

対話型 JPEG 2000 画像伝送方式 JPIP の実装

Implementation of JPIP, interactive JPEG 2000 image transmission, system

原 潤一

株式会社リコー リコー未来技術研究所

wave@nts.rioh.co.jp

あらまし

ここで紹介する対話型 JPEG 2000 画像通信システムの実装は、JPEG 2000 標準の画像通信規格である JPIP 画像通信プロトコルを用いて実装されています。JPIP では、画像伝送効率を高めるために、JPEG 2000 画像が持つ解像度・品質・位置のスケーラビリティを利用した画像通信方式が定義されています。JPIP では 2 種類のメディアタイプが定義されており、ウェーブレット処理単位であるタイル毎に送る JPT-stream と、符号化処理単位であるプリシнкт毎に送る JPP-stream があります。この 2 種のメディアタイプによる転送モードを本 JPIP 実装で用いており、特に JPT-stream の実装では、単純にタイルデータ転送する代わりに、サーバー側で画像表示に必要なパケットを使用してタイルデータを再構築して転送する方式の実装を行っています。更に、本実装ではレガシークライアントでの表示を考慮から、転送画像データとして JPEG を用いるモードも実装しています。これら送信方式の比較とそのときのシステム実装について説明します。

Abstract

An implementation of a JPEG 2000 interactive communication system is described, which is based on JPIP image communication protocol. JPIP specifies an image communication scheme, which utilizes JPEG 2000 resolution/quality/position scalability so as to enhance transmission efficiency. Two transmission modes are defined in JPIP, tile-based JPT-stream mode and precinct-based JPP-stream mode. In this paper, we implement four transmission modes; tile-based, precinct-based, precinct-tile-based, and jpeg-based modes. Comparison between these transmission modes and details of system implementation are also explained.

1. はじめに

JPEG 2000^[1]は 2000 年に規格化された画像復号化の国際標準であり、デジタルシネマ、医療分野や、アーカイブ分野で広く利用されている画像フォーマットです。この規格は、任意の圧縮率で優れた画像品質のコードストリーム生成を可能にし、JPEG 等の旧来の画像符号化規格には無い解像度・品質・位置のスケーラビリティ特性を有します^[2]。この特性により、画像全体のコードストリームを復号することなく画像の一部のみを復号することができ、画質劣化が少ない編集が可能です。更に、JPEG 2000 コードストリームの状態での画質調整とデータ量削減が可能になります。

