

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4603947号  
(P4603947)

(45) 発行日 平成22年12月22日(2010.12.22)

(24) 登録日 平成22年10月8日(2010.10.8)

(51) Int.Cl.

H04N 1/41 (2006.01)

F 1

H04N 1/41

B

請求項の数 6 (全 23 頁)

(21) 出願番号 特願2005-213373 (P2005-213373)  
 (22) 出願日 平成17年7月22日 (2005.7.22)  
 (65) 公開番号 特開2006-101482 (P2006-101482A)  
 (43) 公開日 平成18年4月13日 (2006.4.13)  
 審査請求日 平成20年7月1日 (2008.7.1)  
 (31) 優先権主張番号 特願2004-257420 (P2004-257420)  
 (32) 優先日 平成16年9月3日 (2004.9.3)  
 (33) 優先権主張国 日本国 (JP)

(73) 特許権者 000001007  
 キヤノン株式会社  
 東京都大田区下丸子3丁目30番2号  
 (74) 代理人 100076428  
 弁理士 大塚 康徳  
 (74) 代理人 100112508  
 弁理士 高柳 司郎  
 (74) 代理人 100115071  
 弁理士 大塚 康弘  
 (74) 代理人 100116894  
 弁理士 木村 秀二  
 (72) 発明者 石川 智恵  
 東京都大田区下丸子3丁目30番2号 キ  
 ャノン株式会社内

最終頁に続く

(54) 【発明の名称】画像通信システム、サーバ装置およびその制御方法、並びにコンピュータプログラム

## (57) 【特許請求の範囲】

## 【請求項 1】

階層符号化されたJPEG2000画像データを格納したサーバと、該サーバからJPEG2000画像データをダウンロードして表示可能なクライアントと、を含む画像通信システムであって、該サーバは、

前記JPEG2000画像データに対するリクエストを前記クライアントから受信した場合、該リクエストが、画像サイズと符号化フォーマットとを指定する特定R O I要求であるか否かを判定する判定手段と、

前記判定手段により前記リクエストが前記特定R O I要求であると判定された場合に、前記JPEG2000画像データを復号し前記画像サイズにリサイズし前記符号化フォーマットの画像に変換することにより生成されるサムネイル画像データと、前記JPEG2000画像データに含まれ該JPEG2000画像データの符号化条件を示すJPEG2000メインヘッダと、を含む返送データを生成する生成手段と、

前記返送データを前記クライアントに送信する送信手段と、  
 を備えることを特徴とする画像通信システム。

## 【請求項 2】

前記サーバと前記クライアント間における通信は、JPIPプロトコルに準拠することを特徴とする請求項1に記載の画像通信システム。

## 【請求項 3】

前記クライアントは、前記返送データに含まれる前記JPEG2000メインヘッダが示す符号

化条件を利用して、前記JPEG2000画像データに対する前記サーバとのインタラクティブ通信を行うことを特徴とする請求項1又は2に記載の画像通信システム。

【請求項4】

階層符号化されたJPEG2000画像データを格納し、該JPEG2000画像データをクライアントに提供するサーバ装置であって、

前記JPEG2000画像データに対するリクエストを前記クライアントから受信した場合、該リクエストが、画像サイズと符号化フォーマットとを指定する特定ROI要求であるか否かを判定する判定手段と、

前記判定手段により前記リクエストが前記特定ROI要求であると判定された場合に、前記JPEG2000画像データを復号し前記画像サイズにリサイズし前記符号化フォーマットの画像に変換することにより生成されるサムネイル画像データと、前記JPEG2000画像データに含まれ該JPEG2000画像データの符号化条件を示すJPEG2000メインヘッダと、を含む返送データを生成する生成手段と、

前記返送データを前記クライアントに送信する送信手段と、  
を備えることを特徴とするサーバ装置。

【請求項5】

階層符号化されたJPEG2000画像データを格納し、該JPEG2000画像データをクライアントに提供するサーバ装置における制御方法であって、

前記JPEG2000画像データに対するリクエストを前記クライアントから受信した場合、該リクエストが、画像サイズと符号化フォーマットとを指定する特定ROI要求であるか否かを判定する判定工程と、

前記判定工程により前記リクエストが前記特定ROI要求であると判定された場合に、前記JPEG2000画像データを復号し前記画像サイズにリサイズし前記符号化フォーマットの画像に変換することにより生成されるサムネイル画像データと、前記JPEG2000画像データに含まれ該JPEG2000画像データの符号化条件を示すJPEG2000メインヘッダと、を含む返送データを生成する生成工程と、

