+\Delta_x, 2(y'+\beta')+\Delta_y), \\ \Delta_x, \Delta_y &\in \{0,1\} \end{aligned} \quad (24)$$

上の(c)で動きベクトルが推定できなかった場合、3に進む。推定できた場合は4に進む。

- 3) 次式(双一次補間法)により画素値を低解像度画像から補間する。

$$\begin{aligned} \tilde{I}_{bk}(2x'+\Delta_x, 2y'+\Delta_y) &= \frac{1}{4} \left\{ (1-p)(1-q)I_{bk}\left(\left\lfloor \frac{\mu_x}{2} \right\rfloor, \left\lfloor \frac{\mu_y}{2} \right\rfloor\right) + p(1-q)I_{bk}\left(\left\lfloor \frac{\mu_x}{2} \right\rfloor + 1, \left\lfloor \frac{\mu_y}{2} \right\rfloor\right) \right. \\ &\quad \left. + (1-p)qI_{bk}\left(\left\lfloor \frac{\mu_x}{2} \right\rfloor, \left\lfloor \frac{\mu_y}{2} \right\rfloor + 1\right) + pqI_{bk}\left(\left\lfloor \frac{\mu_x}{2} \right\rfloor + 1, \left\lfloor \frac{\mu_y}{2} \right\rfloor + 1\right) \right\}, \end{aligned} \quad (25)$$

$$\Delta_x, \Delta_y \in \{0,1\}, \mu_x = 2x' + \Delta_x + 1, \mu_y = 2y' + \Delta_y + 1, p = \frac{1}{4} + \frac{\mu_x \bmod 2}{2}, q = \frac{1}{4} + \frac{\mu_y \bmod 2}{2}$$

- 4)  $bk-1$ 番目の高解像度画像  $\tilde{I}_{bk-1}$  と、推定した  $bk$  番目の高解像度画像  $\tilde{I}_{bk}$  の間のフレーム間差分

$$S_{bk, bk-1} = \tilde{I}_{bk} - \tilde{I}_{bk-1} \quad (26)$$

を計算する。

- 5)  $bk$  に 1 を加える。

本研究では絶対値差分に関するしきい値を  $\tau=100$  とした。

**Step 4:** ここで  $bk = f_w$  ならば終了する。そうでなければ Step 5 に進む。

**Step 5:** ( $f_w$  番目の高解像度画像の推定)

- 1) Step 1 で求めた後ろ向きの動きベクトルの候補リスト  $V_{f_w+1, f_w}$  の各要素の符号が反転したベクトルからなるリスト  $V_{f_w, f_w+1} = V_{f_w+1, f_w}^{-1}$  を用いて、低解像度画像  $I_{f_w}$  と  $I_{f_w+1}$  の間でブロックマッチング(式(18))を行い、低解像度画像  $I_{f_w}$  の各画素に対して前向きの動きベクトルを割り当てる。今、 $f_w$  番目の低解像度画像  $I_{f_w}$  の点  $(x, y)$  に割り当てられた動きベクトルを  $\mathbf{v}(x, y) = (v_x, v_y)^T$  とすると、

$$\begin{aligned} \tilde{I}_{f_w}(2x+\Delta_x, 2y+\Delta_y) &= \tilde{I}_{f_w+1}(2(x+v_x)+\Delta_x, 2(y+v_y)+\Delta_y), \\ \Delta_x, \Delta_y &\in \{0,1\} \end{aligned} \quad (27)$$

によって  $f_w$  番目の高解像度画像  $\tilde{I}_{f_w}$  の各点の画素値を求める。ただし、式(18)の括弧内の値(絶対値差分)がある値  $\tau$  より大きければ、動きベクトルの推定に失敗したとし、2に進む。動きベクトルの推定に成功した場合は4に進む。

- 2)  $I_{f_w}$  を起点として後ろ向きに動きベクトルを追跡し、 $I_{bk-1}$  番目のフレームまでの動きの軌道を推定する。この処理は次のように行う。今、動きベクトルの推定に失敗した  $I_{f_w}$  の点を  $(x', y')$  とおく。

- a)  $\eta = f_w, \alpha' = x', \beta' = y'$  とする。

- b) ここで  $\eta = bk-1$  ならば(e)へ進む。そうでなければ  $\alpha = \alpha', \beta = \beta'$  として(c)へ進む。

- c) 点  $(\alpha, \beta)$  について低解像度画像  $I_\eta$  と  $I_{\eta-1}$  の間のブロックマッチングを行う。Step 1 で求めた動きベクトルの候補リスト  $V_{\eta, \eta-1}$  を用いて、次式により  $I_\eta$  の点  $(\alpha, \beta)$  の動きベクトルを求める。

$$\mathbf{v}(\alpha, \beta) = \arg \min_{\mathbf{v}_r \in V_{\eta, \eta-1}} \left\{ \sum_{w \in D} |I_\eta(\alpha + w_x, \beta + w_y) - I_{\eta-1}(\alpha + w_x + v_{rx}, \beta + w_y + v_{ry})| \right\} \quad (28)$$

上式の括弧内の値を最小にする  $\mathbf{v}_r$  を  $\mathbf{v}^0 = (v_x^0, v_y^0)^T$  とおく。前と同様に、上式の括弧内の値

(絶対値差分)が  $\tau$  より大きければ、動きベクトルの割り当てに失敗したとして終了する。

d)  $\alpha' = \alpha + v_x^0, \beta' = \beta + v_y^0$  とする。  $\eta$  から 1 を引き、(b) へ戻る。

e)  $bk-1$  番目の高解像度画像  $\tilde{I}_{bk-1}$  から、次式によって  $fw$  番目の高解像度画像  $\tilde{I}_{fw}$  の画素値を求め。

$$\begin{aligned} \tilde{I}_{fw}(2x'+\Delta_x, 2y'+\Delta_y) &= \tilde{I}_{bk-1}(2(x'+\alpha')+\Delta_x, 2(y'+\beta')+\Delta_y), \\ \Delta_x, \Delta_y &\in \{0,1\} \end{aligned} \quad (29)$$

上の(c)で動きベクトルが推定できなかった場合、3に進む。推定できた場合は4に進む。

3) 双一次補間法(式(26))の  $bk$  を  $fw$  に置き換える)により画素値を低解像度画像から補間する。

4)  $fw+1$  番目の高解像度画像  $\tilde{I}_{fw+1}$  と、推定した  $fw$  番目の高解像度画像  $\tilde{I}_{fw}$  の間のフレーム間差分

$$S_{fw, fw+1} = \tilde{I}_{fw} - \tilde{I}_{fw+1} \quad (30)$$

を計算する。

5)  $fw$  から 1 を引く。

**Step 6:**  $bk = fw$  になるまで Step 3 から Step 5 までの処理を繰り返す。

#### d. DCT スペクトル合成

一般的な画像信号は低周波成分を多くもつという性質がある。つまり、画像のもつ情報の大部分は低周波成分に集中しているということが言える。したがって、動き補償を行った高解像度画像の低周波成分に低解像度画像のスペクトルを合成することによって、生成画像を真の高解像度画像に近づけることができる。

今、動き補償された高解像度画像の DCT スペクトル(サイズ  $N_h \times N_h$ ) を  $C_h(u, v)$  とし、それに対応する位置の低解像度画像の DCT スペクトル(サイズ  $N_\ell \times N_\ell$ ) を  $C_\ell(u, v)$  とする(ただし  $N_h = \sigma N_\ell$ )。このとき、合成された DCT スペクトル  $C(u, v)$  を

$$C(u, v) = \begin{cases} w_h(u, v)C_h(u, v) + \frac{N_h}{N_\ell} w_\ell(u, v)C_\ell(u, v) & \text{if } 0 \leq u, v \leq N_\ell \\ C_h(u, v) & \text{otherwise} \end{cases} \quad (31)$$

により求める(サイズ  $N_h \times N_h$ )。サイズの異なるスペクトルを合成するため、低解像度画像のスペクトル  $C_\ell$  にエネルギー補正係数  $N_h/N_\ell$  をかける。また、 $w_h, w_\ell$  はスペクトル合成の重みである。本研究では  $N_\ell = 8$  とし、 $w_h, w_\ell$  として次式で表される関数を用いた。

$$w_h(u, v) = \begin{cases} 0 & \text{if } 0 \leq u < 4 \text{ and } 0 \leq v < 4 \\ \frac{u-3}{5} & \text{if } u \geq 4 \text{ and } v \geq 4 \\ \frac{v-3}{5} & \text{otherwise} \end{cases} \quad (32)$$

$$w_\ell(u, v) = 1 - w_h(u, v) \quad (33)$$

## (2) 評価実験

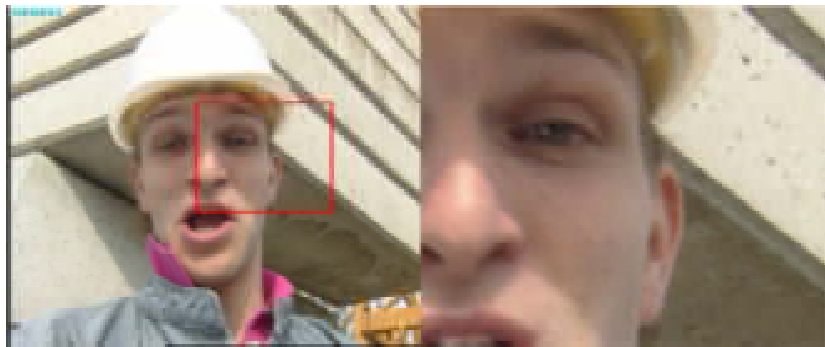
複合センサカメラから得られる 2 種類の動画をシミュレートするため、MPEG テストシーケンス(30 [fps])を用いて高解像度画像生成実験を行った。サイズ  $M \times N$  のテスト画像  $\tilde{I}(x, y)$  に対して 50%の画像縮小を行い( $\sigma=2$ )、サイズ  $\frac{M}{2} \times \frac{N}{2}$  の低解像度画像  $I(x, y)$  を生成する。

そして、原画像を 5 [fps] の高解像度画像、縮小した画像を 30 [fps] の低解像度画像と想定して(R=6)画像生成を行った。すなわち、高解像度画像(原画像)の 1 番目、7 番目、13 番目、…のフレームのみと、低解像度画像(縮小した画像)のすべてを使用して高解像度画像を生成した。本研究の実験に使用した MPEG テストシーケンスを表 4 に示す。



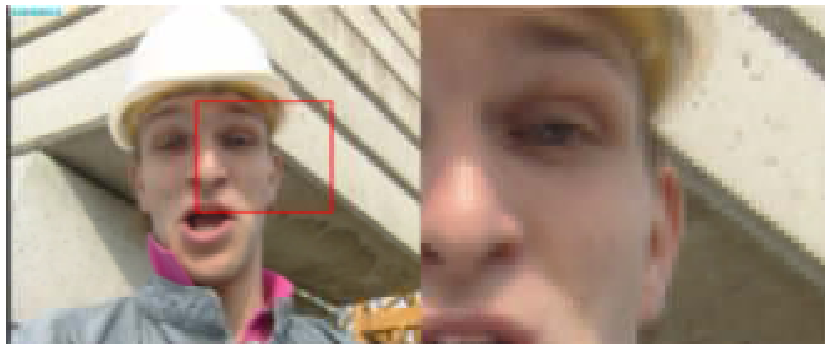
表 4 使用した MPEG テストシーケンス

シーケンス名称	画像サイズ	使用フレーム
Flower garden	352×240	No. 1 ~ No. 115
Football	352×240	No. 1 ~ No. 121
Foreman	352×288	No. 1 ~ No. 295
Hall monitor	352×288	No. 1 ~ No. 295
Mobile	352×240	No. 1 ~ No. 139



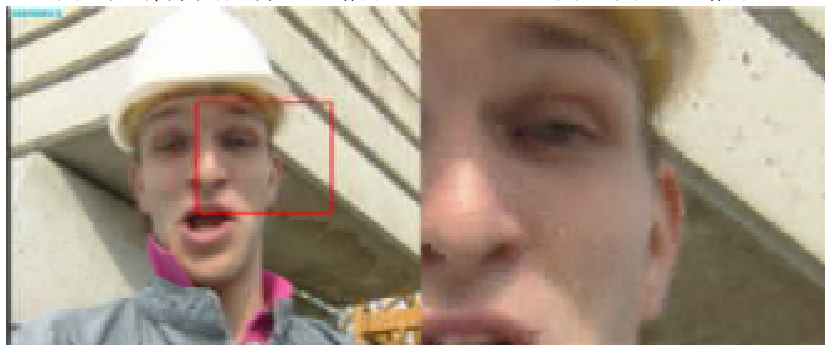
(a) 原画像

(b) (a)を3倍拡大



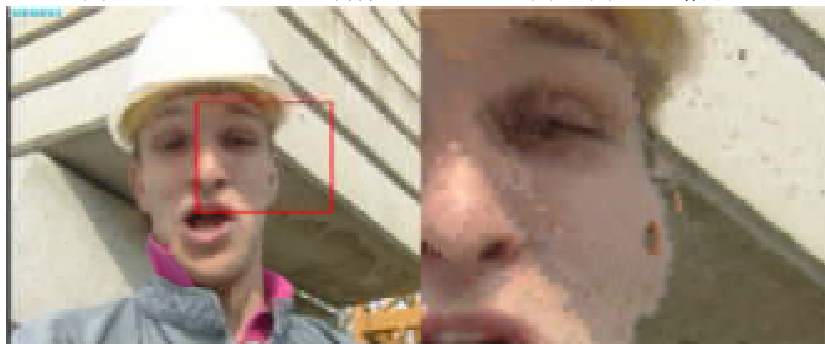
(c) 低解像度画像を2倍拡大

(d) (c)を3倍拡大



(e) DCT スペクトル合成

(f) (e)を3倍拡大



(g) 生成画像(合成なし)

