Vol.2009-MPS-75 No.19
2009/9/10

対話的実時間遠隔可視化システムの構築

小堀由貴[†]　来見真理[†]
石川千里[†]　高田雅美[†]　城和貴[†]

今日，医療現場や気象研究などにおいて三次元可視化手法を利用した解析が活用されている．これら三次元可視化では計算量ならびにデータ量が莫大である一方，その中から解析すべき部分を見つけ出すためには対話的操作が重要となってくる．そこで我々は，遠隔にいる複数の使用者がリアルタイムで解析が可能となる対話的実時間遠隔可視化システムの開発に着手している．本稿では，このシステムで利用されるサーバからの画像配信のための手法について提案する.この手法により，サーバから実時間での画像配信が可能となり，サーバとクライアント間のストレスのない対話的操作の実現が期待できる．

Development of an Interactive Real-time Remote Visualization System

Yuki Kohori[†]　Mari Kurumi[†]
Chisato Ishikawa[†]　Masami Takata[†]　Kazuki Joe[†]

Nowadays, 3D visualization based analysis is well-known for various fields such as medical front and weather research. Although they are computation and memory intensive, the users require interactive operations such that they detect a part of space inside of the given 3 dimensional data. To deal with the contradictive processes, we start the development of an interactive real-time remote visualization system, which allows users to perform real-time analysis from remote sites. In this paper, we present a live streaming method for real-time remote visualization. Using the method, we get interactive operations between remote users and real-time visualization servers.

1. はじめに

医療現場ではCTやMRIなどの三次元データに適用される三次元可視化手法を利用した画像診断が欠かせない[1]．三次元可視化手法は，気象研究では台風の構造[2]や気流の動き[3]の解析に利用されている．これらの三次元可視化手法を利用した解析で扱うデータ量は膨大であり，計算量も莫大となる．一方，解析すべき部分を素早く見出すためにはインタラクティブな操作が必要となり，計算量ならびにデータ量を勘案すれば一般に困難と言える．この問題に対して，高性能な計算機資源を利用することは問題の直接解決になるが，コストの問題が残ることになる．そのためユーザは可能であればそのような計算資源を遠隔利用することを考える．ここで，複数ユーザの同時利用や，通信帯域の問題があって，対話的操作の実現は極めて困難となる．

我々は，遠隔での三次元可視化計算を対話的に操作するシステムを開発している．本稿では，このシステムで利用されるサーバからの画像連続配信のための手法について提案する．

以下，2. 章において既存研究について述べる．3. 章において我々が開発しているインタラクティブな遠隔操作を可能とするシステムの概要について記述する．4. 章において開発しているシステムで利用される画像配信のための手法について提案する．この手法は，対象画像のFLV（Flash Video）形式変換，共有メモリを用いたFLV変換プログラムと配信プログラム間の通信，ストリーミング配信によって構成される．

2. 既存研究

既存の研究としては，東北大学が行っているスーパーSINETを利用した大規模遠隔可視化処理がある[4]．スーパーSINETとは，日本全国の主要な大学や研究機関の間を10Gbps の広帯域で接続している研究用広域ネットワークである．大規模遠隔可視化処理に関する研究では，東北大学情報シナジーセンターからスーパーSINET を介して北海道大学情報基盤センターの超高速可視化サーバをインタラクティブに遠隔利用し，実効帯域幅や描画性能を評価しているものである．ここでは，可視化サーバで生成した画像を利用者の端末に表示するために，SGI 社のOpenGL VizServer を用いている．また，可視化サーバ上のプログラムは演算サーバからのデータ受信に加えて可視化処理，およびインタラクティブ操作への対応の実現のために，OpenGL のユーティリティライブラリであるGLUTを用いている．

このシステムの利点は，演算サーバによるシミュレーションの途中結果として得ら

[†] 奈良女子大学大学院人間文化研究科
Graduate School of Humanities and Sciences,　Nara Women's University