前記返送データを前記クライアントに送信する送信工程と、  
を含むことを特徴とする制御方法。

【請求項6】

コンピュータを、請求項4に記載のサーバ装置の各手段として機能させるためのコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は画像通信システムおよびサーバ装置、および画像通信方法に関し、特に、サーバが保持する画像をクライアント側で縮小表示する画像通信システムおよびサーバ装置、および画像通信方法に関する。

【背景技術】

【0002】

近年のデジタルカメラの普及は目覚ましく、その画素数も膨大になり、200万～500万画素程度が一般的になっている。さらに高機能なモデルとして1000万画素クラスのものも普及し始め、今後も高精細化が進むであろうことは想像に難くない。

【0003】

さらに、デジタルカメラで撮影した画像を、オンラインでアルバムサービスを提供するサーバにアップロードして利用するユーザも増えている。このようなシステムでは、様々な種類の画像閲覧ソフトが利用されるが、一般に、クライアント側においてまずサムネイル画像による一覧表示を行い、ユーザがその中から拡大表示したい画像を選択して、サムネイル画像よりも大きいサイズによる表示を行うことが多い。

【0004】

しかし、大きな画素数からなるデジタルカメラの画像データを、その拡大表示のたびに

10

20

30

40

50

全てサーバからダウンロードすると、クライアントの端末に画像が表示されるまでに長い待ち時間が必要とする。

#### 【0005】

このような問題を解決するために、複数解像度を持ち、さらに各解像度のデータを複数のタイルに分割して符号化した画像データをサーバに保存し、表示に必要な部分データのみをダウンロードして表示する技術が開発されている。解像度を複数持つことでクライアントの表示サイズの変更に対応し、タイル分割することによって、クライアントのユーザが画像の一部の領域にランダムにアクセスすることを可能にしている。

#### 【0006】

このような技術の代表的なものとして、JPEG2000とJPIPを用いたものがある（特許文献1，特許文献2参照）。ここでJPEG2000とは、複数解像度を持ち、タイル分割されている画像符号化方式として、2001年に標準化された画像符号化方式である。JPIPとは、JPEG2000にしたがって符号化されたファイルへ断片的にアクセスし表示するためのプロトコルとして2004年9月に策定され、同年11月に標準化された、JPEG2000 image coding system - Part 9: Interactivity tools, APIs and Protocols(以下、JPIPと記す)である。

10

#### 【0007】

特許文献1には、サーバに格納してあるJPEG2000画像データの部分データを送受信するサーバ／クライアントシステムの技術が開示されている。すなわち、クライアントは、サーバに格納されているコードストリームのどの部分が、既にクライアントのバッファに格納されているかの情報に基づいて、未だバッファに格納していない圧縮データの部分をサーバに要求する。一方、サーバは、要求された圧縮データの部分データを抜き出してクライアントに返送する。最終的に、クライアント側で受信したデータと既に格納してあったデータを統合し、画像を得るものである。

20

#### 【0008】

特許文献2には、サーバ側に格納されているJPEG2000符号データを、クライアント側で高速にプログレッシブ表示するための技術が開示されている。すなわち、サーバはクライアントから指定された解像度レベルのサブバンド符号データをファイルから取り出してクライアントに送信する。一方、クライアントはサーバに欲しい解像度レベルを指定するリクエストを発行し、現在表示している画像の解像度レベルの復号化されたサブバンドの係数と、サーバから受信した符号データを復号して得られたサブバンドの係数とを合わせて画像データを復元し、プログレッシブ表示を実現するものである。

30

#### 【0009】

このような技術を使って、サーバ・クライアント間でJPEG2000の画像データを効率的に送受信するためには、上述した特許文献1および特許文献2等に記載された技術にも見られるように、クライアントが受信したデータをキャッシュ(保存)しておく必要がある。これは、JPEG2000の各解像度レベルの画像は、その解像度レベルの画像より一つ小さい解像度レベルの画像データからの差分データであるためである。既に受信した断片的なデータをクライアント側でキャッシュしておくことにより、拡大したり画質を向上させたりする際にも、受信済みのデータとの差分データのみを送受信すれば済む。これにより通信データ量を削減し、効率的な画像通信システムが実現される。

40

【特許文献1】特開2003-23630公報

【特許文献2】特開2004-40674公報

#### 【発明の開示】

#### 【発明が解決しようとする課題】

#### 【0010】

上述したように、JPEG2000の画像データの一部分を受信してサムネイル画像を得るためには、クライアントはサムネイル表示に必要な解像度レベルを指定し、サーバに要求する必要があった。