(h) (g)を3倍拡大

図 31 “Foreman” フレーム No. 82

## 高解像度画像の生成

“Foreman”の82番目のフレームの生成結果を図31に示す。各テストシーケンスのPSNR測定結果を表5に示す。表には、拡大した低解像度画像のPSNRも併記してある。なお、表に示してあるPSNRは生成したフレームのPSNRの平均値である。

テストシーケンス“Flower garden”と“Football”，“Mobile”ではPSNRの値が25 [dB]以下と、他の3つのシーケンスよりも結果が悪かった。結果が悪かったこれら3つのテストシーケンスには、シーン中に占める動領域が大きいという共通点がある。このため、動きベクトルの推定に失敗し、低解像度画像から補間を行った画素が生成画像中に多く含まれていたためにPSNRが下がったことが考えられる。より精度の高い動き推定法については今後の課題とする。

表5 PSNRの測定結果（輝度成分）

シーケンス名称	DCTによる合成	低解像度を2倍拡大
Foreman	30.04	27.87
Football	24.37	23.38
Garden	20.53	18.82
Hall monitor	30.46	25.19
Coastguard	28.79	25.21
Mobile	19.65	18.57

## スペクトル合成の有効性

提案手法におけるDCTスペクトル合成の有効性を検証するため、テストシーケンス“Foreman”を用いて、DCTスペクトル合成処理を行わずに画像を生成した。“Foreman”の82番目のフレームの生成結果を図31に示す。

スペクトル合成を行わなかった場合、シーケンス全体のPSNRは28.84 [dB]となった。これを表5の結果と比較すると、DCTスペクトル合成によって+1.20 [dB]のゲインが得られていることになる。したがって、スペクトル合成により、生成画像が真の高解像度画像に近づいていることができる。このため、提案手法によって得られる生成画像の画質は動き推定や動き補償の性能に大きく依存するということがわかる。

## 4-2-2 モーフィングアプローチによる高解像度画像の作成

高解像度低フレームレート動画からはテクスチャ情報を利用し、低解像度高フレームレート動画からはモーション情報を利用することで、それぞれ優れた特徴から高品質な高解像度高フレームレート動画の獲得を提案する。そのために、高解像度だが低フレームレートの動画を撮影するカメラと、低解像度だが高フレームレートの動画を撮影するカメラからなる複合センサカメラを構築し、同一視点・同一視野を持つ時間的・空間的なサンプリングの異なる動画を取得する。この2種類の動画を用いて、高解像度高フレームレートの動画を得ることを目指す。本提案手法では、モーフィングというコンピュータグラフィックス(CG)分野でよく用いられている技法によって、高解像度低フレームレート動画の中間フレームを補間し、高解像度高フレームレート動画を生成する。

### (1) 高解像度高フレームレート動画の生成

#### a. 高解像度動画生成

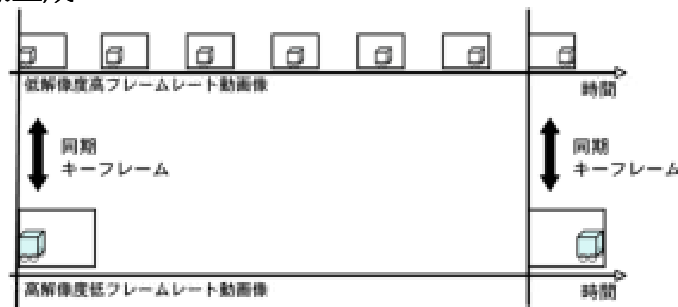


図32 入力動画

提案手法では、図32のような同一視点・同一視野の2種類の動画を入力とする。低解像度高フ

フレームレート動画は、各フレームの解像度を抑える代わりにフレームレートが高い。一方、高解像度低フレームレート動画は、フレームレートを抑える代わりに各フレームの解像度を高めている。また、これら2種類の動画は、フレームレートが異なるものの、高解像度画像の撮像周期で同期撮像されることで、キーフレームと呼ぶ同一時刻のフレームを持つ。低解像度高フレームレート動画は時間的サンプリングの面で優れ、高解像度低フレームレート動画は空間的サンプリングの面で優れている。本提案手法では、これら両者の長所を活かして、高解像度高フレームレート動画を生成する。具体的には図 33 に示すように、高解像度画像中の動領域をモーフィングにより生成し、静止背景と合成することで、高解像度画像のフレームを補間する。一般的に入力の高解像度低フレームレート動画のように、フレームレートが低い場合、特徴点のトラッキングが難しい。また、モーフィングでの特徴点の位置変化は線型移動と仮定することになる。そこで、本提案手法では、フレーム数の多い低解像度画像から動領域の特徴点を検出し、特徴点のトラッキングにより求めた位置関係変化を高解像度画像のモーフィングに用いる。このように低解像度高フレームレート動画を用いることで、トラッキングを比較的容易に行え、さらには、非線形な動領域の変化をモーフィングにより再現できる点が本手法の特徴である。また、キーフレームは、両画像において時間的に対応がとれているため、高解像度と低解像度の特徴点の対応付けも容易である。

具体的な提案アルゴリズムのフローを図 34 に示し、以下に詳細を述べる。

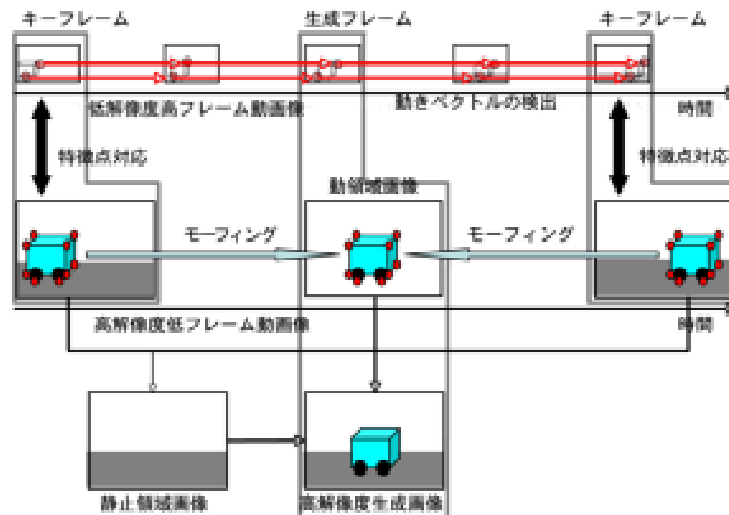


図 33 高解像度動画の生成

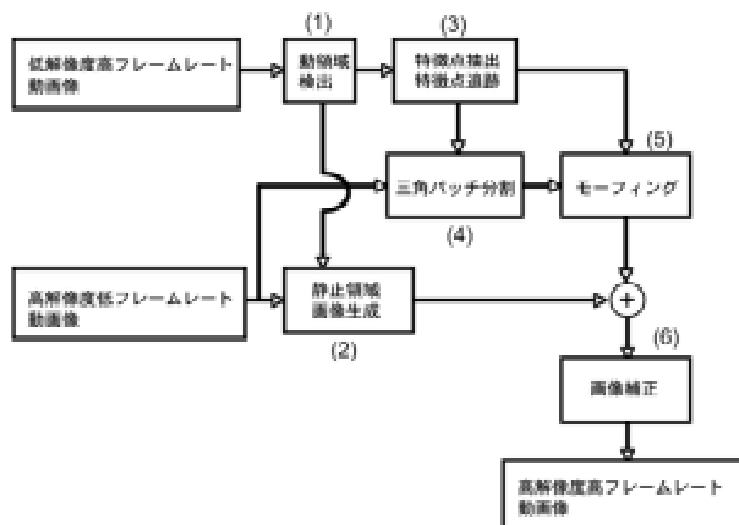


図 34 高解像度動画生成の流れ

- (1) 低解像度高フレームレート動画を用いて、動領域を検出する。
- (2) 動領域検出結果から静止領域マスクを作成し、高解像度高フレームレート動画と AND をとることで、静止領域の画像生成する。

- (3) 動領域から特徴点を抽出する。この特徴点を低解像度高フレームレート動画像において追跡することで、特徴点の動きベクトルを検出し、動領域のモーション情報を得る。特徴点抽出にはハリスオペレータを、特徴点追跡には IPAN トラッカ (Image and Pattern Analysis Group tracker) を使用する。
- (4) 低解像度画像において抽出された特徴点位置を、同じ時刻の映像が得られているキーフレームにおいて高解像度画像に適用する。この特徴点位置をもとに、ドロネー分割によって、高解像度画像の動領域を三角パッチにポリゴン分割する。
- (5) (3) で得られた動きベクトルと (4) で得られた高解像度ポリゴンを用いて、高解像度の補間フレームの動領域画像をモーフィングにより生成する。モーフィングは、ポリゴン単位で処理することで、形と色の両方を補間できるワーピング法を用いる。
- (6) (2) で得られた静止領域画像と (5) で得られた動領域画像を重ね合わせることで、高解像度の補間フレームを生成する。ただし、特徴点抽出・追跡の精度の問題で動領域のテクスチャの欠けや歪みが生じることがある。これに対して、同じ時刻の低解像度画像を利用し画像補正を行う。

## b. 動領域検出

本手法では、補間フレームの静止領域は、高解像度動画像キーフレームの静止領域を使用し、動領域はモーフィングによって生成する。よって、まず低解像度高フレームレート動画像の動領域を検出し、静止領域と動領域を分けたマスク画像を生成する。マスク画像は、低解像度高フレーム画像を用いたフレーム間差分、二値化、拡大法により生成される。

図 35 の生成フレームにおいて、前フレームと後フレームのそれぞれと差分をとることで得られたフレーム間差分画像を二値化し、その AND をとることで動領域の大まかな輪郭が現れる。ただし、動領域内部でテクスチャの変化がない平坦部分は、二値画像の静止領域側に入っていることがあるので、動領域を拡大処理する。これで、動領域と静止領域を分割したマスク画像が得られる。

## c. 静止領域画像生成

高解像度高フレームレート動画像から生成画像の静止領域画像を生成する。図 36 に示すように、生成フレームの前後の近傍高解像度キーフレームから対応するマスク画像を用いて静止領域を抽出する。

ここで、各高解像度フレームに対応するマスク画像は低解像度フレームより生成しているため、マスク画像を高解像度画像の解像度に拡大して適用する。このようにして生成された近傍フレームの静止領域画像の OR をとることで、安定した静止領域画像を得る。

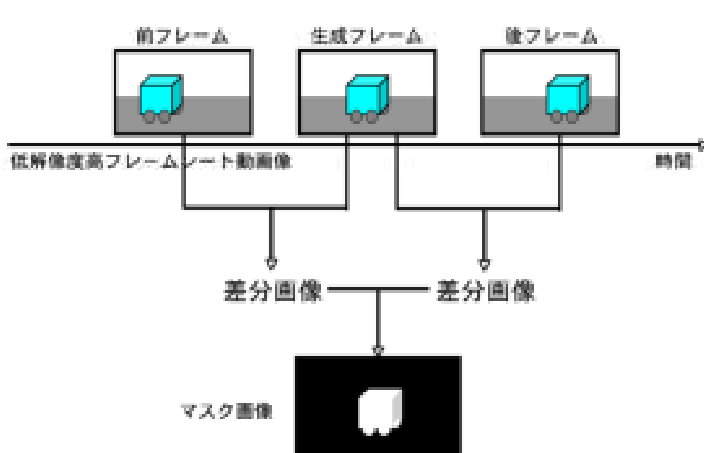


図 35 動領域検出

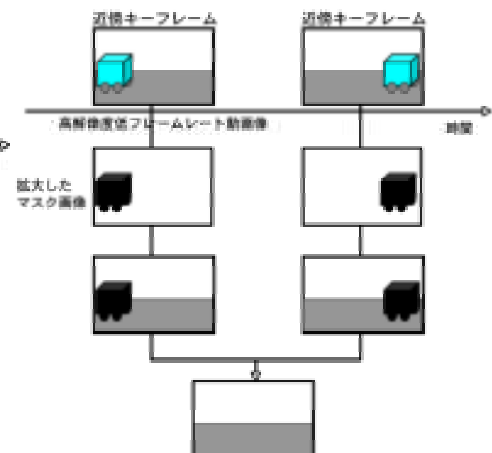


図 36 静止領域画像生成

## d. 特徴点抽出・追跡

モーフィングに用いる動領域のモーション情報を、低解像度高フレームレート動画像から動領域中の特徴点を抽出し、キーフレーム間隔で追跡することで求める。本手法では、特徴点抽出に、ハリスオペレータを使用した。ハリスオペレータは、画像中の輝度勾配により算出した応答関数の

値が局所最大となる点を特徴点として検出する．このため，動領域テクスチャの回転，拡大，縮小が生じてても，比較的安定して対応点として検出できるため，本研究が対象とする動領域中の対応追跡に向いている．

特徴点追跡には，IPAN(Image and Pattern Analysis Group)トラックを使用する．このトラックへの入力は，追跡する各フレームそれぞれでの特徴点座標である．すなわち，あらかじめ各フレームで得た特徴点を隣接フレーム間で対応づけるという手法をとる．隣接3フレーム間の特徴点  $P_{k-1,n}, P_{k,j}, P_{k+1,m}$  について式(34)で示すコスト関数が最小になる組み合わせを対応点とする．

$$\delta(P_{k-1,n}, P_{k,j}, P_{k+1,m}) = w_1 \left( 1 - \frac{\overrightarrow{P_{k-1,n}, P_{k,j}} \cdot \overrightarrow{P_{k,j}, P_{k+1,m}}}{\| \overrightarrow{P_{k-1,n}, P_{k,j}} \| \| \overrightarrow{P_{k,j}, P_{k+1,m}} \|} \right) + w_2 \left( 1 - \frac{\sqrt{\| \overrightarrow{P_{k-1,n}, P_{k,j}} \| \| \overrightarrow{P_{k,j}, P_{k+1,m}} \|}}{\| \overrightarrow{P_{k-1,n}, P_{k,j}} \| + \| \overrightarrow{P_{k,j}, P_{k+1,m}} \|} \right) \quad (34)$$