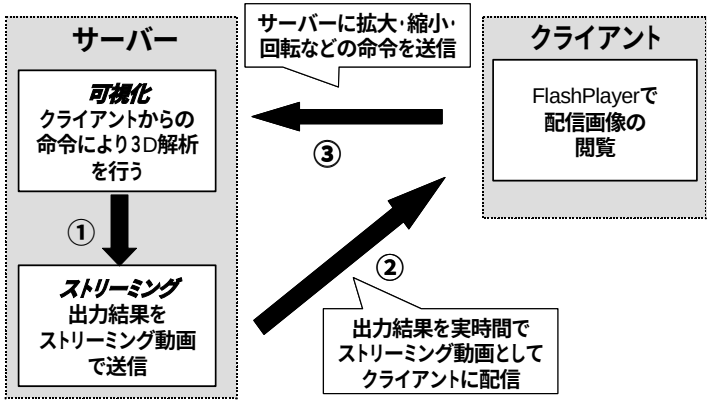

図 1　システムの概念図

れた大容量データを，遠隔地の可視化サーバに順次転送し，リアルタイムに可視化ができている点である．このシステムにより，物理的に離れた場所にいる複数の利用者が高価な可視化サーバを共有し，限られた計算機資源を有効に活用できる．しかし，欠点として，広帯域のネットワークであるスーパーSINETが必要であること，複数のデータの可視化を同時に行えないこと，インタラクティブ操作に対する結果の表示が遅いことがあげられる．そのため，一般ユーザが利用することは困難である．

本研究では，広帯域を用いず，一般ユーザでも利用可能な500kbpsから1Mbps前後という比較的少ない帯域でのストレスのない配信を行うシステムを開発する．また，遠隔での効果的な解析作業実現のため，データ通信がリアルタイムに行えるようにすることを目指す．

3. インタラクティブな操作性を実現するためのシステム

本章では，我々が開発中のシステムの概要について説明する．図 1 は，システムの概念図である．

本システムにおけるサーバとは，気象観測などの膨大なデータに対して可視化を行う可視化サーバと，その可視化結果をクライアントにリアルタイムで配信するストリーミングサーバで構成される．これにより，可視化結果をリアルタイムでストリーミング動画として，ネットワークを通じクライアントに送信する．

クライアントは Flash Player によって，ストリーミング動画を閲覧できる．ユーザは図 1 の②によって配信された動画に対し，表示部分の拡大，縮小，回転などのインタラクティブな操作命令を，マウス操作，またはキーボード入力でサーバ側に送信する．図 1 の③の通信による命令を受け取ったサーバは命令に従い可視化サーバでデータを計算し再描画を行う．そして図 1 の①で，ストリーミングサーバに可視化結果を送り，ストリーミング配信することにより，ユーザの命令どおりの動画が閲覧できる．これらの繰り返しにより，ストレスのない遠隔でのインタラクティブな操作が実現する．

本システムにより，遠隔にいる複数の使用者がリアルタイムでの閲覧ができるようになる．また，サーバで生成された画像を拡大縮小，回転など，使用者が遠隔でインタラクティブに，場所・時間を問わず操作できるようになるため，解析作業の効率向上に繋がる．

4. クライアントとサーバのリアルタイム画像通信方法の提案

本章では，図 1の②を実現するために，クライアントとサーバ間でリアルタイムに画像を通信するための方法を提案する．

以下 4.1 節において目的と概要を述べる.4.2 節で，可視化サーバより生成された BMP 形式の画像をリアルタイムで FLV 形式に変換するプログラムについて説明する．4.3 節でストリーミングサーバに動画を認識させるために使用する共有メモリを用いた外部アプリケーションとの通信方法について述べる．4.4 節でストリーミングサーバに認識させた動画を配信するストリーミング配信について記述する．

4.1 目的と概要

本システムは，遠隔でインタラクティブに操作できるリアルタイム可視化を目的としている.そのためには可視化結果をいかに効率良く,かつ高速にユーザに配信できるかが重要である．

従って，可視化サーバからアップロードされた画像を，ストリーミングサーバは遅延なくリアルタイムにクライアントへストリーミング配信し閲覧できるようにする．そのためには,アップロードされた画像をリアルタイムでストリーミングサーバに取り込むことが必要となる．そこで，メモリ上での FLV 変換を行う処理，変換されたデータを共有メモリによって実時間でストリーミングサーバに転送する処理，さらに転送したデータをリアルタイムでストリーミング配信しクライアントに表示させる,データの送信処理と受信処理を開発する．

4.2 FLV 変換

FLV とはアドビシステムズが開発している動画ファイルフォーマットである[5]．FLV は，ユーザがコーデックなどを気にせず簡単に利用でき，You Tube などの動画投稿サイトなどにも使用されていることから，インターネット上での動画再生ファイルの 1 つとして大変注目を浴びている．

