

全方位画像に対応したストリーミング再生プレイヤーの開発 Developing Streaming Player for Hyper Omni Vision

上谷 巧[†] 西尾 佳祐[†] 岩井 儀雄[†] 長原 一[†] 谷内田 正彦[†] 鈴木 俊哉[‡]
Takumi Uetani Keisuke Nishio Yoshio Iwai Hajime Nagahara Masahiko Yachida Toshiya Suzuki

概要

動画像のストリーミング配信技術が向上するにつれて、クライアントが高解像度の映像を自由な視点で見られる動画像への需要が高まってきている。本論文では360度の情報を一度に獲得できる全方位視覚センサ [1] を用いて高品質な動画像を生成し、視点情報に基づいて動画像を配信するシステムについて述べる。本研究ではネットワークを通じて視点情報をサーバとやり取りし、ユーザが見たい領域をストリーミング再生するプレイヤーを作成する。また画像サイズや変換の計算量とフレームレートに関して評価実験を行い、実際のシステムにおけるストリーミング配信の実現可能性について検討する。

1. はじめに

近年、ブロードバンドの普及やパーソナルコンピュータの性能向上に伴いリアルタイムの動画配信が可能となってきた。インターネット環境もADSL(Asymmetric Digital Subscriber Line)からFTTH(Fiber To The Home)へと移行しつつあり、より大容量のデータをより高速に転送できるようになってきた。これにより、高解像度の動画配信だけでなく、インターネットのインタラクティブ性(双方向性)を活かした動画配信システムのニーズが高まってきている。

我々は、複合センサカメラシステム [2, 3] により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の二つの全方位動画像を取得し、この二つの動画像から高解像度高フレームレートの動画像を生成する方法を提案している [4, 5]。生成された高解像度高フレームレートの全方位動画像は既存のプロトコルで配信するにはデータ量が大きすぎるので、クライアントが指定した部分のみを分割して複数の伝送路により多チャンネルで配信し、表示するシステムを提案する。本研究ではこのシステムにおいて、クライアントが見たい部分を簡単に指示でき、それを提示できるようなアプリケーションの開発を目指す。

2. 想定システム

2.1 全方位高精細動画配信システム

本研究で目標とするシステムの概略図を図1に示す。全方位視覚センサによって撮像された高解像度低フレームレートの動画像と低解像度高フレームレートの動画像の二つの動画像から、高解像度高フレームレートの動画像を生成する。生成された高解像度高フレームレートの動画像はサーバに保存されクライアントに配信される。ここでクライアントに配信する際において、クライアントが見たい領域のみを分割して多チャンネルで配信する。クライアント側では、多チャンネルで配信された複数の

部分画像を合成して一つの高解像度高フレームレートの全方位動画像を作成し、それを幾何変換によって透視投影画像に変換してクライアントが望む領域を出力する。クライアントが別の視点を見たい場合は、視点変更情報をサーバ側に伝え、サーバはそれに対応した部分画像を再び多チャンネルで配信することによってクライアントが見たい部分を見られるようなインタラクティブな動画配信システムを構築する。

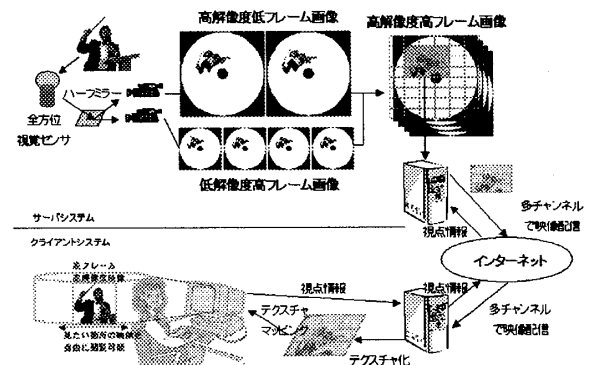


図1: システム概略図

2.2 インタラクティブ配信

本節では、全方位動画像のインタラクティブ配信の有用性について述べる。図2に従来の配信システムと提案システムの比較を示す。低解像度の全方位動画像配信は、従来の配信システム [6] で困難なく実現できるが、解像度とフレームレートが高くなると、必要な通信帯域が増大する。インターネットの通信帯域は有限であるため、クライアントがサーバに視点情報を渡すことによって、ユーザの現在見ている方向の高解像度動画像だけを部分配信する。このサーバ・クライアント間のインタラクティブ配信によって、通信帯域を節約することができる。