第1項は動きベクトルの向きの変化，第2項は動きベクトルの大きさの変化を示している．

動領域中の特徴点は，動画像内において一時的なオクルージョンによって消失，出現する．本手法では1フレームのみのオクルージョンならば，特徴点の軌跡を接続できるように対応している．IPANトラックは，前提条件として特徴点軌跡は滑らかで，速度ベクトルが急激に変化しないことを仮定している．

### e. 三角パッチ分解

モーフィングにより動領域をフレーム補間するためには，領域分割する必要がある．本手法では，ドロネー分割を用いて特徴点を頂点とした三角パッチにより動領域を分割する．ドロネー分割は，入力として特徴点の座標情報のみを使用する．図37にドロネー分割の流れを示す．

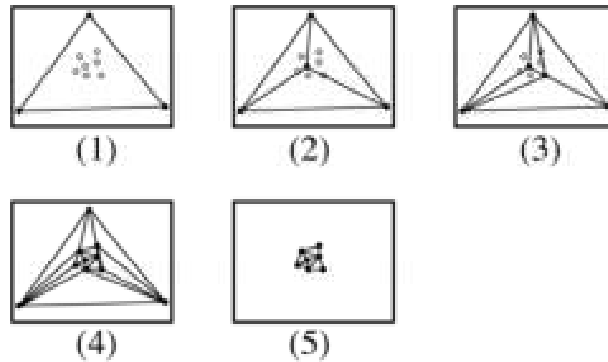


図37 ドロネー分割

- (1) 分割に使用する全ての特徴点を囲む十分大きな三角形を置く．
- (2) 特徴点を1つずつ処理していく．ある三角形の内部に特徴点が位置する場合は，その三角形が3つに分割される．
- (3) 特徴点が2つの三角形の共有边上に位置する場合は，その三角形がそれぞれ2つずつに分割，つまり2つの三角形が4つになる．
- (4) (2)または(3)の操作を繰り返し，全ての特徴点を処理する．
- (5) (1)で置いた三角形およびその三角形と頂点を共有する三角形を削除する．

### f. モーフィング

本手法では，モーフィングにより高解像度動画のキーフレーム間のフレームの動領域画像を生成する．モーフィングに用いる動領域のテクスチャ情報は生成フレームの近傍の高解像度キーフレームから得られる．図38のように，モーフィングは三角パッチ単位で処理を行う．特徴点追跡，ドロネー分割により近傍の高解像度キーフレームと生成フレームの三角パッチは対応がとられている．モーフィングでは，対応している高解像度キーフレーム中の対応パッチ内の輝度情報を補間により生成する．具体的には，生成フレームの動領域画像パッチ内の画素  $P(x, y)$  に対応する座標  $P'(x', y'), P''(x'', y'')$  をキーフレーム1, 2のソースパッチ1, 2からそれぞれ探し出す．

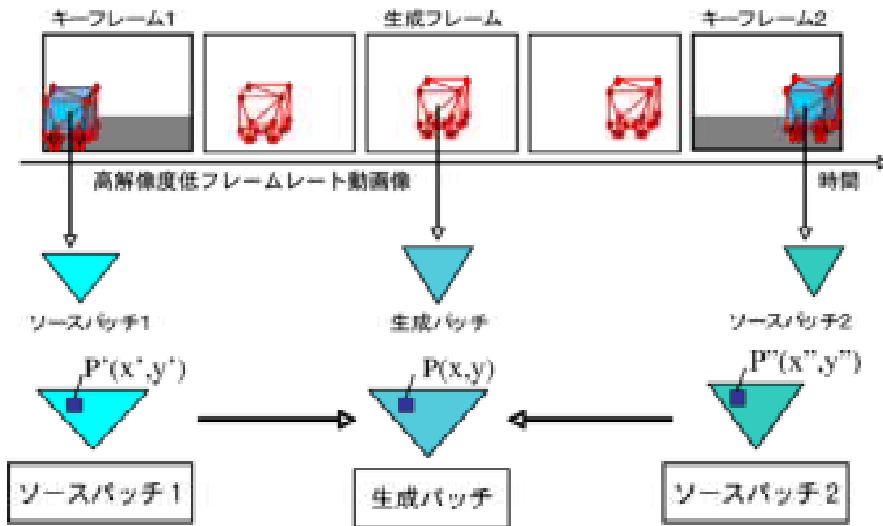


図 38 モーフィング

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (35)$$

対応付けには式(35)のアフィン変換式を用いる. 特徴点追跡によって, 生成パッチの3つの頂点とソースパッチの3つの頂点の対応付けができているので, この値を代入し, 連立方程式を解くことで6つのパラメータ ( $a_{11}, a_{12}, a_{21}, a_{22}, b_1, b_2$ ) を求める. パラメータが決まると生成パッチ画素  $P(x, y)$  に対応するソースパッチ上の座標を計算できる. 輝度値は座標  $P', P''$  それぞれの4近傍画素輝度値からの線形補間(式(41), 図48)によって求める. このようにして求められた近傍キーフレーム中の対応画素  $P'(x', y'), P''(x'', y'')$  の輝度値  $I(x', y'), I(x'', y'')$  から, 式(36)を用いて生成パッチ中の対応画素の輝度値  $I(x, y)$  を求めることで, 高解像度動領域画像のパッチを生成することができる. ここで, 式(36)において動領域の時間的輝度変化は線型であると仮定している.

$$\begin{aligned} I(x', y') = & (I(x_1, y_1)(1-dx) + I(x_2, y_1)dx)(1-dy) \\ & + (I(x_1, y_2)(1-dx) + I(x_2, y_2)dx)dy \end{aligned} \quad (36)$$

### g. 画像補正

カットによるシーンチェンジや激しい動きや輝度変化により特徴点検出やトラッキングが失敗した場合, モーフィングによる動領域の生成において画像の欠けや歪みが発生し, 画質が著しく低下する場合がある. このような, 生成画像の破綻部位を検出し, 補正を行うことで画像劣化を回避する.

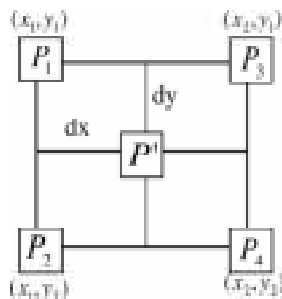


図 39 線形補間

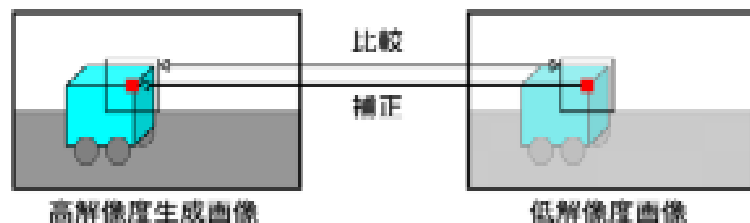


図 40 画像補正

高解像度生成画像を同時刻の低解像度画像と同じ大きさに縮小し, 画素ごとに任意のウインドウサイズで差分をとり, このウインドウ内の平均を評価することで, 大域的に比較し画像の破綻部分を検出する. 検出されたウインドウの中心ピクセルを低解像度から補間することで, 生成画像の破綻部分を補正する. ここで, 低解像度画像と生成画像との比較に窓平均をもちいているのは, ノイ

ズや本手法による解像度向上を破綻部位であると誤検出しないためである。このように、生成画像内の部位ごとに補正を行うことで、高解像度処理が成功した部位を保ったまま、破綻部位のみを補正するため、画像全体としての画質向上を図ることができる。

## (2) 実験

提案する高解像度画像生成手法を検証するために、画像生成実験を行った。実験では、デジタルビデオカメラ A201bc(BASLAR 社, 1006×1016pixel, 30fps)で撮影した動画を以下のように縮小やフレームを間引くことで、複合センサカメラの出力画像と想定して用いた。図 41 に入力動画を例として示す。

- ・ 低解像度高フレームレート動画像：503×508pixel, 30 fps
- ・ 高解像度低フレームレート動画像：1006×1016pixel, 30/7fps



図 41 入力画像：(左)高解像度画像，(右)低解像度画像

提案手法による高解像度画像生成結果を図 42 に示す。この結果より、高解像度動画像には存在しないフレームが、本手法により生成できていることが確認できた。また、本手法により、低解像度高フレームレート動画像同様、30fps の画像が生成でき、同様の動きが再現されていることが確認できた。解像度の向上を分かりやすく示すために、図 43(右)に図 42 の矩形枠部分を拡大して示す。また、比較のため、図 43(左)に低解像度高フレームレート動画像の対応するフレームの対応する部位を示す。図 43(左)と(右)の比較により、視覚的に明らかに本提案手法による生成画像の解像度が向上していることがわかる。本手法において、特徴点追跡が失敗した場合、モーフィングによる動領域の生成が破綻し、画質が著しく低下する場合がある。図 44(左)に特徴点抽出に失敗した場合の生成画像を例として示した(この例では、画像補正処理は適用していない)。このような場合においては、本手法の特徴点の対応関係の前提条件が崩れ、動領域に画像の欠けが発生し、画質が劣化している。この結果は、図 44(中)の低解像度高フレームレート画像と比較して、主観的には画質の悪い画像となっている。図 44(右)に画像補正処理を行った例を示す。図 44(右)の画像補正を行った結果は図 44(中)の低解像度画像を用いて補正を行っているために、著しい画質改善は見られないが、図 44(左)の未補正結果の様な画質劣化は改善されている。以上の結果よりカットによるシーンチェンジや激しい動きや輝度変化により特徴点検出やトラッキングが失敗した場合における画像劣化を補正できることを確認した。



図 42 高解像度生成結果(1006×1016 pixel)



図 43 (上)低解像度画像, (下)高解像度生成画像

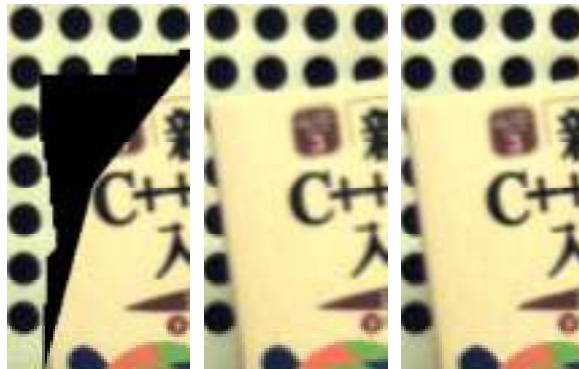


図 44 (左)破綻した高解像度生成画像 (中)低解像度画像 (右)補正結果

### (3) おわりに

本研究では、同一視野・同一視点で、解像度とフレームレートの異なる2種類の動画像から、高解像度高フレームレート動画像を生成するためのアルゴリズムを提案した。また、実験により複合センサカメラを想定した、2種類の動画像を入力として、本提案手法の有効性を確認した。

本手法において、高解像度高フレームレート動画像を得るには特徴点抽出・追跡が重要である。現在のところ、複雑な動きや激しい画像の輝度変化に関しては、特徴点追跡に失敗する場合があります。このようなシーンに関しては大きな解像度改善が得られないという問題がある。特徴点追跡手法として用いた IPAN トラックは、それ自体では高精度の計算が可能であるが、あらかじめ抽出した特徴点の対応をとっていく手法であるため、特徴点抽出の時点で、各フレームで対応点が抽出されていなければ精度の良い結果が得られない。また、特徴点抽出手法として用いたハリスオペレータでは画像の輝度勾配を用いるため、動領域と背景の輝度値が近いと動領域のコーナー(特徴点)の抽出が困難となるといった問題もある。

今後、特徴点抽出自体の精度向上や近傍特徴点との幾何的や運動的な関係を拘束として用いることで、特徴点抽出や追跡手法を改善し、さらに適応範囲が広く安定した手法への改善を図っていく予定である。また、本手法は、入力動画像の解像度差やフレームレート差が大きいほど、センサの低コスト化や映像データ量の削減の意義が大きい。しかしながら、これらの差が大きいほど、生成画像の画質改善が困難となる。今後、この解像度やフレームレートの様々な組み合わせに対し、生成画像の主観や客観的画質を評価を行うことで、本手法の限界および、解像度やフレームレート差のバランスを評価していく予定である。

### 4-3 高解像度全方位カメラ用光学系の設計

提案した複合視覚センサ用の全方位ミラーの設計試作を行った。全方位ミラーと組み合わせることにより、複合視覚センサを全方位の広視野角で利用することができる。本研究では、双曲面ミラーを用いた全方位光学系を設計作成した。試作した複合センサに用いた2台のカメラ(画像サイズ4008×2672画素, セルサイズ9.0×9.0μmと画像サイズ1008×1018画素, セルサイズ9.0×9.0μm)のCCDを用いている。試作した全方位ミラーは、これらカメラを想定して光学設計を行った。