#### 【0011】

たとえば、特許文献1に記載された技術を利用するためには、クライアントはサーバが

50

格納しているコードストリームの構造を知る必要がある。そこでまず、クライアントの端末装置はサーバに対して画像データのメインヘッダを要求する。クライアントはメインヘッダ受信後にこれを解析し、該解析結果に基づいて、サムネイル画像を表示するために必要なJPEG2000の解像度レベルを特定する。そして、特定した解像度レベルのデータを取得するためのリクエストをサーバに発行してサムネイル画像を取得する。これは、通信を始めるにあたって、JPEG2000で符号化された元の画像データの解像度レベル数や各解像度レベルの画像サイズが分からぬいためである。

#### 【0012】

この方法でクライアント端末装置にサムネイル画像を表示するためには、サーバ／クライアント間で少なくとも2回の通信が発生する。ネットワークの回線速度が速くなつた今日では、statelessシステムの場合、サーバ／クライアント間においてセッションを張るために要する時間の、データの送受信に占める割合が比較的大きい。したがつて、サムネイル画像取得のために2回の通信を要していたのでは、サムネイル画像表示までの時間が長くなつてしまつという問題が生じる。

#### 【0013】

あるいは、特許文献2に記載の技術を利用する場合、クライアントが欲するサムネイル表示に必要な解像度レベルに達するまで、リクエスト／レスポンスを繰り返す必要があり、やはり複数回の通信を行わざるを得なかつた。

#### 【0014】

また、同様の問題が、ROI(Region Of Interest)取得時にも発生する。ROI部分を端末画面いっぱいに表示するためには、JPEG2000のヘッダを解析し、ROIの大きさを特定した後に、ROI表示のための画像のサイズを指定することができる。そのため、サーバ／クライアント間で少なくとも2回の通信が発生してしまう。

#### 【0015】

本発明は上述した問題を鑑みてなされたものであり、サーバ／クライアント間における1回の通信で、クライアントが所望の画像データを受信する画像通信システムを提供することを目的とする。

#### 【0016】

さらにクライアント側において、上記データ受信時に、受信した画像データの元となる画像データの符号化条件を知ることによって、該画像データに対するサーバ／クライアント間のインタラクティブな操作を、例えばJPIP通信により可能とする画像通信システムを提供することを目的とする。

#### 【課題を解決するための手段】

#### 【0017】

本発明は、上記目的を達成するための一手段として以下の構成を備える。

#### 【0018】

すなわち、階層符号化されたJPEG2000画像データを格納したサーバと、該サーバからJPEG2000画像データをダウンロードして表示可能なクライアントと、を含む画像通信システムであつて、該サーバは、前記JPEG2000画像データに対するリクエストを前記クライアントから受信した場合、該リクエストが、画像サイズと符号化フォーマットとを指定する特定ROI要求であるか否かを判定する判定手段と、前記判定手段により前記リクエストが前記特定ROI要求であると判定された場合に、前記JPEG2000画像データを復号し前記画像サイズにリサイズし前記符号化フォーマットの画像に変換することにより生成されるサムネイル画像データと、前記JPEG2000画像データに含まれ該JPEG2000画像データの符号化条件を示すJPEG2000メインヘッダと、を含む返送データを生成する生成手段と、前記返送データを前記クライアントに送信する送信手段と、を備える。

#### 【0019】

例えば、前記サーバと前記クライアント間における通信は、JPIPプロトコルに準拠することを特徴とする。

#### 【発明の効果】

10

20

30

40

50

**【 0 0 2 0 】**

本発明によれば、サーバ／クライアント間における1回の通信で、クライアントが所望の画像データを受信することができる。

**【 0 0 2 1 】**

さらにクライアント側において、上記データ受信時に、受信した画像データに付加されている、該画像データの符号化条件を示す付随データを取得することによって、該画像データに対するサーバ／クライアント間のインタラクティブな操作を、例えばJPIP通信によって効率良く行うことができる。