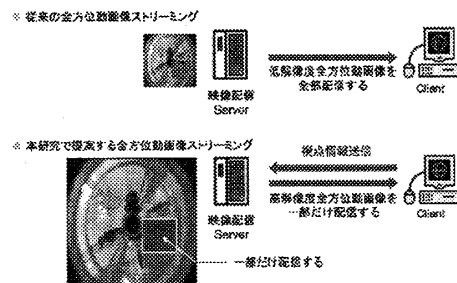


図2: 従来のストリーミングサーバとの比較

[†]大阪大学大学院基礎工学研究科
[‡]株式会社映蔵

2.3 多チャンネル化

2.2節で述べたように、サーバからはユーザの見たい方向の高解像度動画が部分配信される。本研究で想定する全方位高精細動画の解像度は4096×4096画素である。しかし、既存の通信プロトコルでは1チャンネルあたりに配信できる情報量に限界がある。例えば、現在インターネット配信などに多く利用されているMPEG-4でも、メインプロファイルのレベル4での最大解像度はHDTV(1920×1088画素)に規定されている[7, 8]。1チャンネルではそれ以上の高解像度な部分動画を配信できない。

提案する多チャンネル動画配信システムの通信の概略を図3に示す。全方位高精細動画を複数の低解像度動画に細かく分割して、配信サーバの動画データベースに蓄積する。サーバはクライアントに要求された視点情報に対応する低解像度動画を選択して、その複数の動画を多チャンネルで同時にクライアントに転送する。本研究では、サーバから配信される全方位動画を多チャンネルで受信するために、通信プロトコル(RTSP)を改良し、クライアントの通信モジュールを構築する。

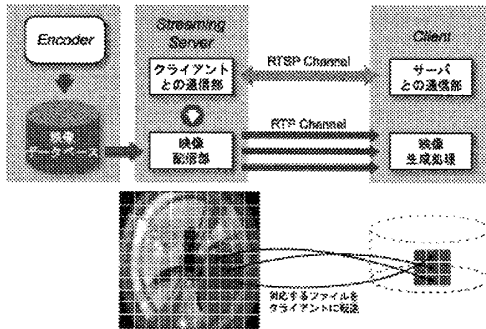


図3: サーバ・クライアント間通信概略図

3. システム設計

この章では想定システムで実際に動画を配信するために、具体的なプロトコルの改良とクライアント側のプレイヤーの開発について述べる。

3.1 プロトコルの改良

多チャンネルインタラクティブ配信[9]を実現するためにRTSPを改良する。RTSP(Real Time Streaming Protocol)は、RealNetworks社、Netscape Communications社、Columbia大学が共同で策定した。IETFによりRFC2326として勧告されている[10]。本研究において拡張したRTSPの流れを図4に示す。図4で点線の部分が改良点で以下の通りである。

• OPTION

一般的なストリーミングサーバは、クライアントが受信したいファイルを指定してOPTION請求することにより、サーバの対応するRTSP Methodをクライアントに回答するが、提案する拡張プロトコルではクライアントは受信したい全方位動画の分割画像が格納されたディレクトリを指定する。

• DESCRIBE(全体的な記述情報)

一般的なストリーミングサーバは、クライアントのDESCRIBE請求により、クライアントが受信したい単一のメディアファイル記述情報をクライアントに回答する。提案する拡張プロトコルでは、まず全体的な記述情報をサーバから得る。記述情報の中には、全方位センサのパラメータ、全方位動画を構成する分割された全ての低解像度動画のファイルリスト、各低解像度動画の位置とサイズ情報が記述されている。