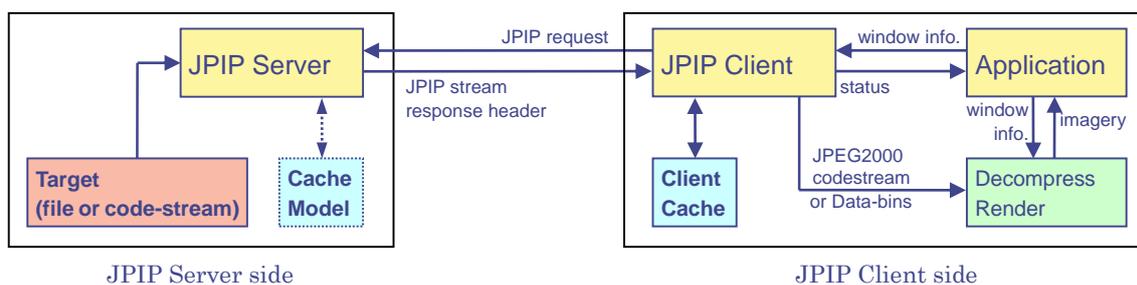


図 1. JPIP interactive image communication system

JPIP と呼ばれる JPEG 2000 Part 9^[3]は、クライアントサーバーモデルの通信規格であり、JPEG 2000 画像を効率良く閲覧可能にするための対話型のプロトコル規格です。JPIP では、効率の良い画像伝送を可能にするために、JPEG 2000 コードストリームが持つ上記スケーラブルな機能を利用しています。画像を取得するクライアントは、現在関心があるフォーカスウィンドウを指定する簡単な記述文を用いてデータ要求を定式化し、クライアント自身が直接圧縮されたターゲットファイルの特定部分にアクセスすることなく、目的の画像領域のデータを取得することができます。換言すれば、クライアントはビューウィンドウ領域、画像成分と目的の空間分解能を識別し、サーバーにその情報を用いて画像を要求します。要求を受けたサーバーは、JPEG 2000 コードストリームをデータビンに分割し、クライアント側に送信済みのデータビンを再送するという無駄が生じないように、JPEG 2000 コードストリームの必要部分のデータを最適に返信します。

JPIP での JPEG 2000 データの分割送信方法には 2 種類あり、ひとつはウェブレット処理単位であるタイル単位毎に、もうひとつは符号化処理単位であるプリシント単位毎に画像データを送信します。送信される画像データの単位は、その何れか方式に基づいたデータ形式となっています。両方の伝送方法において、コードストリームのメインヘッダは、それ自身のデータビンが割り当てられています。タイルベース伝送では、例えば、送信対象のタイルに属する全タイルパート符号を連結してタイルヘッダを付加することによって形成されたデータストリームからなるデータビンを送信します^[4]。プリシントベース伝送では、データビンとして、各タイルヘッダとプリシントを割り当てます。送信方式にタイルベース伝送を用いると、JPIP システムの構築が容易ですが、伝送がタイルパート単位というデータ自体の柔軟性の低さが、空間的アクセス性能の低下を招いてしまいます。プリシントベース伝送方式では、例え画像全体が単一タイルとして圧縮された場合であっても、細かい単位での効率的な空間アクセス性能によるデータ取得が可能ですが、クライアントとサーバーの両方の実装に余分なコストを必要とします^[5]。

この問題に対応するために、筆者はタイルベース伝送を使用してプリシントベース伝送のデータ転送と同等の柔軟性を実現する方式を提案しました^[6]。この方式では、多少の実装複雑性をサーバー側に要求しますが、クライアント側の実装複雑度はタイルベースと同じになります。この機能実現のために、要求されたビューウィンドウ領域に関連する JPEG 2000 コードストリームのパケットデータを選択し、その選択したパケットデータとそれに関連するヘッダー情報とを結合して新た

なタイルパートを生成し、その再構築データをタイルベース伝送することにより、プリシントベース伝送の柔軟性を持つタイルパートデータ伝送の実現が可能となります。

本稿は次のように構成されています。最初に、2 節で 2 つの JPIP 送信モードであるタイルとプリシントの送信方式の概説を行い、それぞれの送信方式の違いを述べます。次に、3 節で以前提案した文献[6]のタイルベース伝送方法の概説を行った後、4 節でこれらのシステム実装の概要、5 節で各方法の比較の概説を行い、文献[6]の方法の有効性を示します。また、最初に標準化された JPIP 規格では、運用上のコスト発生の可能性があるので、技術追加された内容についても簡単に言及します。

2. JPIP の 2 つの返信モード

JPIP には、2 種類の返信方法、タイルベース伝送とプリシントベース伝送があります。タイルベース伝送は、ウェーブレット処理単位であるタイル毎に返信するモードであり、プリシントベース伝送は、符号化処理単位であるプリシント毎に返信するモードです。

JPIP は通信路として HTTP や UDP を用いることを想定したプロトコルであり、JPEG 2000 データ交換のために通信路に流れる返信データのメディアタイプには、「JPT-stream」と「JPP-stream」という 2 種類のタイプがあり、それぞれ、タイルベースとプリシントベースの伝送に対応します。これらメディアタイプの各メッセージは、メッセージヘッダーが先行して送信され、その後、単一のデータビンから成るメッセージを連結したシーケンスとして送信されています。このデータビンは、JPEG 2000 コードストリーム内の情報から構成されたストリームの構築が可能になるような方法で JPEG 2000 の部分データを含みます。メッセージが任意時点で終了しても、最小限の制約でメッセージ自身の意味を失うことがなく、受け取り側で再順序付け可能になるように、各メッセージは完全に自己完結可能な記述となっています。