図 1の①において，可視化サーバからストリーミングサーバに対して，毎秒30枚の静止画が送信される．この静止画は，BMP形式で保存されているため，FLV形式に変換する必要がある．

既存のFLVファイル変換方法としては，変換ソフト ffmpeg などを用いる方法がある[6]．この変換はまずconvert コマンドを使用して，BMPをMPGに変換し，ffmpeg でMPGをFLVに変換する．Pentium Dual-Core 1.6GHz のCPU,1GBのメモリを搭載した計算機で, この変換ソフトの処理時間は，1枚約351KBである400 x 300のBMP画像10枚を変換するために，約3．043秒である．リアルタイムでのインタラクティブ操作を実現させるためには，この方法よりも高速な変換方法が必要となる．

そこで，リアルタイムでFLV動画を作成するために，RGB情報を取得し，FLVファイルへ変換する全ての流れをメモリ上で行うことにより，処理時間の短縮を目指す．画像の圧縮には，zlibライブラリを用いる[7]．zlibとは，zipやgzip， PNG画像に使用されている可逆圧縮アルゴリズムをライブラリ化したものであり，GNUライセンスで配布されており自由に使用可能なアルゴリズムである．また，国際的な圧縮方法でもあり，高い信頼性がある．

図 2は，提案するFLV変換の流れを表す．BMPファイルヘッダは14byte固定となっており，最初の2byteがファイルタイプ‘BM’かどうかでBitmapファイルを識別する．まず, BMPファイルのヘッダ情報とメタデータを取得し，メモリへの書き込みを行う．そして，BMPファイルのRGB情報が存在する間，データの読み込み，zlib圧縮，データとタグのメモリサイズであるPrevious Tag Sizeの値を計算し，メモリに書き込む動作を繰り返す．変換には, 圧縮後のデータバッファサイズなどが必要なため, 圧縮部分には基本関数は使用せずに，それらの値が取得できる高次関数compress2()を使用する．compress2 ()は9段階の圧縮率を指定できる[8]．

図 3 に画像データの圧縮における概要を示す．色ビット数，画像の幅など, BMP の画像データは，**図** 3 のように画像の左下から右上に向かって#1 から#16 の順に上下が反転した状態で HDD に格納されている．このデータをブロック幅 block_w，高さ block_h として，いくつかのブロックに分けて圧縮を行う．BMP ファイルには画像 1 行ごとの情報量が 4 の倍数でなければならないという決まりがある．よってブロックの縦横幅はそれぞれ 4 の倍数値にし，#13 のような画像の最後の余り部分は任意の値でよい．1 ブロックにおける圧縮も画像データと同様に左下から右上に行う．