• SET_PARAMETER

提案する拡張プロトコルでは、クライアントは全方位動画の見たい部分の視点情報を、SET_PARAMETERを介してサーバに送る。サーバはその視点情報をもとに対応するファイルを選択する。クライアントはそのファイルリストをサーバから得る。

視点方向が変更された場合、新しく再生するファイルは今まで再生していたファイルと時間の同期をとる必要がある。提案システムではサーバから多チャンネルで同時にRTPパケットを受信するので、同じパケット番号のRTPパケットをクライアント側で受信できる。サンプル時間が同じであれば、各ストリームのパケットが同時にクライアントに転送されるのでクライアント側では同時刻の動画を復元することができ同期が保証される。

現在は1チャンネルでの1ストリームの受信は成功しており、多チャンネルは今後の課題である。

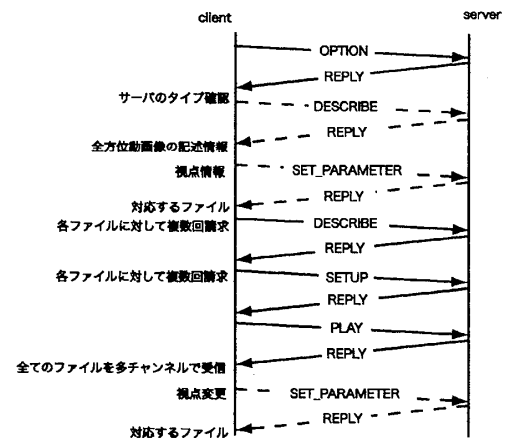


図4: 提案システムのRTSPの流れ

3.2 クライアントの設計

クライアントが見たい領域を自由に見られるようなプレイヤーを開発するために、本研究ではMPlayer[11]をベースプログラムとした。この節では目的のプレイヤーを開発するためにMPlayerに加えた改良について述べる。

MPlayerはOpenGL[12]のモジュールをサポートしているので、GLのテクスチャマッピング機能を用いて全方位動画を透視投影変換して表示するようにした。テクス

チャマッピングとは画像を格子状に切り出し格子点のみを変換に応じてマッピングを行い、格子点間は線形補間によって近似変換する手法であり、全方位画像を透視投影画像に変換するような非線形変換に対して有効である。

また多チャンネル配信によって配信された複数ファイルを1つの画面で再生するために、複数のファイルを読み込んで同時に再生できるように改良した。ストリーミング配信をする際、サーバから再生ファイルリストを受け取るようにプロトコルを設計するのでこの機能は必要である。この流れは図5のようにになっている。モニタに出力する前にすべての動画列をデコード・テキストチャ貼付けを行い表示することにより複数ファイルの同時再生を可能にした。アルゴリズムは各ファイルからフレームをテキストチャに貼付け、全て貼付け終わってから表示するのでファイル間の同期は保たれる。

MPlayerは様々なファイル形式に対応しているが現在では実験で用いているQuickTime形式にしか対応させていない。また同じサイズの複数ファイルを同時に再生することは出来るが、異なったサイズのファイル同時再生は今後の課題である。

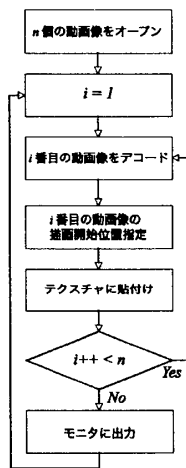


図5: 複数ファイル再生

4. 評価実験

この章では、動画再生の各プロセスでの実行時間を計測し、多チャンネルでストリーミング配信したときの推定フレームレートを導出する。そして、提案システムでのストリーミング配信可能性を検討する。