**【発明を実施するための最良の形態】****【 0 0 2 2 】**

以下、添付の図面を参照して、本発明をその好適な実施形態に基づいて詳細に説明する。  
。

**【 0 0 2 3 】****< 第 1 実施形態 >**

本実施形態では、既に生成済みの複数のJPEG2000ファイルを保有しているサーバが、クライアントからの要求を受け、その要求を解析し、指定されたJPEG2000画像からサムネイル画像を生成し、クライアントに送信する方法について説明する。クライアントのユーザは、Windows（登録商標）をOSとして動作するパーソナルコンピュータを使用して、アルバムサービスを提供しているホームページを開き、そこに記載されているJPEG2000画像のサムネイル画像を取得するためのリンク情報に従ってサムネイル取得の要求をサーバに送信し、JPIPの仕組みを使ってサムネイル画像を取得し、表示するものとする。

**【 0 0 2 4 】**

図1は、本実施形態のクライアント端末装置の概略構成を示す図である。CPU101は該端末装置全体の動作をコントロールし、一次記憶部102に格納されたプログラムの実行などを行う。一次記憶部102は主にメモリであり、二次記憶部103に記憶されたプログラムなどを読み込んで格納する。二次記憶部103は例えばハードディスク等が該当する。一般に、一次記憶部102の容量は二次記憶部103の容量よりも小さく、一次記憶部102に格納しきれないプログラムやデータなどは二次記憶部103に格納される。また、長時間記憶しなければならないデータなども二次記憶部103に格納される。本実施形態では、プログラムを二次記憶部103に格納しておき、プログラム実行時にこれを一次記憶部102に読み込んで、CPU101が実行する。

**【 0 0 2 5 】**

入力デバイス104としては、例えばマウスやキーボード等が該当し、プログラムなどに割り込み信号を送ったりするために用いられる。出力デバイス105としては例えば、モニタやプリンタ等が考えられる。

**【 0 0 2 6 】**

なお、クライアント装置の構成方法は他にも様々な形態が考えられるが、本発明の主眼ではないので説明を省略する。

**【 0 0 2 7 】**

図2は、本実施形態を実現するサーバ／クライアントシステムの概略を示す図である。201はクライアント1、202はクライアント2であり、図1で説明した構成を有する。クライアント1またはクライアント2のような複数のクライアントが、有線、無線を問わず、ネットワーク203によってサーバ204と通信することが可能である。

**【 0 0 2 8 】**

サーバ204は、画像ファイルを蓄積する大容量の記憶装置205を備えている。ここで記憶装置205としては例えば、ハードディスク等が該当する。この記憶装置205には、JPEG2000符号化方式により符号化された画像データが数多く格納されている。クライアント201, 202は、サーバ204が格納しているJPEG2000画像データから断片化されたデータを受信し、保存する。

**【 0 0 2 9 】**

10

20

30

40

50

次に、一般的なJPEG2000の符号化データについて、図3を用いて説明する。JPEG2000符号化データは、Main Header301と呼ばれる、画像全体の符号化条件について記述されたデータを先頭部分に有し、その後ろにタイルのデータが並ぶ。タイルデータは、Tile Header302と呼ばれるタイルの符号化条件を記述したヘッダデータを有し、その後ろにタイルの符号化データを構成するpacket303と呼ばれる符号化単位データを有する。

#### 【0030】

図3は、Layer-resolution level-component-position progression(LRCP)に沿ってpacketデータが記録された、JPEG2000ファイルの構成を示している。LRCPに準じた場合、layer / resolution / component / positionの順に記録される。このようなデータの並び方はprogression orderと呼ばれる。なお、ここで記されているpositionとは、JPEG2000符号化データにおけるprecinctに相当する。

10

#### 【0031】

progression orderとしては次の5種類が定義されており、progression orderの変更は、packetデータを並べ替えることで容易に実現できる。

#### 【0032】

- \* Layer-Resolution level-Component-Position (LRCP)
- \* Resolution level-Layer-Component-Position (RLCP)
- \* Resolution level-Position-Component-Layer (RPCL)
- \* Position-Component-Resolution level-Layer (PCRL)
- \* Component-Position-Resolution level-Layer (CPRL)

20

なお、図3に示すタイルのデータは、タイル番号が昇順になるように、JPEG2000符号化データの中で並んでいるが、タイルのデータはタイル番号順に並ぶ必要は無い。

#### 【0033】

また図3では、タイルのデータは一つのまとまったデータとして説明したが、実際には、各タイルを複数のタイルパートに分割することができる。したがってタイルヘッダには、タイル番号とタイルパート番号の2つの番号が記載されている。タイルとタイルパートの関係を、図11に示す。

#### 【0034】

図4に、JPEG2000符号化データにおける解像度スケーラビリティ、すなわち解像度(画像サイズ)とresolution番号との関係を示す。最も小さい解像度の画像のresolution番号を0とし、resolution番号が一つ増加するごとに画像の幅と高さが2倍になっていく。