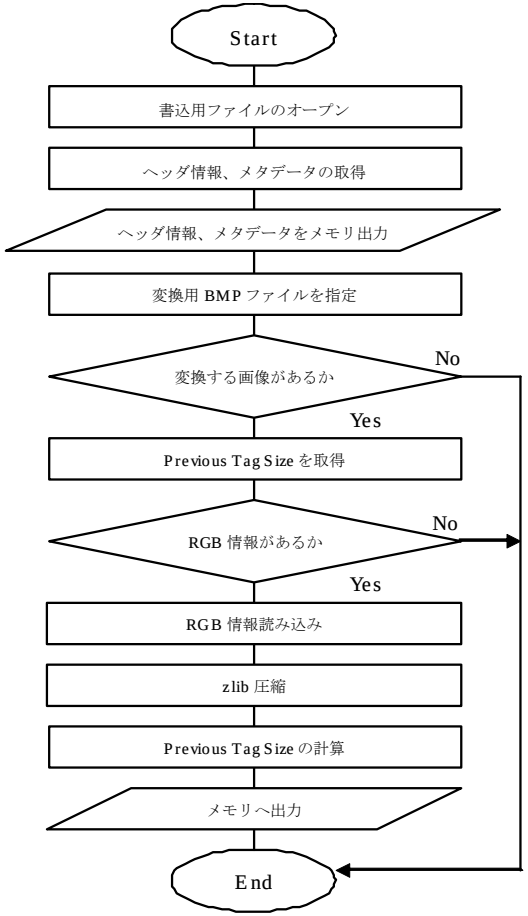

図 2　FLV 変換の流れ

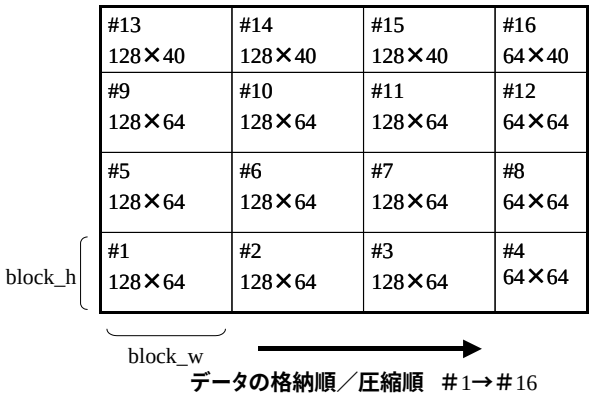

図 3 画像データの圧縮

4.3 FLV 変換プログラムと配信プログラム間の通信

メモリ上で FLV 形式に変換した動画をリアルタイムで配信するための方法について説明する．

一般的には，ストリーミングサーバが動画を読み込む際，HDD に毎回アクセスすることになるが，この方法では，リアルタイムな動画配信は困難である．そこで，FLV 変換プログラムと Flash 上のストリーミング配信プログラムとの間でメモリを共有し，ファイル内容をメモリ上に反映させることで，仮想アドレスを与え，データ受け渡しを高速化することを提案する．

図 4 に提案する通信方法の概念図を示す．Flash は，Web 上のアプリケーションということもあり，メモリを操作できない仕様となっている．このため，Flash だけでは共有メモリの技術を使用できない．そこで，External API の使用により，外部アプリケーションとの通信を確立し，データ受け渡しを高速化する．External API とは ActionScript の一部で，Flash Player のコンテナとして機能する Web ブラウザなどの外部アプリケーションで実行されるコードと ActionScript 間の通信用メカニズムを提供する[9]．External API を用いてクライアントに FLV データを通信するために，ストリーミングサーバとして red5[10]を採用し，Java を用いてプログラム開発を行う．これにより，FLV 変換プログラムと配信プログラム間の通信を共有メモリで実現することが可能となり，データ送受信の高速化ができる．

提案する通信方法では，FLV 変換プログラムを C 言語で開発し，ストリーミングサーバは Java で開発される．そのため，C 言語と Java 間での共有メモリを用いた通信

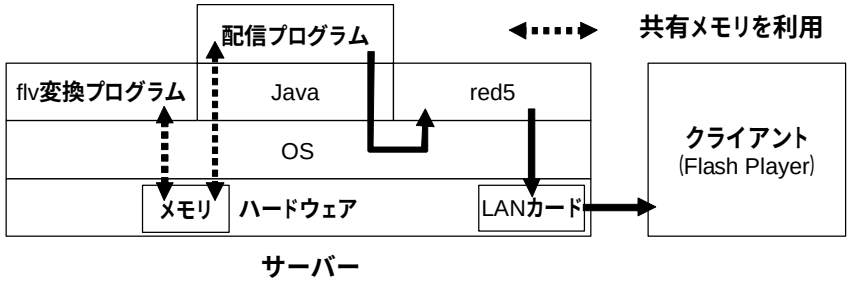

図 4 提案する通信方法の概念図

が必要となる．図 5にC言語のプログラムとJavaのプログラム間通信の概念図を示す．まず，C側のプログラムで，shmget()で共有メモリを作成する[11]．ここで共有メモリの領域確保に成功するとユニークなIDを返される．Java側はそのIDに従って共有メモリにアクセスする．その後，C側はshmat()で共有メモリにデータをマップする．CからJavaの機能を呼び出すためにはJNI（Java Native Interface）を用いる．JNIはJavaのプログラムから他の言語で開発されたネイティブコードのプログラムを利用するためのAPIである[12]．JNIを用いるために，まず共有メモリにアクセスするためのJavaプログラムのコンパイルにより作成されたクラスファイルから，C言語で使用するヘッダーファイルを作成する．そのヘッダーファイルからVC++によってC言語のdllファイルを作成する．最後にJavaファイルを実行することでデータが共有できる．C側はshmctl()で共有メモリを削除して終了する[13]．