図 2 に JPIP システムのタイルベースとプリシントベースのアーキテクチャ概要図を示します。プリシントベースシステムには、タイルベースシステムと比べて、「Precinct Identifier」と「Tile-part Builder」の 2 モジュール追加と、「Tile-part Fetcher」の代わりに「Packet Fetcher」が必要です。この 3 モジュール実装から、プリシントベースのシステム実装には、タイルベースと比べて追加の実装コストが必要となります。

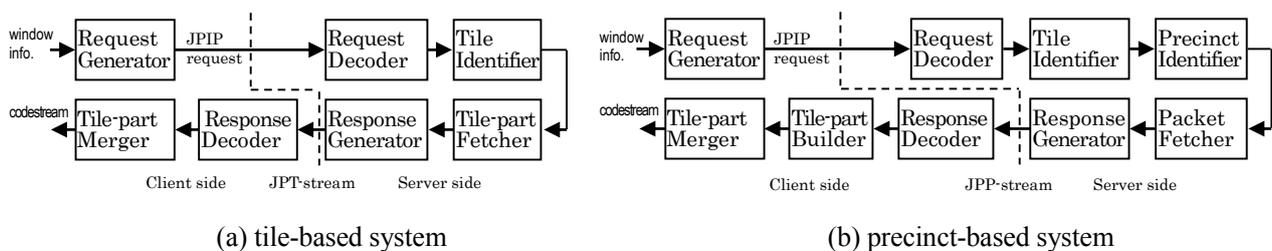


図 2. JPIP client and server architecture

A. JPT-stream - タイルベース伝送

タイルベース伝送では、最初に、要求されたターゲット画像名、その要求範囲を示すビューウインドウ領域、縮小レベルを指定するスケールなどの情報から「Request Generator」にて JPIP メッセージを生成してサーバーに送ります。サーバーの受信メッセージは、「Request Decoder」にてそれら情報が復号され、復号されたビューウインドウ領域から「Tile Identifier」にて要求領域内に属するタイルが選択されます。選択されたタイルの左上と右下のコーナー (tx_0, ty_0) 及び (tx_1, ty_1) の参照グリッド上の位置は次式より求められます。

$$tx_0 = \lfloor (wx_0 - T_{ox}) / T_{sx} \rfloor \quad ty_0 = \lfloor (wy_0 - T_{oy}) / T_{sy} \rfloor \quad (1)$$

$$tx_1 = \lfloor (wx_1 - T_{ox}) / T_{sx} \rfloor \quad ty_1 = \lfloor (wy_1 - T_{oy}) / T_{sy} \rfloor \quad (2)$$

ここで (wx_0, wy_0) 及び (wx_1, wy_1) は、それぞれ正規化されたビューウインドウの左上と右下のコーナーであり、要求された縮小レベルに対応するサブバンドドメイン中の参照グリッド上の位置になります。 (T_{ox}, T_{oy}) 及び (T_{sx}, T_{sy}) は、要求された縮小レベルに対応するサブバンドドメイン中の参照グリッド上でのタイルのオフセットとサイズを示します。「Tile-part Fetcher」にてタイルパートを取得し、応答メッセージを「Response Generator」にて生成します。クライアントで受信された応答メッセージは「Response Decoder」にて復号され、各タイルパートデータはキャッシュに蓄積され、キャッシュ中のタイルパートは「Tile-part Merger」にて復号可能なコードストリームに変換され出力されます。アプリケーション側の JPEG 2000 復号器により、このコードストリームを復号することで、アプリケーション上で画像を表示することが可能になります。