30

#### 【0035】

また、各layer内には、resolution番号の小さい順にデータが格納されている。Layer番号は復元する画像の原画に対するS/N比に対応し、layer番号が小さいほどS/N比が悪くなる。一つのJPEG2000ファイル内でのresolution番号とlayer番号、component番号の最大値は、エンコーダによって予め設定され、そのパラメータにしたがってエンコードされており、その情報は符号化データ内に格納されている。

#### 【0036】

本実施形態においてサーバが保有しているJPEG2000画像データは3つの解像度からなり、それぞれは以下のようなサイズである。

40

#### 【0037】

```
Resolution level 0 ... 400 × 300[pixel]
Resolution level 1 ... 800 × 600[pixel]
Resolution level 2 ... 1600 × 1200[pixel]
```

このようなデータを持つサーバに対して、クライアントが $320 \times 240$ [pixel]程度のサムネイル画像を取得するべく、JPIPの定義に従って次のようなコマンドをサーバに送信した場合について考える。

#### 【0038】

```
http://www.image.com/JPIP.cgi?target=sample.jp2&roi=ganon-thumbnail&fsize=320,240&type=image/jpeg
```

50

ここで、送信文字列内の"target"は、リソース名を指定するフィールドであり、"roi"はROI領域の名前を指定するフィールド、"fsiz"は画像全体の大きさを指定するフィールド、"type"は返送データ形式を指定するフィールドである。したがってクライアントは、sample.jp2画像の"ganon-thumbnail"という名前のROI領域の画像データを要求している。その際、画像全体が320,240[pixel]程度の大きさであり、返送されるデータ形式はJPEGであることが指示されている。なお、"ganon-thumbnail"というROIは、特別なデータを要求するための特別コマンドであり、本実施形態のサーバ／クライアント間では、サムネイル画像の符号化データの後ろに、JPEG2000符号化データのmain headerデータがアペンドされたデータを意味するものと定義されている。

## 【0039】

10

以下、上述したようなデータのやり取りを行うサーバの動作を、図5のフローチャートを用いて説明する。

## 【0040】

まずステップS501において、サーバはクライアントからのリクエスト文字列を受信する。ここではリクエスト文字列として、以下の文字列を受信したとする。

## 【0041】

<http://www.image.com/JPIP.cgi?target=sample.jp2&roi=ganon-thumbnail&fsiz=320,240&type=image/jpeg>

するとステップS502において、サーバはステップS501で受信したリクエスト文字列を解析する。本実施形態の場合、サーバは、sample.jp2画像の"ganon-thumbnail"という名前のROI領域の画像データを、画像全体が320,240[pixel]程度の大きさになるように要求されており、さらに、返送する時のデータ形式はJPEGであることが、この解析により得られる。

20

## 【0042】

次にステップS503では、ステップS502で解析したリクエストがROI要求を含んでいたか否かを判断する。ROI要求がある場合にはステップS504へ、ROI要求が無ければステップS508へ処理を進める。上記リクエスト文字列の場合、"ganon-thumbnail"と言う名称のROIが要求されているので、ステップS504へ進む。

## 【0043】

30

ステップS504では、要求されたROIがサムネイル画像に対応しているか否かを判断する。サムネイル画像に対応するROI要求であればステップS505へ、それ以外のROI要求であればステップS508へ処理を進める。本実施形態のサーバにおいては、"ganon-thumbnail"というROIがサムネイル画像として定義されていることを予め知っているため、ステップS505へ進む。

## 【0044】

ステップS505では、ステップS502におけるリクエスト文字列の解析によって得られた指定サイズ、具体的にはfsizフィールドで指定されたサイズになるように画像全体をリサイズし、やはりリクエスト文字列の解析によって得られた返送データ形式、具体的にはtypeフィールドで指定された返送データ形式に変換してサムネイル画像を作成する。

## 【0045】

40

この例では、fsiz=320,240と指定されているので、この値を参考にリサイズすることになる。サーバが保有しているJPEG2000画像データは、

```
Resolution level 0 ... 400 × 300[pixel]
Resolution level 1 ... 800 × 600[pixel]
Resolution level 2 ... 1600 × 1200[pixel]
```

というサイズからなるため、アスペクト比が4:3である。ここでfsizの値も320 : 240 = 4 : 3であるので、指定されたサイズへのリサイズが可能である。したがってこの場合、横が320[pixel]、縦が240[pixel]のJPEG画像に変換する。