また、更に高解像度化に向く 2 枚反射全方位光学系のシミュレーションによる光学特性の評価とミラーの設計を行った。

#### 4-3-1 双曲面型全方位視覚センサ

双曲面を用いた全方位視覚センサは、図 45 に示すように、鉛直下向き of 双曲面鏡と上向き of カメラから構成される。双曲面鏡は(37), (38) 式のように表され、二つの焦点(0, 0, c), (0, 0, -c)を持ち、カメラは双曲面の中心軸と同軸に主点が双曲面鏡のその一方の焦点(0, 0, -c)にくるよう配置されている。このように配置することで、センサの周囲 360° を一度に撮像することができる。また、図 45 に示すように双曲面鏡のミラー焦点に向かって入ってくる光は、双曲面の特性によりミラーに反射後、もう一方の焦点に向かう。すなわち、画像上での任意の写像点 p(x, y) と俯角、方位角 の関係は、(39)-(41) 式から一意に決まる。この式から、全方位入力画像を図 46 に示すようにミラー焦点を視点とした透視変換画像やパノラマ画像に容易に変換できるという特性を持つ。

$$\frac{X^2 - Y^2}{a^2} - \frac{Z^2}{b^2} = -1, Z > 0 \quad (37)$$

$$c = \sqrt{a^2 + b^2} \quad (38)$$

$$i = \tan^{-1}(y/x) \quad (39)$$

$$a = \tan^{-1} \frac{(b^2 + c^2) \sin c - 2bc}{(b^2 + c^2) \cos c} \quad (40)$$

$$c = \tan^{-1} \frac{f}{\sqrt{x^2 + y^2}} \quad (41)$$

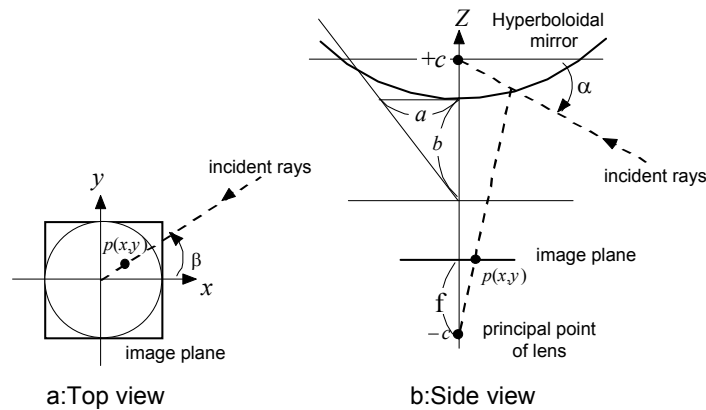


図 45 全方位視覚センサの構造

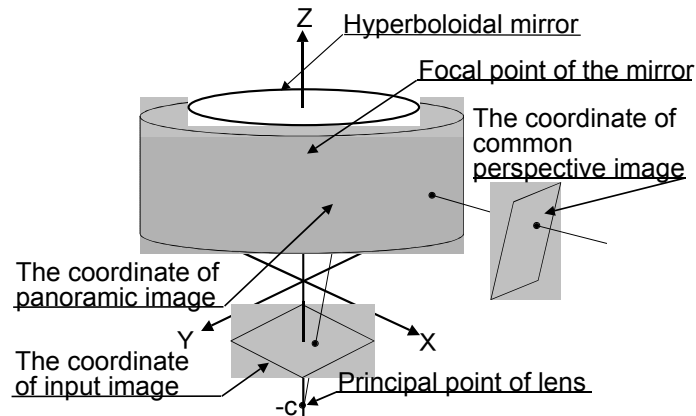


図 46 変換画像の幾何学的関係



図 47 試作した双曲面全方位ミラー

表 6 試作双曲面全方位ミラーのパラメータ

曲率半径	R = 40.856
コニック係数	k = -1.6154

試作センサカメラに用いるレンズの光学特性(焦点深度や光学収差)を実験的に評価した。2台のカメラの CCD (画像サイズ 4008×2672 画素, セルサイズ 9.0×9.0 μm と画像サイズ 1008×1018 画素, セルサイズ 9.0×9.0 μm) と光学的評価により得られたレンズ特性を想定して, 全方位複合センサカメラの光学シミュレーションを行った。このシミュレーションの結果をもちいて, 単一双曲面形状の全方位ミラーの設計を行い, 試作を行った。図 47 に試作した全方位ミラーを示す。

#### 4-3-2 二枚反射全方位視覚センサ

二枚反射全方位視覚センサの光学系では, 反射曲面として 2 葉双曲面を用いた。2 葉双曲面とは双曲線を Z 軸周りに回転させることで得られる曲面である。ここで, 図 49 に 2 葉双曲面の断面を示す。この双曲面は次式で表される。

$$Z = \frac{r^2}{R + \sqrt{R^2 - (1+k)r^2}} = \frac{-1}{1+k} \left[ R - \sqrt{R^2 - (1+k)r^2} \right] \quad (42)$$

なお,  $k (< -1)$ ,  $R$  はそれぞれコニック係数, 曲率半径を表し, 双曲面の形状を定義する定数である。本光学系では 1 枚のミラーにつき, 2 葉のうち片方を使用し, 2 枚の双曲面ミラーの形状が同一である必要はない。双曲面の近い方の焦点を  $F1$ , 遠い方の焦点を  $F1'$  とすると, 各焦点の Z 座標は以下のようになる。

$$F1 : Z = \frac{\sqrt{-k}}{\sqrt{(1+k)^2}} R - \frac{-1}{(1+k)} R \quad (43)$$

$$F1' : Z = \frac{\sqrt{-k}}{\sqrt{(1+k)^2}} R + \frac{-1}{(1+k)} R \quad (44)$$

2 枚反射光学系は, 図 48 に示すように, 鉛直下向きの双曲面ミラー (第一ミラー) と鉛直上向きの双曲面ミラー (第二ミラー), そして第一ミラーの内側にカメラを設置した構成をとる。双曲面は図 49 の示すように, 焦点  $F1$  に向かって入射する光線は他方の焦点  $F1'$  に向かって反射するという幾何的な性質をもつ。そこで 2 枚ミラーシステムでは, 第一ミラーの一方の焦点  $F1'$  と第二ミラーの一方の焦点  $F2$  を一致させ, 第二ミラーの他方の焦点  $F2'$  とレンズ中心  $P$  を一致させる構造をとる。この構造によって, 第一ミラーの焦点  $F1$  に向かって入射した光線がカメラで撮像されるため, 2 枚ミラーシステムにおいても単一の投影中心が実現される。つまり, 光学特性が透視投影となるので, 2 枚ミラーシステムは双曲面ミラーを用いた従来の全方位視覚センサ同様, 様々な画像処理に適している。ここで,  $F1$  に対する入射視野角 (図 48  $\angle \alpha$ ) とカメラに対する入射視野角 (図 49  $\angle \beta$ ) の関係を示す。図 48 の第一ミラーの焦点  $F1$  を原点とした極座標系を考える。第一ミラーのコニック係数と曲率半径をそれぞれ  $k_1, R_1$  とすると, 式(42)は以下のように変形できる。

$$\rho = \frac{R_1}{1 + \sqrt{-k_1} \cos \alpha} \quad (45)$$

ただし,

$$Z = R_1 / (1 + \sqrt{-k_1}) - \rho \cos \alpha, \quad r = \rho \sin \alpha \quad (46)$$

入射光線  $L1$  が第一ミラー上の点  $P1$  で反射する。点  $P1$  におけるミラーの傾きは,

$$m1 = \frac{dr}{dZ} = \frac{-R_1 - (1+k_1)Z}{r} \quad (47)$$

となる。式(46)を用いて、極座標表示にすると、

$$m1 = -\frac{\sqrt{-k_1 + \cos \alpha}}{\sin \alpha} \quad (48)$$

と表される。入射角と反射角が等しいという反射の性質を用いると、入射光 L1 の角度 ( $\angle \alpha$ ) と反射光 L2 の角度 ( $\angle \beta$ ) の関係は次のようになる。

$$\tan \beta = \frac{(1+k_1)\sin \alpha}{2\sqrt{-k_1 + (1-k_1)\cos \alpha}} \quad (49)$$

第二ミラーのコニック係数と曲率半径をそれぞれ  $k_2, R_2$  とすると、式(49)の  $\angle \alpha$  と  $\angle \beta$  の関係を用いて、 $\angle \beta$  と反射光 L3 の角度 ( $\angle \gamma$ ) の関係は次のように表すことができる。

$$\tan \gamma = \frac{(1+k_2)\sin \beta}{2\sqrt{-k_2 + (1-k_2)\cos \beta}} \quad (50)$$

式(49)と式(50)より、2枚ミラーシステムに対する入射視野角 ( $\angle \alpha$ ) とカメラに対する入射視野角 ( $\angle \gamma$ ) は2枚の双曲面ミラーのコニック係数 ( $k_1, k_2$ ) を用いて、

$$\tan \gamma = \frac{(1+k_1)(1+k_2)\sin \alpha}{2(\tilde{k}_1 + \tilde{k}_2 - k_1\tilde{k}_2 - \tilde{k}_1k_2) + (1+k_1(k_2-1) + 4\tilde{k}_1\tilde{k}_2 - k_2)\cos \alpha} \quad (51)$$

と表すことが出来る。ただし、 $\tilde{k}_1 = \sqrt{-k_1}$ 、 $\tilde{k}_2 = \sqrt{-k_2}$  である。

したがって、入射光線 L1 (入射方位角  $\theta$ 、入射角  $\alpha$ ) と画像上の写像点 (方位角  $\theta'$ 、半径  $r'$ ) の関係は、 $f$  を焦点距離とし、

$$\begin{aligned} \theta' &= \theta \\ r' &= f \tan \gamma \end{aligned} \quad (52)$$

と表せる。また、式(51)よりカメラに対する入射角は、第一ミラーに入射する角度とミラーのコニック係数によって決まり、曲率半径には依存しないことがわかる。

2枚ミラーシステムは、従来の単一ミラーの場合に比べ、カメラに対する入射視野角が小さくなり、像面湾曲が抑制されるという特性がある。図 48 の  $\angle \beta$  と  $\angle \gamma$  の比較から、カメラに対する入射視野角が小さくなっていることが確認できる。この像面湾曲の抑制により、2枚ミラーシステムは全方位視覚センサの高解像度化や小型化を実現できる。

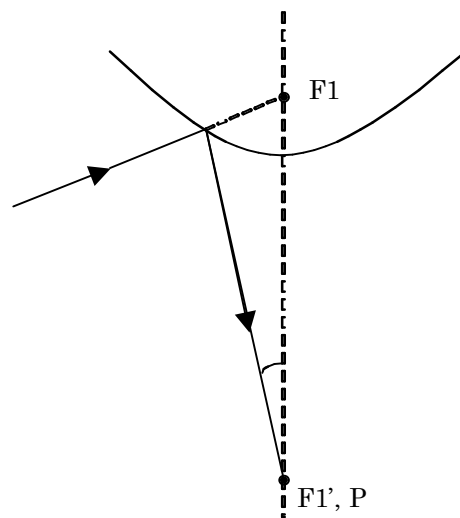
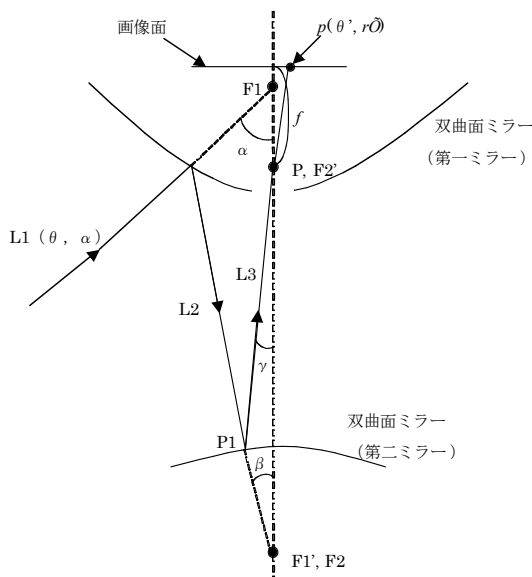


図 48 2枚ミラーシステムの幾何学特性 図 49 双曲面ミラーの幾何学特性

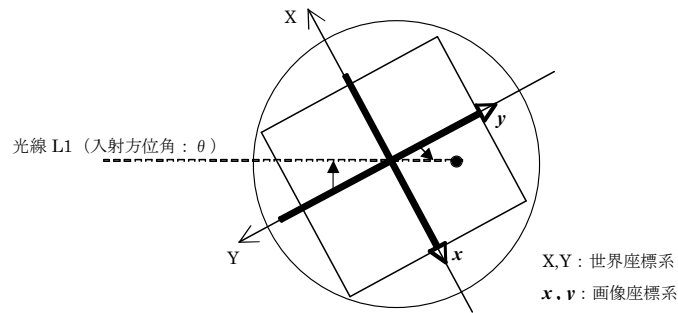


図 50 入射光線の方位角と写像点の方位角の関係  
(図 48 を上方からみた図)

### (1) 抑制効果の確認

2 枚ミラーシステムの像面湾曲抑制効果の検証のためのシミュレーション実験を行なった。

従来の 1 枚ミラーと提案する 2 枚ミラーの全方位センサをシミュレーションし、イメージ面での光線の収束の様子を観察した。1 枚ミラーの双曲面ミラーのパラメータは、前節の試作ミラーも物を比較として用いた。レンズは焦点距離が 80mm の理想レンズを使用した。1 枚ミラーのパラメータを表 6 に、2 枚ミラーのパラメータを表 7 に示す。像面湾曲の大きさは焦点位置のタンジェンシャル面における移動量で表した。この実験では、視野角を仰角 30 度～-60 度でシミュレーションしたので、仰角 30 度方向から入射した光線の焦点位置から、-60 度方向からの入射光線の焦点位置の鉛直上方 (Z 軸正方向) への移動量で評価した。

双曲面ミラーを 1 枚使用した従来の全方位視覚センサの全体の様子を図 51(a) に示す。仰角 30 度～-60 度まで 10 度間隔で光線が入射している。イメージ面を拡大した様子を図 51(b) に示す。黒点は、それぞれの光束が収束している点を表す。また、双曲面ミラーを 2 枚使用した 2 枚ミラーシステムの全体の様子を図 52(a) に、そのイメージ面を図 52(b) に示す。図 51 と図 52 のイメージ面の様子を見比べると、2 枚ミラーシステムの方が、像面湾曲が抑制されていることが確認できる。