前述したように、タイルはウェーブレット変換の処理単位であり、タイル分割は JPEG 2000 画像の空間単位の分割となるため、この伝送方式のアクセス性能はタイルサイズに依存します。JPEG 2000 規格では、Start-Of-Tile-part (SOT) マーカーがタイルパートの開始位置と長さを保持するため、容易にタイルパートを取得して連結することが可能です。したがって、タイルベース伝送を行う JPIP システム実現は比較的に容易に行えますが、このアプローチによる方法では、送信単位がタイルという比較的に大きな領域単位毎になるため、空間的な柔軟特性が低下します。

B. JPP-stream - プリシнктベース伝送

プリシнктベース伝送では、タイルベース伝送と同様に、最初に、ターゲット画像名とビューウインドウ情報などがサーバーに送信され、サーバー側でそれらの情報が復号され、ビューウインドウに属するタイルが選択されます。プリシнктベース伝送では、更に、ビューウインドウ情報と選択されたタイル情報を用いて、「Precinct Identifier」にて要求領域に属するプリシнктが選択されます。選択されたプリシнкт領域の左上と右下のコーナー (px_0, py_0) 及び (px_1, py_1) は次式より求められます。

$$px_0 = \lfloor \max(tx(n)_0, wx'_0) / 2^{ppx} \rfloor$$
$$py_0 = \lfloor \max(ty(n)_0, wy'_0) / 2^{ppy} \rfloor \quad (3)$$

$$\begin{aligned}
 px_1 &= \lfloor \min(tx(n)_1, wx'_1) / 2^{PPx} \rfloor \\
 py_1 &= \lfloor \min(ty(n)_1, wy'_1) / 2^{PPy} \rfloor
 \end{aligned}
 \tag{4}$$

ここで、 $(tx(n)_0, ty(n)_0)$ 及び $(tx(n)_1, ty(n)_1)$ は要求されたサブバンドドメイン中の参照グリッド上の n 番目のタイルの左上と右下のコーナーを、 PPx 及び PPy はそれぞれプリシントの幅と高を、 (wx'_0, wy'_0) 及び (wx'_1, wy'_1) は各サブバンド中でのウェーブレットフィルタの係数参照範囲に応じて拡大されたビューウインドウ領域の左上と右下のコーナーを示します。応答メッセージ送信のために、「Packet Fetcher」にて選択されたプリシントに関連するパケットを選択し、そのパケットから応答メッセージを作成します。クライアントは、応答メッセージを取得し、そのメッセージを復号して各パケットに分解してキャッシュに蓄積し、そのキャッシュ中のパケットを用いて復号可能なコードストリームを生成して出力します。出力されたコードストリームを復号することで、アプリケーションが要求した領域の画像を表示することが出来ます。

JPEG 2000 のパケットヘッダ生成は、コードブロックのインクルージョンとコーディングパス情報と、同じプリシント内のコードブロックの符号化状態を使用し、タグツリー符号化するという複雑な処理で実現されています。したがって、パケット符号を識別し取り出すことはタイル符号を特定して取り出すことに比べて大変難しくなります。JPEG 2000 標準ではパケット符号の位置特定や符号長を保存するという便利なマーカーツールとして提供します。しかし、これらのマーカーツールがコードストリーム中に存在することは希であり、更に、システムは2つの追加モジュールを含んでいるため、プリシントベースのシステムはタイルベースのシステムよりも複雑性が増します。このように、プリシントベース伝送では複雑性が増しますが、プリシントのサイズはタイルよりも小さいため、タイルベース伝送よりもより効果的な空間アクセス性能を有します。

3. 再構築タイルを用いた送信方式

前節で示したように、タイルベース伝送方式は実装複雑度が低いですが伝送性能が悪くなり、プリシントベース伝送方式は優れた伝送性能を有するが実装複雑度が高くなり、タイルベース伝送方式の実装コストに比べて追加の実装コストを必要とします。ここでは、システムの伝送方式がタイルパート伝送に基づいている場合であっても、優れた空間のアクセス性能を提供する方法について概説します。文献[6]で提案されたこの伝送方法の基本的な考え方は、アーキテクチャ概要図として図3に示すように、サーバー側で伝送効率の良いタイルパートデータを再構築することです。