4.4 ストリーミング配信

本節では，C 言語プログラムと Java プログラム間で共有しているデータをストリーミング配信する方法について述べる．

ストリーミングとは，動画データをパケット単位で分解してユーザ側へダウンロードさせる手法である．ユーザは動画ファイル全体をダウンロードし終わるのを待つことなく受信したパケット単位で即再生できるため，要求と再生が同時に実現できる．その際，ユーザが受け取ったパケットは一旦メモリに記憶されるが，パケット単位で再生が終了するごとにメモリが解放される．よって，ユーザのハードディスクやキャッシュ等にも，特別な方法を使わない限り動画データが残ることはないため，クライアント側の計算機が高性能である必要はない．

本ストリーミング配信実現のために以下のものを使用する．

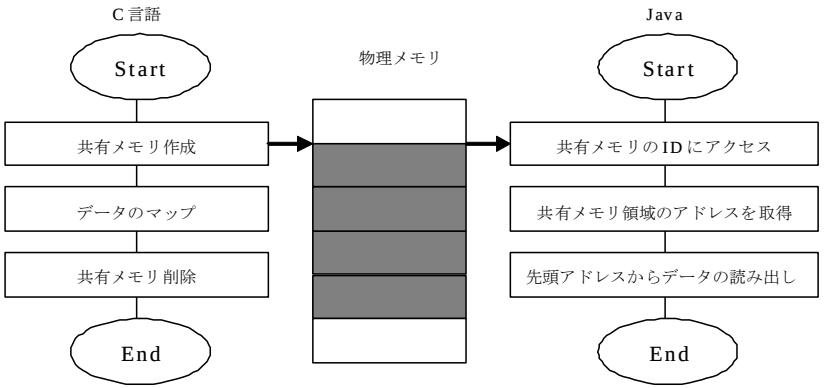

図 5　C 言語のプログラム，Java のプログラム間通信の概念図

- ストリーミング配信サーバ
  - red5
- クライアント
  - Macromedia Flash Player
- Java コンテンツ制作ツール
  - Eclipse
- Flash コンテンツ制作ツール
  - Adobe Flash CS3 Professional

ストリーミング配信サーバは red5 を使用する．red5 とは，オープンソースで提供されているストリーミングサーバである．また Java で実装されているため，メモリ操作が可能である．

Flash Player は全世界 90%以上のパソコンにインストールされ幅広く使われており，特殊なプラグインを必要せず，すぐに導入することが可能となるため採用する．配信用および受信用のアプリケーションは SWF 形式のため，Windows でも Macintosh でも，Flash Player さえあれば利用可能である．

Eclipse は，サーバ側で使用する Java を製作するために用いる[14]．

Flash CS3 は，クライアント側で使用する flash を製作するために用いる[15]．

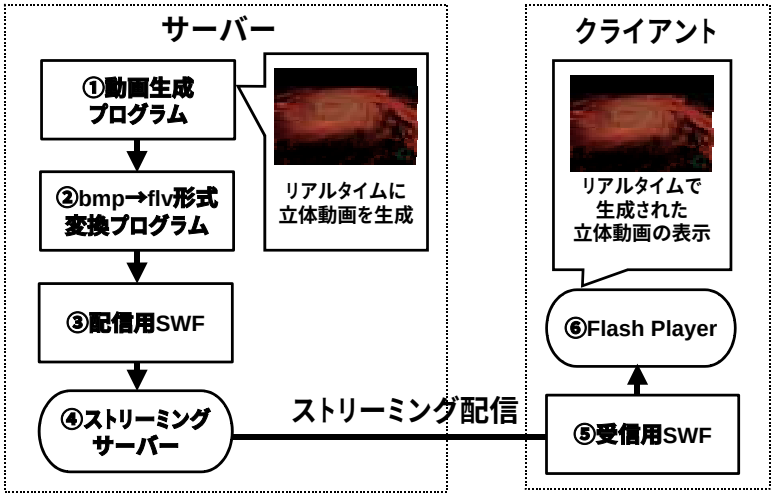

図 6　ストリーミングサーバとクライアントの関係図

図 6は，本提案方法によるストリーミングサーバとクライアントの関係図である[16]．

サーバでは，はじめに図 6 の③にあたる C と Java 間でのデータ通信プログラムより受け取ったデータを，図 6 の④にあたる配信プログラムであるサーバ側で読み込み，red5 に認識させる．red5 への接続が成功したら，認識させたデータを読み出し配信する．次に web.xml，red5-web.xml，red5-web.properties にそれぞれ作成したアプリケーションを指定し，パスを通す．プログラムは Eclipse のワークスペースでコーディングした後，red5 の webapp フォルダにコピーし，red5 を再起動することによって有効になる．