また、図 53 には像面湾曲の大きさのグラフを表す。図 53 より、シングルミラーの場合、視野角が大きくなるほど、指数的に像面湾曲が大きくなるのに対し、2 枚ミラーの場合は、視野角全域にわたり像面湾曲が低減されほぼフラットになっている。2 枚ミラーシステムの方がミラー 1 枚の従来のセンサより像面湾曲が 4% 程度に抑えられていることがわかる。

この結果をもとに 2 枚反射光学系の全方位ミラーを設計した。図 54 に設計した光学系を示す。今後、この設計をもとに試作を行い評価を行っていく。

表 7 二枚ミラーのパラメータ

	第一ミラー	第二ミラー
曲率半径	R = 40.856	R = 40.856
コニック係数	k = -1.6154	k = -1.6154

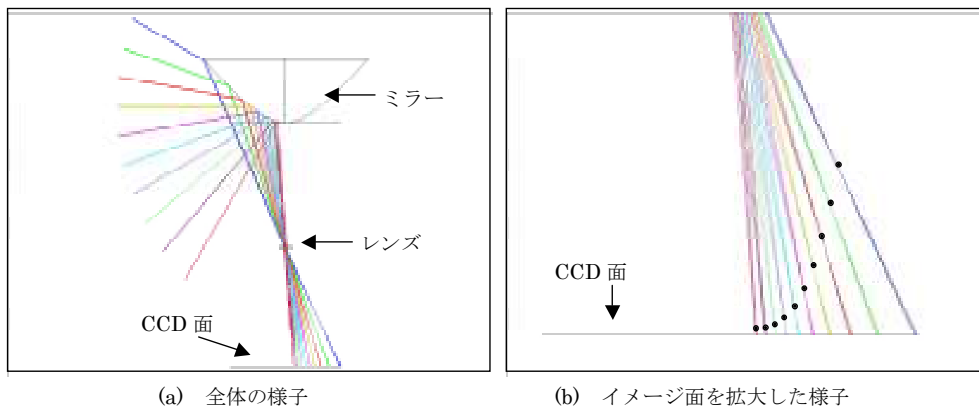
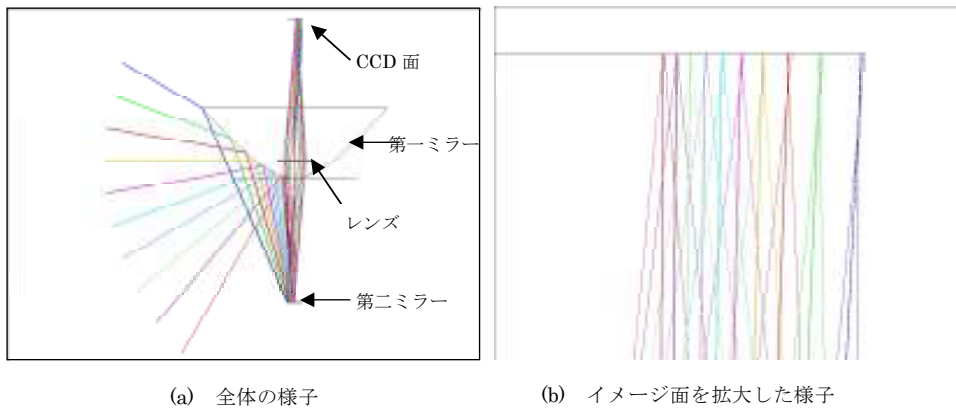


図 51 従来の全方位視覚センサ



(a) 全体の様子

(b) イメージ面を拡大した様子

図 52 2 枚ミラーシステム

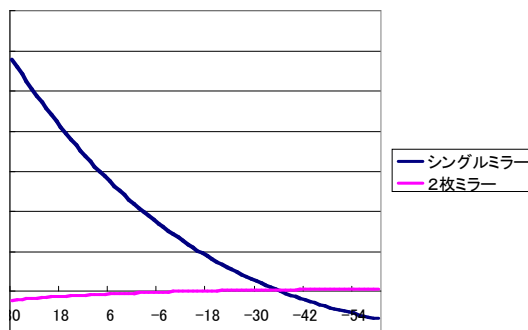


図 53 従来のシステムと 2 枚ミラーシステムの像面湾曲の大きさの比較

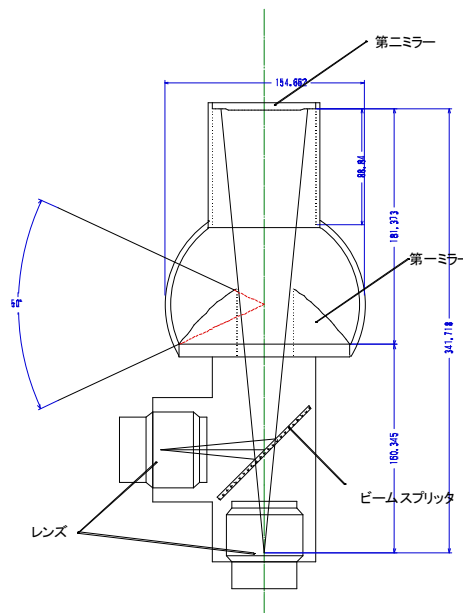


図 54 二枚反射全方位ミラー設計

#### 4-4 全方位高解像度リアルタイム動画像入力記録方式の開発

##### 4-4-1 複合センサカメラシステムを利用したリアルタイム動画像記録

ここでは、複合センサカメラで撮像した画像データを実時間で PC 上に取り込んで、ディスプレイに表示させつつ RAID システムに記憶させるソフトウェアを実装した。なお、開発には Visual C++ 6.0 (Microsoft 社) を用いた。

画像を取り込んで保存する際に、取り込みの終了を待ってから画像の表示、保存を行うと画像取

り込みの終了を待つ時間が無駄である。そこで、画像の取り込みから保存までの一連の処理を「ダブルバッファ処理」を用いて行う。ダブルバッファ処理は、画像の取り込みと、画像の表示、保存を並列に行うものであり、ダブルバッファを用いた時の処理と用いない時の画像2フレーム分の取り込み、保存処理の流れを図55に示す。具体的には、画像を取り込むためのメモリを2つ用意しておき、メモリ1で画像の取り込みを行い、それに並行してメモリ2に格納された画像データの表示、保存を行う。それぞれの処理が終了した後に、メモリ2で画像の取り込みを行い、それに並行してメモリ1に格納された画像データの表示、保存を行う。これらの処理を繰り返して行う。

ダブルバッファ処理を用いることで、次の画像の取り込みまでの待ちの時間が大幅に減り高速に取り込み処理が実現できるようになった。

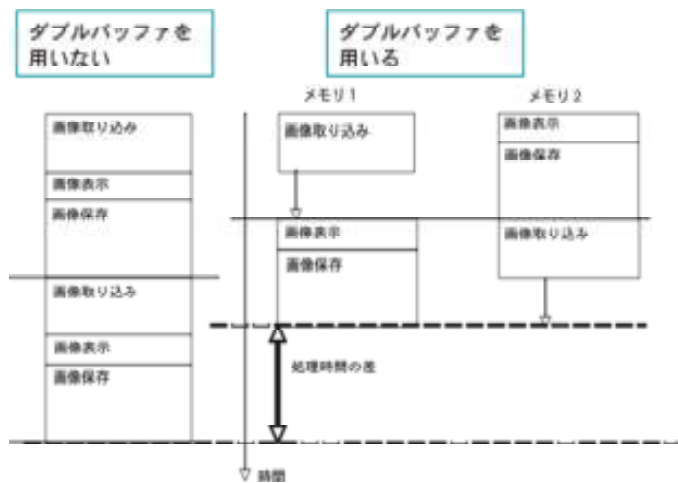


図 55 ダブルバッファ処理

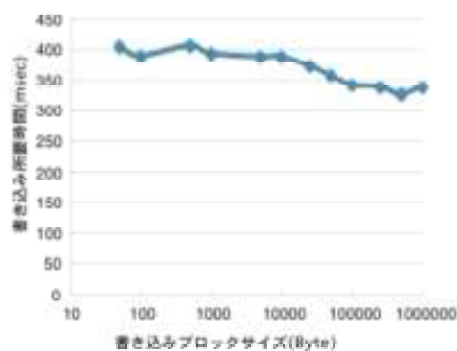


図 56 書き込みブロックサイズと所要時間

#### 4-4-2 書き込み速度の最適化

次に、書き込みブロックサイズについて考える。書き込みをする際にデータの転送速度等の関係上、データの書き込みブロックサイズを変えると書き込み速度も変わる。そこで、書き込み速度の最適化を行うために次に示す予備実験を行った。実験内容は30MbyteのデータをRAIDシステムのハードディスクに書き込む際に、書き込みブロックサイズを変化させていき、その時の書き込み所要時間を計測するというものである。低解像度高フレームレートカメラに関する結果を図56に示す。図56より、書き込みブロックサイズが500KByteの時に書き込み所要時間が短いことがわかる。この結果より書き込みブロックサイズを500KByteに決定した。この際の平均書き込み速度は91MByte/secである。

以上で述べたことを踏まえて、画像データを取り込み、保存、表示の一連の処理を実時間で処理するソフトウェアを開発した。

#### 4-4-3 パルスジェネレータの作成

本システムでは高解像度低フレームレートカメラと低解像度高フレームレートカメラという2台のフレームレートの異なるカメラを同期させて撮像する。高解像度低フレームレートカメラは4.29fps、低解像度高フレームレートカメラは30fpsである。そのため、1:7の割合で高解像度低フレームレート動画と低解像度高フレームレート動画を同期して撮像する。すなわち、図57に示すように高フレームレート画像列の1フレーム目と低フレームレート画像列の1フレーム目、高フレームレート画像列の8フレーム目と低フレームレート画像列の2フレーム目が同期するように撮像する。

2台のカメラの同期をとるために、パルスジェネレータを作成しそこから2台のカメラにパルス信号を送る。本システムで用いるカメラは4.29fpsと30fpsであるので高解像度低フレームレートカメラに4.29Hz、低解像度高フレームレートカメラに30Hzのパルス信号をそれぞれ送ることで2

台のカメラの同期をとる。

4.29Hz と 30Hz のパルス信号を出力するためのパルスジェネレータを Max+Plus II Baseline(Altera 社)というソフトを用いて設計した。設計した回路は、ヒューマンデータ社の CSP-001 という FPGA(Field Programmable Gate Array)評価ボードを用いて実装した

パルスジェネレータからは 30Hz と 4.29Hz のパルス信号を出力するので、まず FPGA 評価ボードのベースクロック信号を 30Hz にまで分周して、高フレームレートカメラにはそのまま 30Hz の信号を、低フレームレートカメラには 30Hz の信号に 7 進カウンタに通して 4.29Hz に分周したものを出力する。それぞれのカメラに送られたパルス信号は、カメラの取り込みボードに送られ、そこから取り込みトリガが生成され、カメラがトリガを検知して画像を取り込んでいく。これにより、高解像度低フレームレート動画像と低解像度高フレームレート動画像を 1:7 の割合で同期して撮像できた。

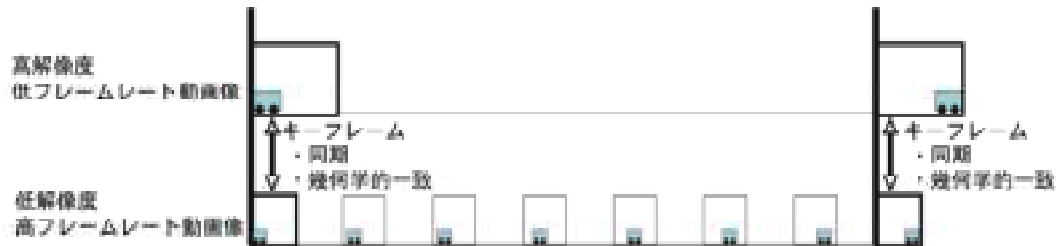


図 57 画像の同期

#### 4-5 高解像度動画配信ソフトウェアの開発

我々は図 58 に示すような複合センサカメラシステムより取得した全方位動画像に処理を施して高解像度高フレームレート動画像を生成し、ユーザが指定した部分のみを配信、表示するようなシステムの開発を目指している。本研究は、このシステムで用いられる各デバイスやリアルタイムでの動画像配信を考慮した上で、ユーザの視点情報に基づき動画像を受信し表示するためのクライアントを構築することを目的とする。動画像の伝送には MPEG4 を利用することを想定し、3D メッシュにテクスチャマッピングを行った仮想空間を生成することにより表示を行う。

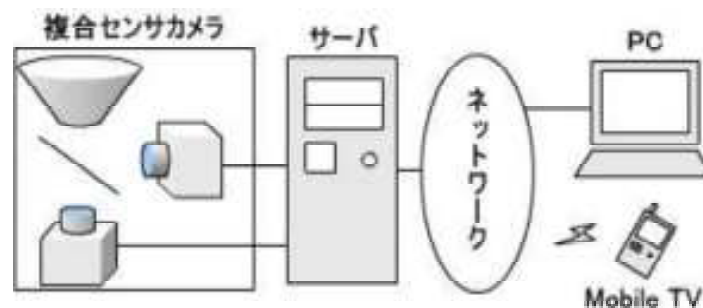


図 58 高解像度動画像配信システム

##### 4-5-1 視点情報に基づく動画像配信

この章では本研究の目的である視点情報に基づく動画像配信に関して述べる。(1)節では本研究が目標とする動画像配信の機能について述べる。次に(2)節では受信クライアント構築にあたり、想定しているシステムの全体像を示す。

##### (1) 問題設定

本研究は視点情報に基づき配信動画像の一部をインタラクティブに受信、表示するためのクライアントを構築することを目的とする。図 59 のように円筒状の仮想空間の中心に観測者が立っていると仮定し、クライアントからの視点情報に基づきサーバ側で表示部分を決定し、これをクライアントへ伝送する。そうするとユーザがあたかも 3 次元空間内でまわりを見渡しているかのように動画像を見ることが可能である。本研究ではこのようなインタラクティブな機能を有するクライアントの構築を目指す。