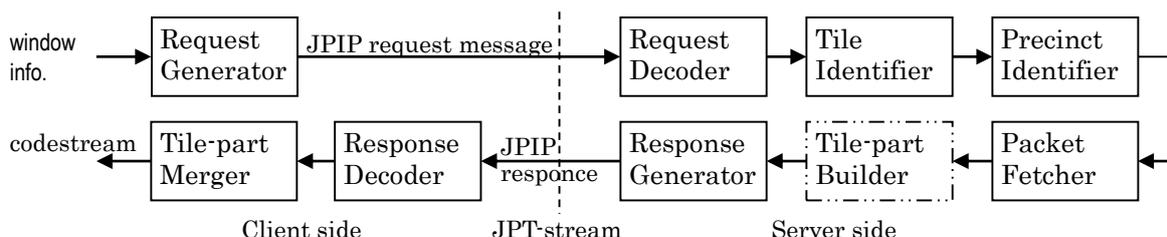


図 3. Reference[6]'s method tile-based system architecture

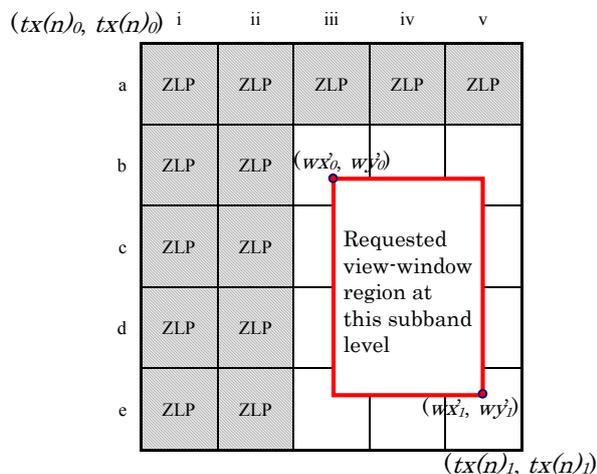


図 4. Example of filling ZLP into a tile at certain subband level.

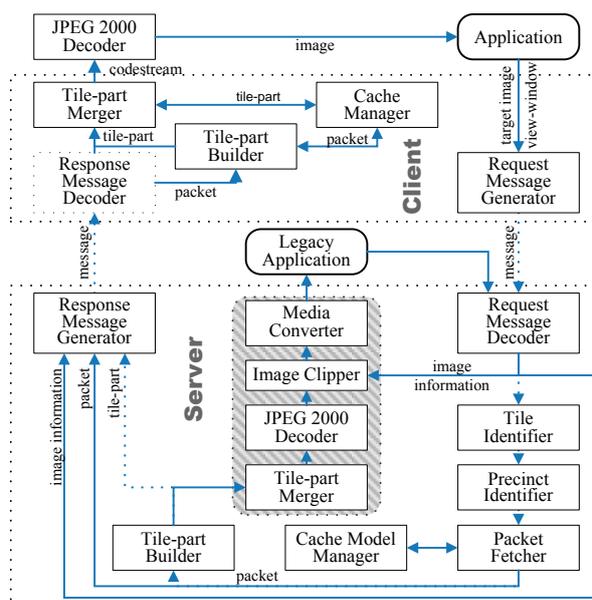
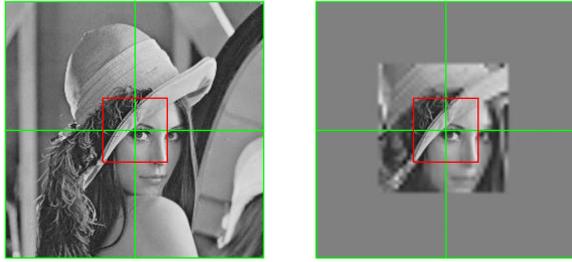