クライアント側では，まず図 6 の⑥にあたる受信表示用 SWF を作成する．受信表示用 SWF に，図 6 の⑤の red5 があるパソコンの IP アドレスを指定し，接続する．次に，動画を扱うオブジェクトを作成し，red5 から送信された映像をクライアント側に表示させる．red5 は図 6 の④の配信プログラムから受け取った映像を，クライアント側からのリクエストに応じて配信する．受信表示用 SWF を実行すると RTMP で配信されている可視化サーバで作成されたストリーミング動画を図 6 の⑦の Flash Player で閲覧することができる．red5 は SWF との間のデータやりとりを RTMP（Real－Time Messaging Protocol）という専用プロトコルを使用して行う．クライアントの Web ブラ

ウザは HTTP を通して Web サーバから HTML や SWF をダウンロードし，次に Flash Player がその SWF に含まれる動作に従って RTMP プロトコルを使用して red5 に接続する．red5 上の可視化サーバが作成した動画は red5 からストリーミングされ，クライアントの SWF で再生される[17]．

5. まとめ

気象研究や医療現場において，膨大なデータの効率的な解析を実現するために，遠隔にある計算機を用いて，リアルタイムにインタラクティブな操作が可能な可視化システムが求められている．本研究では，このシステムで利用可能な可視化データの通信方法について提案した．

その提案手法では，可視化サーバから送られてくる BMP 画像をリアルタイムで配信するために，メモリ上で FLV 形式に変換する．また，FLV データをリアルタイムでストリーミングサーバ red5 に認識させるために, C 言語のプログラムと Java のプログラム間で共有メモリを使用した．

今後の課題として，我々が開発しているシステムに適用し，性能評価をすべきである．

参考文献

1) A Pocket Atlas Michael L. Grey , Jagan , Mohan Ailinani:CT and MRI Pathology:,McGraw-Hill Medical (2003/04/01)
2) 村山貢司：台風学入門，山と渓谷社, (2006/07)
3) 社河内敏彦, 辻本公一, 前田太佳夫：流体力学 ― 基礎と応用 ―，養賢堂(2008/08)
4) 滝沢寛之，小林広明：スーパー SINET を利用した大規模遠隔可視化処理の評価，東北大学情報シナジーセンタースーパーコンピューティング研究, 東北大学情報シナジーセンター年報，No.3,pp.106-114(2004/06)
5) 永井勝則：初めての Flash Video，オライリー・ジャパン／オーム社(2007/12)
6) 原一浩：FFmpeg で作る動画共有サイト，毎日コミュニケーションズ(2008/02)
7) Zlib
http://zlib.net/
8) Zlib 入門
http://oku.edu.mie-u.ac.jp/~okumura/compression/zlib.html
9) Flash CS3 ドキュメンテーション:External API の使用
http://livedocs.adobe.com/flash/9.0_jp/main/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00000337.html
10) Open Source Flash
http://osflash.org/flv
11) TERAMOTO Takayuki, TAKAHSHI Genya , Consideration of cooperation of a JAVA/C language in network distributed environment, Research reports, Tsuyama Technical College 46 pp.73-78 (2005/02/28)
12) Tim Lindholm , Frank Yellin：The Java Virtual Machine Specification ,SunTM Microsystems, Inc.(1996)
13) 早田恭彦:Java 向けソフトウェア分散共有メモリの実現，社団法人情報処理学会, 42(SIG_3(PRO_10)) pp. 14-26 (2001/03/15)
14) eclipse
http://www.eclipse.org/downloads/index.php
15) ハヤシカオル：ActionScript クリエイティブテクニック Flash CS3／8 対応版,エムディエヌコーポレーション／インプレスコミュニケ(2008/10)
16) 上野亨：FLASH ActionScript バイブル MX のツボ with Flash Communication Server MX，オーム社(2002/09)
17) Yuki Kohori, Yuri Shirakawa, Chisato Ishikawa, Masami Takata, and Kazuki Joe：Real-time 3D Movie Generation by Anaglyph for Live Streaming，International Conference on Parallel and Distributed Processing Technologies and Applications 2008 (PDPTA'08), Vol.II, pp.758-763(2008/7/14)