4.1 実験方法

画像サイズやビットレートの異なる動画列を用いて、実行時間の計測を行う。画像サイズは720×480を基準サイズとし、その1/4倍～3倍までで変更した。ビットレートは90kbpsを基準ビットレートとし、45kbps～360kbpsの範囲で評価した。また実験環境を表1に示す。

4.2 プロセス実行時間の計測

1枚のフレームを描画する為には、デコード、テキストチャメモリへの転送、テキストチャマッピングの処理を行わなければならない。かつ複数のファイルを読み込むと

表1: 実験環境

OS	Fedora Core 4
CPU	Intel Xeon 2.40GHz
コンパイラ	gcc version 3.2.3
グラフィックボード	NVIDIA Quadro4 280NVS
キャッシュ	512KB
メモリ	1GB

きは、デコードとテキストチャメモリへの転送はファイルの数だけ行われる。ここではサイズとビットレートの観点からデコード、テキストチャメモリへの転送、テキストチャマッピングにかかる時間を計測する。計測方法は、プログラム中で各処理が行われる直前と直後で時間を計測し、その差から各処理の実行時間を求め、3回の平均を計測値とする。

実際のシステムで部分画像を生成すると元の画像に対してビットレートは減少する。例えばビットレート180kbpsで720×480の画像を4分割すると、360×240の部分画像のビットレートは45kbpsとなる。そこで720×480を単位サイズとして、これらの画像を単位サイズあたりのビットレートを180kbpsと呼ぶことにする。単位サイズあたりのビットレートを45kbpsとしたときの画像サイズと処理時間の関係を図6に、画像サイズが1倍の時のビットレートと処理時間の関係を図7に示す。

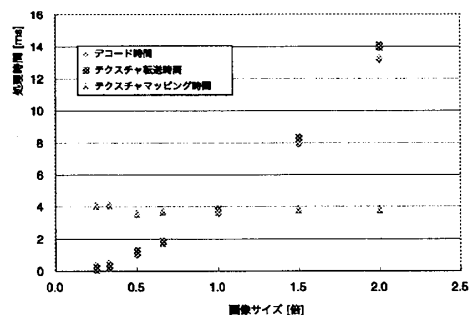


図6: 画像サイズと処理時間 (ビットレート 180kbps)

図6より、テキストチャマッピング時間はサイズに依存しないが、デコード時間とテキストチャ転送時間はサイズが大きくなると増加することが分かる。図7からは、テキストチャ転送時間とテキストチャマッピング時間はビットレートに依存しないが、デコード時間はビットレートが増加すると共に増加することが分かる。

4.3 ビットレートと分割ファイル数における考察

理想的なフレームレートは30fpsなので、各プロセスの実行時間の総和が33msを超えなければ遅延なく再生可能であり各プロセスの実行時間の総和が33msを超えてしまうと遅延が発生する。ここでデコード、テキストチャメモリへの転送、テキストチャマッピングにかかる時間をそ

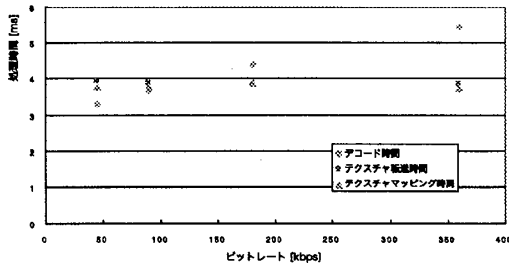


図 7: ビットレートと処理時間 (画像サイズ 1 倍)

それぞれ, T_d, T_t, T_m とし, 分割ファイル数を n とする. ストリーミング配信が遅延なく可能なら, 1 フレームあたりの時間 T が

$$T = nT_d + nT_t + T_m \leq 33$$

となれば 30fps を維持できる.