図 5. Implemented JPIP client-server system architecture

このアーキテクチャでは、プリシントベース伝送方法と同一のモジュールを用いますが、「Tile-part Builder」はクライアント側からサーバー側に移動します。したがって、この伝送システムの複雑度は、プリシントベース伝送システムと同じですが、複雑度が増したのはサーバー側のみとなり、クライアント側の複雑度はタイルベース伝送と同一になります。

選択されたパケットデータとその位置を用いて、「Tile-part Builder」にてビューウィンドウ領域に属するプリシントのパケットデータと関連情報のみから新しいタイル部分を生成します。すなわち、プログレッシブもしくは解像度レベルでの画像アクセスのために、選択されたパケットのみから完全に復号可能な JPEG 2000 タイル部分データを再構築します。これは、プリシントベース伝送方式で伝送されるのと同じパケットのみでのタイル部分生成を意味しますが、再構成符号が JPEG 2000 準拠するためには、例えタイル中の一部分のみの画像を必要とする場合であっても、再構成されたタイル部分中には全てのプリシント情報が必要となります。

この JPEG 2000 準拠タイル部分の生成問題に対応するために、パケット無しの部分に、1 バイトサイズの「Zero-Length-Packet」(ZLP) を用います。「Tile-part Builder」では「Packet Fetcher」で選ばれなかったパケットエリアに ZLP を埋めます。図 4 に ZLP 埋め込み例を示します。この例では、サブバンドレベル中に 25 個のパケットがあり、要求されたビューウィンドウ領域に属するパケットは (iii, b) から (v, e) の矩形領域のみです。「Packet Fetcher」は、この選択領域に属したプリシント領域のパケットデータを取り出します。「Tile-part Builder」は、この取り出されたパケットを埋め込み、取り出されなかった非選択パケット領域に ZLP を代用することで、新たなタイル部分を生成します。つまり、プリシントベース伝送方式では選択されたパケットデータのみを送信しますが、この伝送方式は、選択されたパケットデータ以外に複数の ZLP も送信します。



(a) normal tile-based system (b) reconstructed tile-based system

図 6. Spatial accessibility performance of method [6].

The center rectangle area and the crossed lines are requested view-window area and tile borders respectively.



(a) first resolution (b) second resolution (c) third resolution

図 7. Progressive display avirity of the method [6].

4. システム実装

ここでは、Java 言語で作成した JPIP クライアントサーバーシステムの実装について概説します。このシステムでは、文献[6]で提案されたタイルベース伝送方式と JPIP のプリシントベース伝送方式に基づく送信方法を実現しています。図 5 は本システム実装のアーキテクチャ概要図です。通常の JPIP タイルベース伝送方式は、ここでは紹介しない他のシステムで実現しており、そのシステムは、図 5 の「Precinct Identifier」と「Packet Fetcher」を省くことで実現できます。

本システムでは、JPIP で規定された HTTP と HTTP-TCP の両方の通信路を介して画像を送信し、画像タイプとして、jpeg、JPP-stream 及び JPT-stream の 3 つのメディアタイプを送信可能です。jpeg メディアタイプは Web ブラウザなどのレガシークライアントとの互換性を持たせるためにあり、この互換性より、ブラウザは静的ターゲットとして埋め込まれた JPIP リクエストを含む HTML ページで対象画像の任意領域を取得することが可能です。図中の灰色の部分、この機能の実現のためのモジュールであり、再構成コードストリームを復号した後、ビューウィンドウ領域に応じて画像を切り取り、JPEG 符号化して image/jpeg メディアとして返信します。

本システムのサーバーは、クライアントにパケットが転送済みかどうかを示すパケット送信状態を管理しています。クライアントは、各メディアタイプに対応して JPIP データピン及びデータを格納するキャッシュを有します。このシステムでは、未分化の JPEG 2000 コードストリームを復号するために完全なコードストリームを構築します。次節でこれらの伝送方式の性能を評価します。