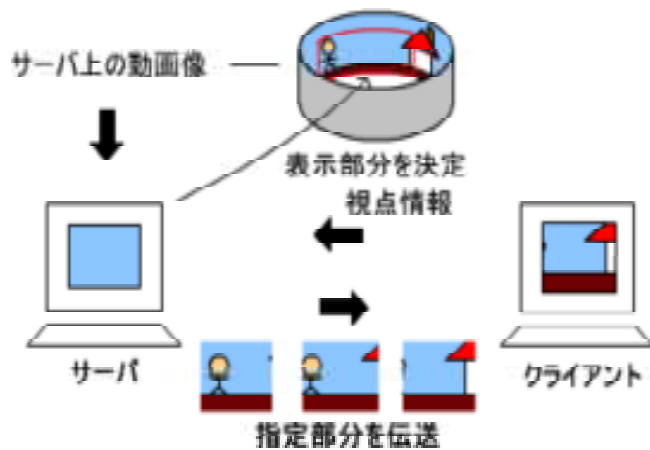


図 59 視点情報に基づく動画配信システム

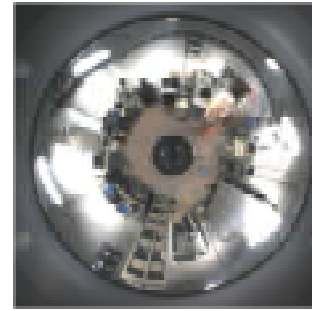


図 60 全方位カメラで取得した画像

## (2) 想定システム

クライアントを構築する上で配信に関わる部分を想定する必要がある。図 59 が想定システムの全体像である。本研究では動画像伝送用の圧縮規格として MPEG4 を用いる。一般に動画像はデータ量が大きく、保存やネットワークを通じた配信を行うために圧縮する必要がある。しかし、動画像の圧縮フォーマットは予測符号化、ビットストリーム変換、同期情報の付加、伝送のためのパケット分割など、非常に複雑な処理を要するため一から構築することは非常に困難である。既存の規格に沿いネットワーク伝送を行うことを想定した方が容易になることは明白である。とりわけ既存の規格の中でも、MPEG4 は一般的によく用いられている規格であり、またネットワーク伝送を前提とした規格であるため、本研究に最適な規格であると考え用いることにした。MPEG4 に関しては 4-5-2 節において詳細を述べる。

さらに本研究では配信動画像として全方位カメラ HyperOmni Vision で撮影した全方位動画像を想定している(図 60)。本研究ではこのような前提の下で動画像受信クライアントの構築を進めることとする。

### 4- 5- 2 MPEG4 規格の調査

#### (1) MPEG4 概略

一般的に動画像はデータ量が大きく、そのまま扱うには大きな保存領域が要求される。そのために動画像データは圧縮を施す必要性が生じる。この章で述べる MPEG4 は数ある圧縮方式の内の一つであり、この問題を解決する手段である。以下に MPEG4 の特徴を挙げる。

- ・ 動画像をオブジェクトに分割して扱う。
- ・ ネットワークを介した伝送を想定しており、圧縮率が非常に高い。
- ・ 様々なインタラクティブ性をサポートする。
- ・ 自然画像、合成画像の両方をサポートする。

MPEG4 は視覚情報の符号化を規定する Visual パート、音響情報の符号化を規定する Audio パート、これらを多重化組み立てを行う System パート、配送を規定する DMIF (Delivery Multimedia Integration Framework) などから成り立っている。また、MPEG4 ではデコード手順 (圧縮データ [ビットストリーム] を元に戻す) のみが定められておりエンコード手順 (動画像データを圧縮する) は定められていない。従ってエンコードは MPEG4 の規格に沿ったビットストリームを最終的に作り出せば、その過程はどのようであっても良い。ネットワーク伝送するためのビットストリームは、Visual 情報のビットストリーム、Audio 情報のビットストリーム、動画像全体構成のビットストリーム、同期情報のビットストリームなどの種類毎に圧縮したビットストリーム (MPEG ではエレメンタリーストリームと呼ばれる) を多重化することによって得られる。それぞれのビットストリームのデコードに関しては上記の各パートで規定されることとなる。また、この多重化を行った最終的なビットストリームのフォーマットを規定しているのが MP4 ファイルフォーマットである。



## (2) System

Systemパートでは動画像をオブジェクト単位で扱う仕組みと同期化の仕組みが規定されている。本研究では動画像の伝送の際に System で規定されている機能を利用することを想定している。

### BIFS

MPEG4 の System パートではオブジェクトの表示方法や特性を指定するためのシーン記述言語として BIFS(Binary Format for Scene)が規格化されている。BIFS は VRML を基に拡張したものである。VRML は 3D 空間を表現するための言語であり、視点、光源、形状、材質、座標などのノードをツリー状に配置することにより 3D 空間を表現する。BIFS も VRML と同様にツリー状に配置されたノード集合で表現される(図 61)。VRML との違いとして

- ・ リアルタイムで伝送されるビデオやオーディオ、2D シーンを表現するノードの追加
- ・ シーンを経時的に更新するためのシンタックスの定義
- ・ 効率的な伝送を可能にするためのシンタックスの定義

などが挙げられる。

BIFS では上記のように送信側からシーンを更新するために二つの方法を用意している。1 つは BIFS Command Frame と呼ぶものである。これはシーンを構成するノードやノードの構成要素(フィールド)の値などを更新するものである。もう 1 つは BIFS Animation Frame と呼ぶものである。これは特定のノードのフィールドを連続的に変化させるものである。両者の間には BIFS Command Frame が比較的静的なシーンの更新であるのに対し、BIFS Animation Frame はストリーミングデータとして伝送することを想定しているといった違いがある。

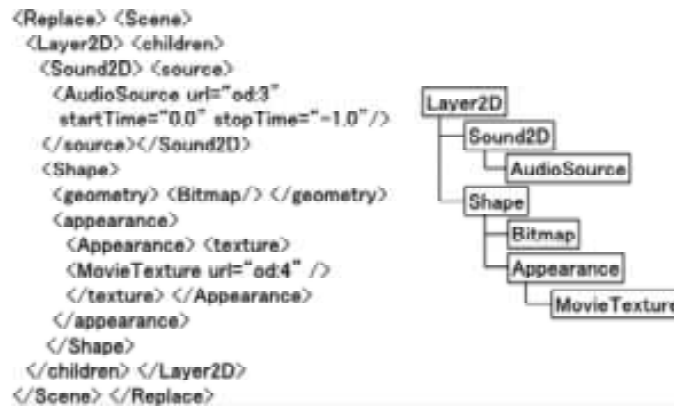


図 61 BIFS サンプル及び階層構造

### Object Descriptor

シーン記述中では、ビデオやオーディオストリームをオブジェクトディスクリプタ ID と呼ばれる 10 ビットの識別子を用いて指定する。オブジェクトディスクリプタはオブジェクトディスクリプタ ID(OD ID)で指定されたストリームを保持する役割を果たしている。図 62 にオブジェクトディスクリプタの全体構造を示す。

OD ID は前述したように BIFS からの参照値が入る場所となる。また、ES Descriptor は各ビットストリームの位置、各種設定情報を保持している。エレメンタリーストリーム ID(ES ID)がビットストリームを特定するための識別子であり、残りは設定情報である。Decoder Config Descriptor はデコーダ設定に関するディスクリプタである。ストリームの種類を判別するためのストリームタイプやデコーダに必要なバッファサイズ、ストリームの最大/平均伝送レートなどを記述する。SL Config Descriptor は同期情報設定に関するディスクリプタであり、伝送のためパッケージ化されたビットストリームのヘッダ構成などを記述する。QoS Descriptor はオプションなディスクリプタであり、ストリーム伝送に関する品質を記述する。IPI Descriptor はオプションなディスクリプタであり、著作権情報などを記述する。OCI Descriptor は残りのオプションな設定を記述する。例としてキーワードの付加や著作者の情報などの記述が挙げられる。

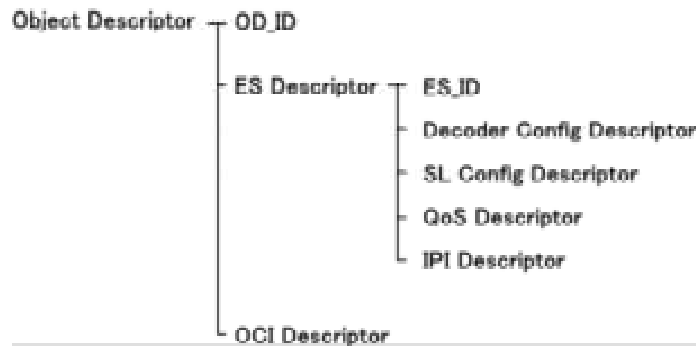


図 62 Object Descriptor の構成

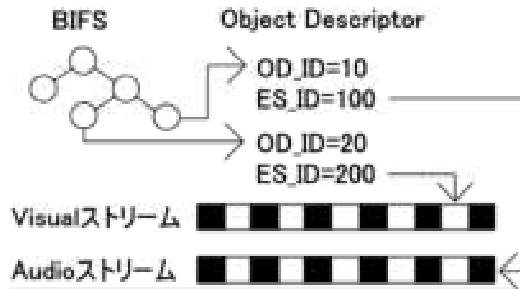


図 63 BIFS と ObjectDescriptor の依存関係

図 63 に BIFS と Object Descriptor の依存関係をまとめて示す。シーンを構成する BIFS の各ノードは OD ID によりオブジェクトディスクリプタを指し示す。さらに各オブジェクトディスクリプタは ES ID により伝送用の Visual, Audio ビットストリームを指し示す。これにより動画像の全体像が表現されることとなる。

### (3) Visual

Visual では MPEG4 で扱う動画像に関する規定を定めている。MPEG4 では自然画像（カメラで撮影した普通の動画像）と合成画像（コンピュータグラフィックスなどで作りだした動画像）の両方を扱うことができる。本研究では Visual パートで規定されている 3DMesh や View-Dependent Scalability の機能を用い、合成画像上に自然画像をマッピングすることによりクライアントを構築する。

#### 3DMesh

一般にディスプレイなどを用いた動画像の表示は 2 次元であるため、2 次元空間上で 3 次元空間を表現するための手段が必要となる。これを実現するのが 3D メッシュである。メッシュとは何らかのジオメトリを定義している頂点の集合と、面を定義しているインデックスのセットであり、各 3 次元物体は多数のメッシュによるポリゴンの集合体として表現される。さらに物体の質感を表現するために用いられるのがテクスチャマッピングである。テクスチャマッピングとはメッシュによって生成されたポリゴンの上に物体の質感をあらわす 2 次元画像を張り付けることを言う。また、この時貼り付けられる 2 次元画像のことをテクスチャと呼ぶ。

#### Scalability

スケーラビリティとは、一般的には既存のハードウェアやソフトウェア構成などを大幅に変更することなく、処理に対する要求の質的、量的変化に適応できる度合いのことを言う。とりわけ動画像の圧縮、配信においては画質と伝送帯域のトレードオフを吸収するための機能といった意味合いで用いられることが多い。スケーラビリティには主に

- ・ 時間的スケーラビリティ

フレーム数による動画像の質の制御。伝送帯域に余裕がある場合には伝送するフレーム数を増やしより滑らかな動画像を配信する。余裕がない場合は動画像として無理なく再生される最低限のフレーム数を伝送することにより質の制御を実現する。

- 空間的スケーラビリティ  
解像度による動画の質の制御. 伝送帯域に余裕がある場合はより解像度の高いフレームを送り画質を良くし, 余裕がない場合は最低限の画質が保たれるような解像度でフレームを送送することにより質の制御を実現する.
- 周波数スケーラビリティ  
動画に DCT やウェーブレット変換を行った際, 高波数領域は人が知覚しづらいといった特徴を利用した質の制御. 伝送帯域に余裕がある場合は高周波数成分の係数も伝送し, 余裕がない場合には低周波数成分の係数のみを伝送することにより質の制御を実現する.

などが挙げられる. MPEG4 ではこれらのスケーラビリティを利用することにより, 柔軟な動画配信を保証している.

### View-Dependent Scalability

View-Dependent Scalability はインターネット回線などを通じた配信を前提とした機能である. View-Dependent Scalability は観測者(ユーザ)が仮想 3 次元空間にいると仮定し, 観測者の 3 次元空間内での視点の情報に基づき動的に表示部分を変更するというインタラクティブな機能を提供している. 図 64 で示すようにクライアント側で指定された視点情報がサーバに伝送され, サーバ側では表示部分に相当する DCT 係数(テクスチャ情報)のみをクライアントに送る. クライアントではバッファに保存しておいたメッシュ情報と視点情報より表示すべきポリゴン部分を計算し, DCT 係数をデコードして得たテクスチャをポリゴンにはりつける. また, この表示部分とテクスチャ更新部分をループさせることにより, 視点情報に基づいた動画の配信が可能となる.

View-Dependent Scalability で重要となる仕様を以下に挙げる.

- 3次元仮想空間は3Dメッシュを用いて表現される. さらに空間内にユーザが存在すると仮定し, その視点の方向により表示部分(ユーザーの見る事ができる部分)が決定される.
- クライアント(ユーザ側)からサーバへは視点情報のみが伝送される. 図 65 に視点情報を保持するクラスを示す.
- サーバからクライアントへは視点情報に基づき必要な部分のテクスチャの DCT 係数が伝送される. また伝送の効率を上げるためテクスチャ情報は DCT ブロック単位で切り出す必要がある.