4.2 節より各プロセスの実行時間が分かったので, これより, 実際に複数の部分画像を同時再生したときの推定フレームレート f を求める. 導出式は

$$f = \frac{1}{nT_d + nT_t + T_m}$$

となる. これより, 単位サイズあたりのビットレートを 45kbps, 90kbps, 180kbps, 360kbps とし, 2 倍のサイズ (1440×960) の動画を 4, 9, 16 のファイルから生成したときの推定フレームレートを導出した結果を表 2 に示す.

表 2: 複数画像再生時の推定フレームレート (単位: fps)

	45kbps	90kbps	180kbps	360kbps
4 分割	30.6	29.5	27.2	24.5
9 分割	29.2	27.0	24.2	21.1
16 分割	25.7	24.1	21.0	17.3

4.4 実験全体に対する考察

4.2, 4.3 節で, 画像サイズ, ビットレートに対し各処理の実行時間を求め, フレームレートについて検証を行った. これらの議論から, 多チャンネルで部分画像を配信する際には, サイズとチャンネル数とビットレートの選択が重要なことが分かった.

5. おわりに

本研究で提案した全方位高精細動画配信システムを実現するためにプロトコルの改良とプレイヤーの開発を行った. またプレイヤーの再生能力の評価実験を行い, 配信可能な画像サイズ, ビットレートの関係を検証した.

今後の課題としては, ストリーミング配信が行えるようにプロトコルを MPlayer に対応させ, ストリーミング配信で用いる mp4 形式をサポートし, 異なったサイ

ズの動画画像を同時に再生できるようにプログラムを改良することである.

謝辞

本研究は独立行政法人情報通信機構「民間基盤技術研究促進制度」の援助を受けた.

参考文献

- [1] 山澤, 八木, 谷内田: “移動ロボットのナビゲーションのための全方位視覚センサ hyperomni vision の提案”, 電子情報通信学会論文誌 D-II, Vol. J79D-II, No. 5, pp. 698-707 (1996.5).
- [2] 重本, 星川, 長原, 岩井, 谷内田, 田中: “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”, CVIM-147, pp. 88-90 (2005).
- [3] 松延, 星川, 重本, 渡邊, 長原, 岩井, 谷内田, 田中: “複合センサカメラを用いた高解像度動画の撮像・提示システム”, 画像の認識・理解シンポジウム 2005, Vol. 2005, No. 7, pp. 1602-1603 (2005).
- [4] 渡邊, 岩井, 長原, 谷内田: “時空間周波数の異なる画像列からの高解像度動画の合成”, 情報科学技術レターズ (FIT2004), Vol. 3, No. L1-004, pp. 169-172 (2004).
- [5] 松延, 長原, 岩井, 谷内田, 田中: “モーフィングによる高解像度高フレームレート動画の生成”, 電子情報通信学会技術研究報告, Vol. PRMU-2004, No.178, pp. 85-90 (2005).
- [6] 山澤 一誠, 石川 智也, 中村 豊, 藤川 和利, 横矢 直和, 砂原 秀樹: “Web ブラウザと全方位動画をを用いたテレプレゼンスシステム”, 電子情報通信学会論文誌 (D-II), Vol. J88-D-II, No. 8, pp. 1750-1753 (2005).
- [7] 藤原洋, 安田浩: 標準ブロードバンド+モバイル MPEG 教科書, アスキー (2003).
- [8] F. Pereira and Touradj Ebrahimi: THE MPEG BOOK, Prentice Hall (2002).
- [9] 楊浩然: 多チャンネル配信による全方位高精細動画配信システムの構築, 修士学位論文, 大阪大学大学院基礎工学研究科 (2006).
- [10] H. Schulzrinne, A. Rao and R. Lanphier, “Real Time Streaming Protocol (RTSP)”, RFC 2326, Internet Engineering Task Force (1998).
- [11] MPlayer - The Movie Player, <<http://www.mplayerhq.hu/DOCS/HTML-single/en/MPlayer.html>>
- [12] Manson Woo, Jackie Neider, Tom Davis: OpenGL プログラミングガイド, アジソン・ウェスレイ・パブリッシャーズ・ジャパン株式会社, 1997