表 1. PERFORMANCE OF SPATIAL ACCESSIVITY

Requested View-window Area	Transmitted Bytes		Advantage of proposed JPT-stream
	Reconstructed JPT-stream	JPP-stream	
R1: (192, 192) - (319, 319)	14,206	13,27	-931
R2: (152, 152) - (279, 279)	19,623	18,89	-731
R3: (0, 0) - (255, 255)	42,696	44,21	1,515
R4: (0, 0) - (127,127)	10,394	10,48	87
R5: (0, 0) - (29, 29)	1,851	1,513	-338

Note1: the codestream size is 166,582 bytes.

Note2: the frame size, fsiz, is (512, 512).

表 2. PERFORMANCE OF RESOLUTION ACCESSIVITY

Requested Frame Size	Transmitted Bytes		Advantage of proposed JPT-stream
	Reconstructed JPT-stream	JPP-stream	
16 × 16	1,049	994	-55
32 × 32	1,778	1,750	-28
64 × 64	4,390	4,424	34
128 × 128	13,629	13,974	345
256 × 256	47,187	48,741	1,554
512 × 512	168,626	174,794	6,168

Note1: the codestream size is 164,962 bytes.

Note2: the transmitted byte of normal JPT-stream is 168,626 bytes.

5. 各送信方式の性能

ここでは、文献[6]で提案された伝送方法とプリシントベース伝送方法を用いたときの空間的アクセス性能及びプログレッシブアクセス性能を評価します。入力データとして JPEG 2000 圧縮された lena 画像（グレース画像，512×512，および8ビット/ピクセル）を用います。空間的及びプログレッシブのアクセス性能評価のために、ウェーブレット分解レベル値に3と6を用い、タイルサイズに256×256を用い画像を均等に4分割します。また、プリシントサイズに16×16の値を用います。以下に示す各性能評価において、未送信状態からの性能評価を行うために、一連の評価テストの開始前にキャッシュ内の全データを消去し、評価テストを行いました。

A. 空間的アクセス性能

文献[6]の手法は、クライアントがタイル全体のデータを受信することなく、画像の一部を表示することができます。通常のタイルベース伝送方法と文献[6]の方法の両方のアクセス性能の一例を図6に示します。このアクセス例でのビューウィンドウは、図中の中央の小さな四角であり、全てのタイルに均等に重なっています。図6から通常のタイルベース伝送方法では、各タイルの全パッケージデータを送信するのに対し、文献[6]の手法はビューウィンドウに関連したパッケージデータのみを送信しています。伝送データ量を性能指標とした空間的アクセス性能を表1に示します。表中では、空間的アクセス領域として、3種類のビューウィンドウサイズ30×30 (R5)，128×128 (R1，R2，R4) 及び256×256 (R3) と、左上のタイル領域のみのアクセス R3，R4 及び R5 を用いました。この表から、文献[6]の伝送手法はプリシントベース伝送と同じように伝送効率が良いことが分かります。また、タイルパートデータ生成に必要な ZLP，SOT，Start-Of-Data (SOD) によりデータ量が多くなるにもかかわらず、一部のアクセスでは文献[6]が勝っています。この理由は、タイルサイズに近い領域のアクセスではプリシント分割が多いと JPT-stream メッセージの数が多くなり、タイルベース伝送よりも多くのメッセージヘッダーを必要とし、送信データサイズを増大させるためです。前述したように、プリシントベース伝送方法により送信された符号データと文献[6]の伝送方法の符号データは同じであり、プリシントベース伝送では各プリシントに対して複数のバイトからなる多数のメッセージヘッダーを必要とするのに対し、文献[6]の方法はタイルパートを生成するのに必要なデータ量とメッセージヘッダーが必要です。各伝送方法でのアクセス性能は、アクセス領域とこれらの条件とのトレードオフで有意差が出てきます。