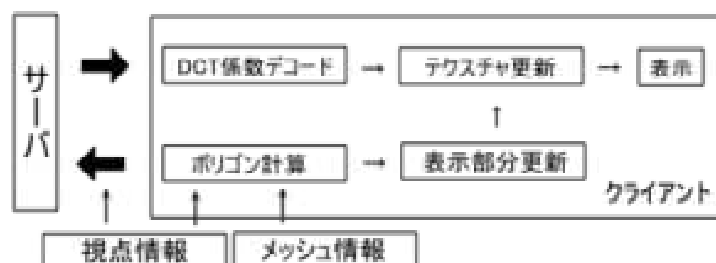


図 64 View-Dependent Scalability の全体構造概念図

### 4-5-3 受信クライアントの構築

この章では実際の実装手順を、まず(1)節で実装方法を述べた後、(2)節で実装した結果を示す。

#### (1) 実装方法

MPEG4 の View-Dependent Scalability を利用しテクスチャ情報が伝送されたという仮定の下, VRML を用いて全方位画像を視点情報に基づいて表示する. VRML は 3 次元グラフィクスを記述するための言語であり MPEG4 と互換性がある. VRML ではマッピングするテクスチャ 1 つ 1 つの各頂点座標(2次元)とそれに対応する 3D メッシュの各頂点座標(3次元)により 3次元物体を表現する. テクスチャ頂点とメッシュ頂点の対応を視覚化したものが図 66 である. これらの対応を VRML で定められた文法に基づきテキストで記述する. 対応関係の記述された VRML ファイルは既存の VRML プレイヤーを用いて表示することが可能である. 従って, 実装はテクスチャの頂点とそれに対応するメッシュの頂点の座標を算出し, VRML の文法に従い対応関係を記述するという手順になる.

```

class ViewDependentObject {
  unsigned int (32) View_dep_object_start_code;
  unsigned int (16) Field_of_View;
  bit (1) Marker_bit;
  unsigned int (16) Xsize_of_rendering_window;
  bit (1) Marker_bit;
  unsigned int (16) Ysize_of_rendering_window;
  bit (1) Marker_bit;
  unsigned int (32)* NextStartCode;
  while (NextStartCode == view_dep_object_layer_start_code){
    ViewDependentObjectLayer vdel;
    unsigned int (32)* NextStartCode;
  }
}

class ViewDependentObjectLayer() {
  unsigned int (32) View_dep_object_layer_start_code;
  unsigned int (16) Xpos1 ;bit (1) Marker_bit;
  unsigned int (16) Xpos2;bit (1) Marker_bit;
  unsigned int (16) Ypos1;bit (1)Marker_bit;
  unsigned int (16) Ypos2;bit (1) Marker_bit;
  unsigned int (16) Zpos1;bit (1) Marker_bit;
  unsigned int (16) Zpos2;bit (1) Marker_bit;
  unsigned int (16) Xaim1;bit (1) Marker_bit;
  unsigned int (16) Xaim2;bit (1) Marker_bit;
  unsigned int (16) Yaim1;bit (1) Marker_bit;
  unsigned int (16) Yaim2;bit (1) Marker_bit;
  unsigned int (16) Zaim1;bit (1) Marker_bit;
  unsigned int (16) Zaim2;
}

```

図 65 視点情報保持クラス

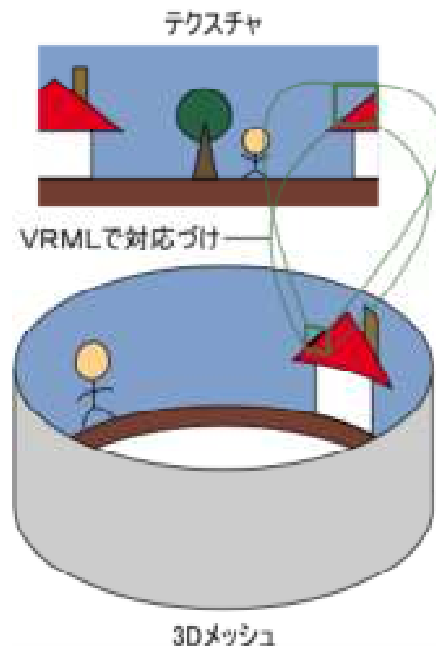


図 66 VRML によるテクスチャと 3D メッシュの対応付け

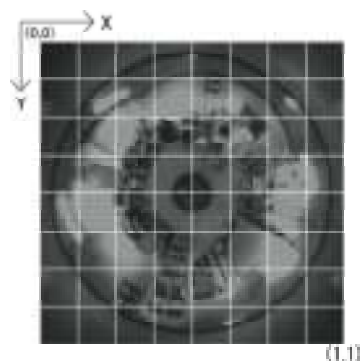


図 67 テクスチャ頂点の決定

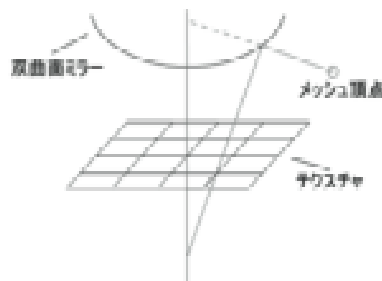


図 68 メッシュ頂点の決定

表示までの手順を順を追って説明する. まず, テクスチャの頂点座標を決定する. テクスチャの座標は左上隅を (0, 0), 右下隅を (1, 1) とした 2 次元座標で表される. テクスチャは伝送効率を高めるため DCT ブロック単位で切り出すことが View-Dependent Scalability で定められているので, これを考慮にいれ格子状にテクスチャの頂点を決定していく (図 67).

続いて決定されたテクスチャ座標に対応する 3 次元空間上の頂点を 4-3-1 節で述べたミラー形状を考慮し算出する (図 68). 求められた頂点を VRML で記述することによりプレイヤーで表示を行う.

## (2) 実装結果

頂点を算出し VRML でマッピングを行った結果を示す (図 69). 球状 (お椀型) メッシュに全方位画像がテクスチャとして貼り付けられていることが見て取れる. 視点は双曲面の焦点の位置とし, この球状メッシュの中心からまわりを見渡すことができる (図 70).



図 69 ポリゴンへのテクスチャマッピング 図 70 視点方向による見え方の違い

#### 4- 5- 4 配信サーバの構築

同一視野を持つ2種類の全方位カメラで撮像した動画像をクライアントで合成するために、動画像配信サーバを構築する必要がある。現在、動画像の転送モードは、図 71 に示すような、以下の2通りを考えている。また、図 72 にシステムブロックを示す。

- 1) 指定した視野の高解像度全画面配信モード
- 2) 高解像度部分画像の配信モード

クライアント側は画像指定プロトコル(RTSP)を利用して、サーバに視点情報を送信し、サーバは視点情報により対応する動画像を映像配信プロトコル(RTP)により、クライアントに配信する。本研究を達成するために、主に通信プロトコルのデザインと配信サーバの動作規定を決定するべく調査を行なった。

- 1) RTSP による画像指定プロトコルのデザイン

基本的考え方は、RTSP の SET PARAMETER 手続きを利用して、サーバに視点情報を指定する。

例：C->S: SET\_PARAMETER rtsp://example.com/foo RTSP/1.0

Cseq: 421

Content-length: 20

Content-type: text/parameters

視点情報変数: value

- 2) RTP, RTCP による画像配信プロトコルのデザイン

RTP Payload Format for MPEG4 に基づいて、MPEG4 の Payload のパケット分割規則を定義する。どのように、MPEG4 ストリームを分割し、RTP パケットにマッピングするかを検討する。

- 3) クライアント要求によるサーバの動作規定

主に、Darwin Streaming Server のオープンソースをベースにし、クライアントの要求による内部処理を規定することで、ストリーミングサーバの配信部を設計する。

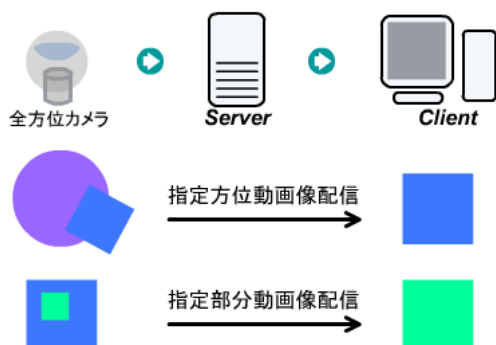


図 71 サーバのイメージ図

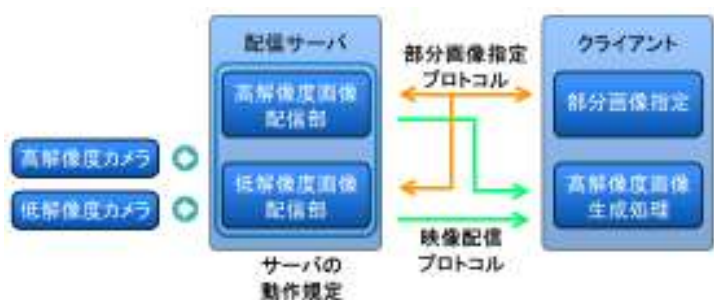


図 72 システムブロック

## (1) 動画配信システムの構成

大きく分けるとシステムは、ビデオアーカイブ、セッションコントロール、トランスポートコントロール、ビデオデコーダディスプレイの4つからなる（図 73 参照）。

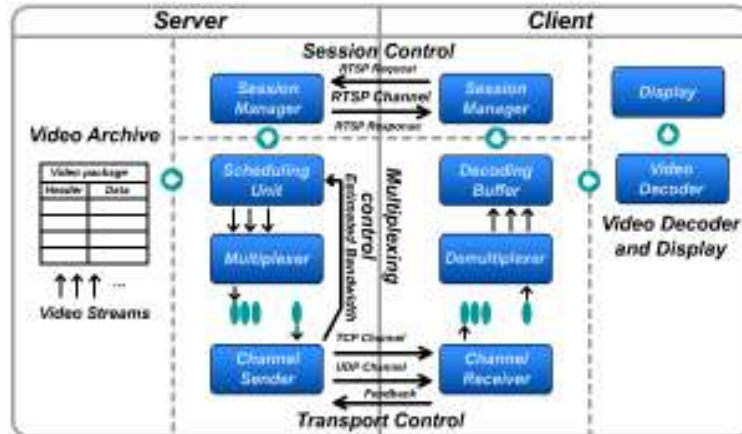


図 73 配信システムの構成図

動画配信は4ステップからなる。

- 1) クライアントはセッション管理ユニットを通じて、サーバーにセッションの開始を要求、サーバの方は許可するかどうかをクライアントに返答する。
- 2) 許可する場合、サーバのセッション管理部はスケジューリングユニットに通知する。そして、ビデオアーカイブからのビデオパッケージは一定のスケジューリングにより、多重化処理されてチャンネル送信部に送られる。
- 3) 送信部からクライアントの受信部にビデオデータを転送する。また、クライアントから転送状況やネットワーク状況などをサーバにフィードバックしてもらう。サーバはこのフィードバック情報により、通信帯域を推定し転送ビットレートを制御する。
- 4) クライアントは受信したデータをデコードして再生する。

表 8 各種ストリーミングサーバの比較

Server	Windows Media	Real Server	Darwin Streaming Server
プラットフォーム	Windows NT Windows 2000	Windows NT Linux, Solaris	Mac, Linux FreeBSD Windows 2000 Windows NT
費用	無料	有料	無料
画質	良い	良い	最も良い
交互性	良くない	一般	最も良い
RTSP 支持	MMS	支持	支持
オープンソース	なし	なし	あり

## (2) Apple Darwin Streaming Server について

表 8 に各種ストリーミングサーバの比較を示す。この比較により、本研究では、オープンソースの Darwin Streaming Server をベースとして開発を行なう。Darwin Streaming Server には以下のような特徴がある。

- 1) MPEG4 Streaming を実現
- 2) 画質を改善：Apple skip-protection technology
- 3) Web-based management

- 4) メディアアクセス認証
- 5) Instant-On 技術：動的に配信画像の質を調節
- 6) 業界標準のサポート：RTSP, RTP, RTCP, SDP
- 7) モジュールベースによる機能の実現

サーバは起動直後、dynamic module を読み込む。つまり、サーバのモジュール機能の代わりに自分が書いたモジュールを dynamic module として置換えることが出来る。サーバの起動の流れを図 74 に示す。

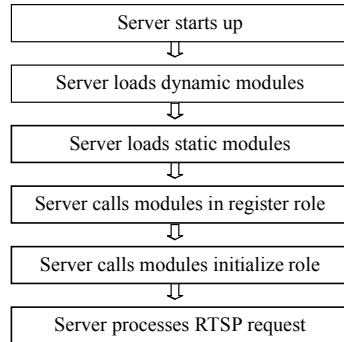


図 74 サーバの起動ステップ

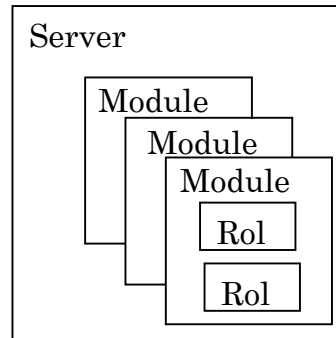


図 75 サーバとモジュール，ロールの関係

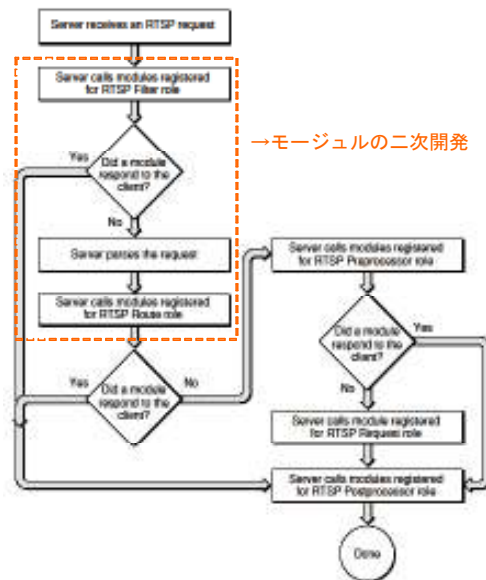


図 76 RTSP の処理プロセス

また、サーバはモジュールにより各種の機能を実現する。モジュールは Role に規定される機能を提供する。本研究では、RTSP 要求処理に対応するモジュールを改良することにより視点情報の処理を実現する。図 75 にサーバとモジュール，role の関係を示す。