B. プログレッシブアクセス性能

文献[6]の方法は、プリシントベース伝送方法と同様に、全てのタイルデータを送信することなく、各解像度の画像にアクセス可能です。図7に文献[6]の伝送方法を用いたときの段階的な解像度アクセス例を示します。また、表2にプリシントベース伝送方法と文献[6]の伝送方法を用いたときの画像の段階的な解像度アクセス性能を示します。各解像度レベルには不足しているプリシントがないので、文献[6]の伝送方法には ZLP を必要としません。表2ではアクセスサイズが64×64を超えると、文献[6]の伝送方法はプリシントベース伝送方法に勝ります。これは、空間的アクセス性能で説明したのと同じ理由となり、符号化条件とアクセス領域に依存します。

6. おわりに

プリシנקトベース伝送時のデータ伝送効率性を持つタイルベース伝送の JPEG 2000 画像通信方式が文献[6]で提案されており、その方式、システム実装と、性能について概説しました。この提案手法では、クライアント実装にタイルベース伝送方式と同じ低いコストなり、また、プリシנקトベース伝送方式と同様に良好なデータアクセス性能を持つことを示しました。この提案方法では、ZLP とビューウインドウ領域に関連したパケットデータを用いたタイルパートデータの再構成により、タイルベース伝送を可能としています。パケットデータ毎にメッセージヘッダー付くプリシנקトベース送信ではなく、タイルパートデータの送信を用いるため、画像領域のアクセスによってはプリシנקトベース伝送よりも性能が良い場合があることも示しました。

このように、JPIP 規格を用いると、タイルベース伝送とプリシנקトベース伝送のどちらを用いても工夫によりデータ伝送効率の良い画像伝送システムを実現することが出来ますが、このデータ伝送性能はデータ伝送状況を保持するキャッシュ管理方法に依存します。HTTP ベースのクライアントサーバーシステムでは、クライアント側のキャッシュとサーバー側のキャッシュ同期問題や、サーバー側キャッシュのライフサイクル問題があります。文献[2]の JPIP 標準では、データ要求にビューウインドウという非明示的な符号データアクセス方法を用いており、データ転送能力はサーバー側のキャッシュ管理方法に依存します。

これらの問題に対応するために、明示的な符号データアクセス方法の追加という JPIP 標準の改訂が検討され、技術修正が発行されました。この技術修正によるデータアクセス方法を用いることで、サーバー側キャッシュの非明示的な管理が可能になります。JPIP 標準では、この明示的アクセス以外に、医療画像で用いられる 3-D 画像伝送やプロファイル設定などの拡張も行われています。

参考文献

- [1] M. Boliek, C. Christopoulos and E. Majani, "JPEG 2000 Part 1 020719 (Final Publication Draft)," ISO/IEC JTC 1/SC 29/WG1 N2678, July 2002.
- [2] M. Gormish, D. Lee and M Marcellin, "JPEG 2000: Overview, Architecture, and Applications," Int. Conf. on Image Processing, Vancouver, Canada, September 2000.
- [3] R. Prandolini, G. Colyer, and S. Houchin, "15444-9:2004 JPEG 2000 image coding system - Part 9: Interactivity tools, APIs and protocols – JPIP," Final Publication Draft Revision 3, ISO/IEC JTC 1/SC 29/WG 1 N3463, November 2004.
- [4] M. Gormish and S. Banerjee, "Tile-based transport of JPEG 2000 images," VLVB03, Madrid, Spain, September 2003.
- [5] D. Taubmana and R. Prandolini, "Architecture, Philosophy and Performance of JPIP: Internet Protocol Standard for JPEG2000," VCIP 2003 SPIE volume 5150 pp. 649-663, July 2003.
- [6] J. Hara, "An Implementation of JPEG 2000 Interactive Image Communication System," 2005 IEEE International Symposium on Circuits and Systems, May 2005.