RTSP の処理モジュールについて説明する。

#### 1) RTSP Filter Role

qtssRTSPReqFullRequest をフィルタリングして変更することが出来る。例えば、あるリクエストに対して、movie のあるフォルダー /a/a.mov を /b/b.mov に変更することができる。

#### 2) RTSP Route Role

属性 qtssRTSPReqRootDir を変えることが出来る。例えば、各国語に対応する場合に各言語のフォルダーに変更する。

#### 3) RTSP Preprocessor Role

リクエストを処理する。各モジュールはリクエストに応答するかどうかを判断する。応答する場

合には、QTSS-Write を使って、クライアントにデータを転送、クライアントに適切な応答を送る。

#### 4) RTSP Request Role

RTSP Preprocessor Role ではリクエストを処理できない場合に、このロールで処理する。QTSS-AppendRTSPHeader, QTSS-SEndRTSPHeaders を使って、クライアントに RTSP レスポンスを送る。QTSS-AddRTPStream, QTSS-Play を使って、クライアントに RTP パケットを転送する。

#### 5) RTSP Postprocessor Role

タスクの統計処理を行なう。

### (3) 今後の予定

今後、視点情報の処理を行なうため、RTSP Filter Role と RTSP Route Role に関するモジュールを改良することで、リクエストを分析し、要求された動画をクライアントに配信するサーバを構築する予定である。

### 4-5-5 システム評価実験

3D メッシュを利用した動画の表示ではポリゴン数が多ければ多いほど線形補間による歪みがなくなり画質が良くなる、しかしポリゴン数が増えると処理にかかる計算量が増えるため表示が遅くなる。すなわち表示にかかる計算量と画質の間にはトレードオフが存在する。以下では将来の動画ライブ配信を見越し、人間が動画を見た時に違和感を感じない程度に歪みを抑えつつ計算量をなるべく減らすため、ポリゴン数の違いによる歪みの違いについて主観的評価を行った結果を示す。

#### メッシュポリゴン数と画質の主観評価

View-Dependent Scalability ではテクスチャを DCT ブロック単位で切り出す必要があるので、1024×1024 ピクセルのテクスチャを 8×8, 16×16, 32×32, 64×64 でそれぞれ切り出しポリゴンにはり付けた場合の歪み具合を比較した。

表9 評価結果

		切り出しサイズ			
		8×8	16×16	32×32	64×64
評価の値	1	0	0	0	10
	2	0	0	7	0
	3	4	6	3	0
	4	6	4	0	0
評価の平均		3.6	3.2	2.3	1

比較はテクスチャをはりつけた画像を被験者にそれぞれ見せ、4段階で主観的評価を行ってもらったという方法をとった。評価は値が大きい程高いとする。すなわち4が最も歪みが少ないと感じたことをあらわし、1が最も歪みが多いと感じたことをあらわす。また歪みの差が感じられない場合は同じ値を答えても良いこととした。これは実際のポリゴン数が違う場合でも目で見て歪みの差を感じていない場合があるのか判断するためである。画面のサイズが1024×768のノート型PCに図93の画像をそれぞれフルスクリーンで表示し主観評価を行った。表示にはVRMLを表示するIE(Internet Explorer)のプラグインblaxxunを用いた。評価結果を表9に示す。評価は10人に行ってもらい、表の中の数字は各評価をつけた人数を示す。実験結果より64×64で切り出した時は明らかに歪みが見て取れると判断できる。また32×32ピクセルも他と比較を行えば歪みを感じるようであることがわかるが、明らかに64×64よりは歪みが少ないと感じていることもわかる。8×8 16×16での切り出しになると、ほぼ同じような評価がされていることより人間の目ではさほど差を感じない程度まで歪みが抑えられていることがわかる。



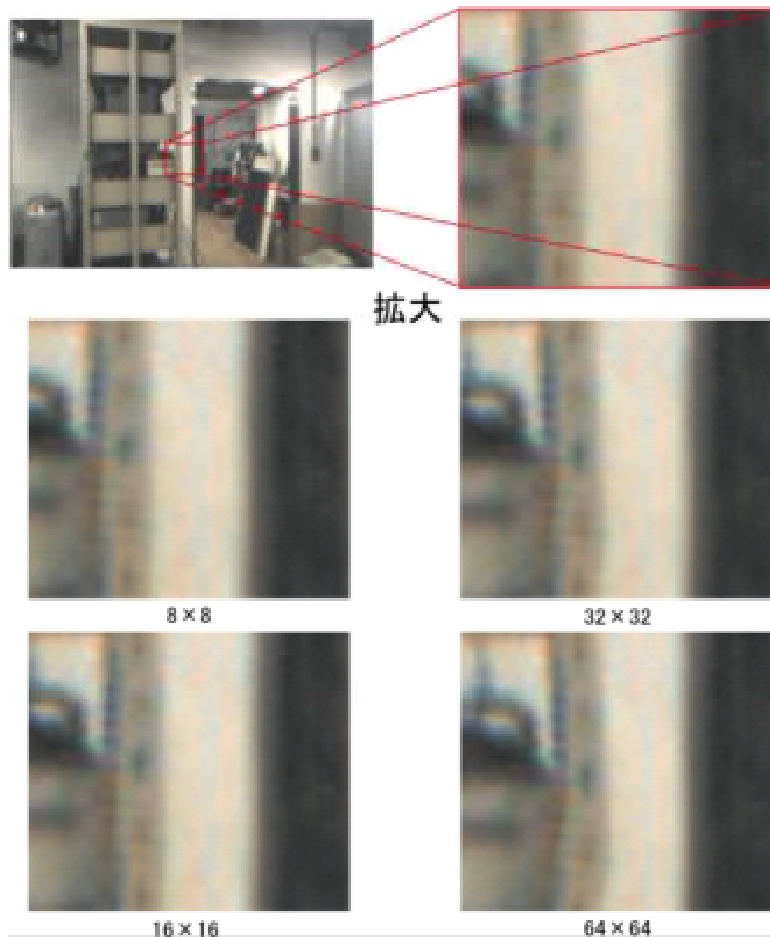


図 77 評価に用いた画像（拡大図）

#### 4-6 ライブ入力，配信システムの開発

動画の配信実験を行うための実験システムの構築を行った。図 78 に本年度構築を行った，システムの概要を示す。実験システムでは，配信サーバにはサーバ OS としての安定性や配信ソフトウェア開発の観点から Linux システムを，クライアントにはあらゆるネットワーク環境やトラフィック負荷を想定して，様々なアーキテクチャや OS (Windows、Linux、MacOS) を混在した形でのシステム選定を行った。また，配信コンテンツは映像蓄積用 RAID に保存され，Ultra SCSI160 インタフェイスを通して，配信サーバと接続される。

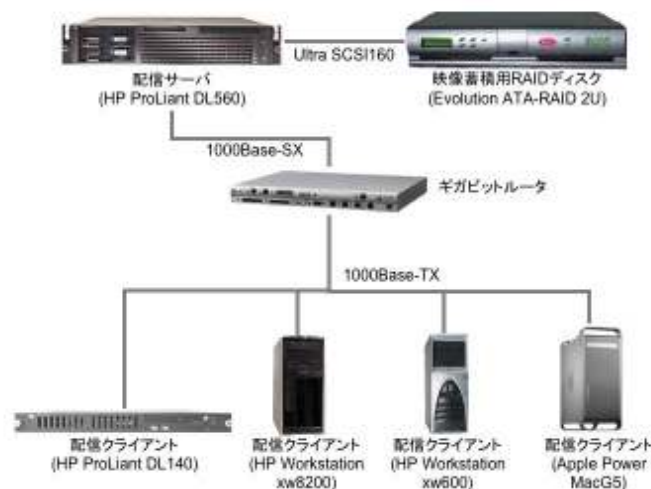


図 78 配信実験システム

配信サーバは 1000Base-SX によりギガビットスイッチングルータを通して接続され，個々のクライアントに対しては，1000base-TX で接続される。実験検証のために，配信サーバに QuickTime Streaming

Serverを導入して、通常のスリーミング配信環境を構築した。このシステムを用いて、配信時のネットワークトラフィックの変化や効率の評価を行っている。今後は、この配信システムを、高解像度低フレームレートと低解像度高フレームレート動画の多重配信に拡張して、評価を行っていく。

また、現在ソフトウェアで行っている高解像度画像と低解像度画像の幾何・光学補正をハードウェアで行うために、FPGA 画像処理ボード(Tsunami: SBS Technologies)を導入した。FPGA 画像処理ボードは Altera 社の FPGA, Stratix II を搭載し、PCI を通して PC と Cameralink インタフェースを通してカメラと通信する形をとる。本ボードを用いて、低解像度カメラの取り込みを部分の回路を実装し、画像取り込みのテストを行った。また、従来、ソフトウェアで行っていた画像補正アルゴリズムをハードウェア実装を考慮して検討や変更を行っている。今後、このボード上の FPGA に画像補正アルゴリズムをハードウェア回路として実装することで、リアルタイムでの画像補正を実現する予定である。

#### 4-7 総括

複合センサカメラの試作では、レンズ主点(投影中心)を合わせることにより、同一の視野で時空間分解能の異なる2種類の動画; a)高解像度(4000×4000画素)だが時間的には粗い動画(1フレーム/秒)と、b)通常の解像度(640×480画素)だが時間的に密な動画(30フレーム/秒)を同時撮像できるセンサカメラを開発している。今年度は、市販の高解像度低フレームレートカメラ(4008×2672画素、4.29フレーム/秒)と低解像度高フレームレートカメラ(1008×1018画素、30フレーム/秒)を用いた複合センサカメラを試作した。本試作センサは1:7の割合で2種類の動画を同期撮像できる。この試作したカメラを実際に利用して屋内外での撮像実験を行った。

リアルタイム高解像度動画の作成では、試作した複合センサカメラから得られる2つの動画から、空間的にも高解像度で、しかも時間的にも密な動画(4000x4000画素、30フレーム/秒)を作成する技術を確認した。時空間解像度の異なる2つの画像系列(高解像度低フレームレート動画、低解像度高フレームレート動画)の情報を統合する手法として周波数アプローチとモーフィングアプローチの2つの手法を考案し、それらのソフトウェアを実装した。

高解像度全方位カメラ用光学系の設計では、複合センサカメラ用に全方位ミラーを設計し、全方位複合センサカメラを試作する。今年度は、試作した複合センサカメラと単一双曲面の全方位ミラーを用いて、全方位複合センサカメラを構成した。

全方位高解像度リアルタイム動画入力記録方式の開発では、複合センサカメラから得られる2つの全方位動画から、空間的にも高解像度で、しかも時間的にも密な全方位動画(4008×2672画素、30フレーム/秒)を、入力および記録する技術を開発する。今年度は、複合センサカメラの2つの出力を、取りこぼしなくリアルタイムで同期記録するシステムを構築した。記録システムは、画像取り込みボードを搭載したPCワークステーションおよび、RAIDドライブにより構成される。また、取り込みと保存をスレッドに分け、並列処理を行うことにより、フレームの取りこぼしなしに画像を記録するソフトウェアを開発した。

高解像度動画配信用ソフトウェアの開発では、蓄積された2種類の全方位画像を圧縮してインターネットにより複数の閲覧装置(クライアント)に配信するサーバシステムと、配信された2種類の全方位動画から閲覧者側において高解像度かつリアルタイム(30フレーム/秒)の動画を作成する技術を開発する。また、全方位画像のうち、閲覧者の要求する視野の部分画像のみを切り出しそれらを能率良く配信し、クライアント側で動画表示するシステムを実現する。今年度は、MPEG4の規格調査と、Streaming Serverを利用した配信サーバの構築、およびVRMLを利用した全方位提示クライアントの試作を行った。

ライブ入力、配信システムの開発では、配信実験に用いる配信システムの構築を行った。また、ライブ入力のために現在ソフトウェアで行っている画像補正のハードウェア化による高速化を想定し、FPGAの画像処理ボードを導入した。

## 5 参考資料・参考文献

### 5-1 研究発表・講演等一覧

- [1] 渡邊清高, 岩井儀雄, 長原一, 谷内田正彦, “時空間周波数の異なる画像列からの高解像度動画像の合成”, 情報科学技術レターズ, vol. 3, no. LI-004, pp. 169-172, Sep. 2004.
- [2] 間下以大, 岩井儀雄, 谷内田正彦 “円錐曲線を用いた全方位視覚センサのキャリブレーション”, 画像の認識・理解シンポジウム(MIRU2004), I315-I320, 函館, Jul. 2004.
- [3] 間下以大, 岩井儀雄, 谷内田正彦 “ミラーの外縁を利用した全方位視覚センサのキャリブレーション”, 情処研報 CVIM, No. 147-, pp. 21-28, 京都, Jan. 2005.
- [4] 重本倫宏, 星川章, 長原一, 岩井儀雄, 谷内田正彦, 田中紘幸, “時間的・空間的分解能の異なる複合センサカメラシステム”, 情処研報 CVIM, No. 147, pp. 83-90
- [5] 松延徹, 長原一, 岩井儀雄, 谷内田正彦, 田中紘幸, “モーフィングによる高解像度高フレームレート動画像の生成”, 信学技報 PRMU, No. 104, No. 573, pp. 85-90 (PRMU2004-178), Jan. 2005.