## 【0046】

もし、リクエストで指定された画像のアスペクト比と、オリジナルのJPEG2000画像のア

50

スペクト比が異なる場合には、オリジナルのJPEG2000画像のアスペクト比に合わせてリサイズし、JPIPでの定義に従って、リクエストされたサイズと返送されたサイズが異なることを示すJPIPレスポンスヘッダ（JPIP-fsiz）によって、クライアントに通知すればよい。例えば、fsiz=320,320というリクエストに対して、JPEG2000画像のオリジナルサイズが $1600 \times 1200[\text{pixel}]$ であった場合には、サーバは $320 \times 240[\text{pixel}]$ にリサイズした画像を返送データとし、JPIPレスポンスヘッダとして、"JPIP-fsiz: 320,240"と一緒に返送する。

#### 【0047】

なお、リサイズの方法としては、Resolution level 2までデコードしてから所望のサイズにリサイズしてもよいし、所望のサイズに最も近いResolution level 0( $400 \times 300[\text{pixel}]$ )までをデコードしてからリサイズしても良い。リサイズ方法については本件の主眼ではないので、詳細な説明は割愛する。10

#### 【0048】

次にステップS506では、要求されたJPEG2000画像からmain header部301を抜き出す。本実施形態の場合、"target"フィールドで指定された"sample.jp2"のJPEG2000符号化データから、main header部301を抜き出す。次いでステップS507では、ステップS505で作成したサムネイル画像データの後に、ステップS506で抜き出したオリジナルのJPEG2000画像データのmain headerデータをアpendした後、ステップS510へ進む。

#### 【0049】

一方、ステップS503にてROI要求がない、あるいはステップS504でサムネイルに対するROI要求でなかったと判断されると、処理はステップS508に進む。ステップS508では、ステップS502で得られたリクエスト文字列の解析結果から要求されたデータを特定し、JPEG2000符号化データからそれらのデータを抜き出す。次にステップS509では、ステップS508で抜き出したデータに基づき、指定された返送データ形式のレスポンスデータを作成し、ステップS510へ進む。なお、ステップS508およびステップS509は通常のJPIPサーバの処理であり、本件の主眼ではないので、ここでの詳しい説明は割愛する。20

#### 【0050】

ステップS510では、ステップS507またはステップS509で作成したデータをレスポンスデータとして返送し、サーバの処理を終了する。

#### 【0051】

このように本実施形態のリクエストによれば、その文字列における"anon-thumbail"がサムネイル画像を意味するROIであると認識できないサーバであっても、通常のROI要求として処理され、 $320 \times 240[\text{pixel}]$ 程度の大きさの、main headerを伴わないJPEG画像が返送される。したがって、クライアントではサムネイル画像を適切に表示することができる。30

#### 【0052】

また、サーバがROI要求自体を処理できない場合でも、ROIを処理できなかったことを通知するJPIPヘッダ"JPIP-roi:no-roi"と、roiフィールド以外のリクエスト文字列より $320 \times 240[\text{pixel}]$ 程度の大きさのJPEG画像を返送するので、やはりクライアントにおいてサムネイル画像の表示が可能になる。

#### 【0053】

また、クライアントが本発明に対応しておらず、サーバから返送されたJPEG画像データの後にmain header部がアpendされていることを認識できない場合であっても、送信されたデータをデコードしてサムネイル画像を表示することが可能である。40

#### 【0054】

もちろん、本発明に対応しているクライアントであれば、サーバから返送されたJPEG画像データの後にアpendされた、オリジナルのJPEG2000符号化データのmain header部を抜き出し、これを解析することによって、サムネイル表示後に拡大操作等を行う際にも、JPIPを用いたインタラクティブな通信をスムーズに開始することができる。

#### 【0055】

以上説明したように本実施形態のサーバ／クライアントシステムによれば、クライアン50

トは、一度の通信で、階層符号化データへ特別コマンド（例えば上記実施形態のサムネイル取得コマンド）に基づいた処理が施された画像（同じくサムネイル画像）及びオリジナルのJPEG2000符号化データのmain headerを取得できる。そのため、サムネイル表示後にJPIPを用いたインタラクティブな通信を行う際にも、既に受信済みであるmain headerを解析することで、表示したいJPEG2000画像のデコード条件を知ることができ、JPIP通信によって必要な部分データを要求することができる。

#### 【0056】

##### <第2実施形態>

以下、本発明に係る第2実施形態について説明する。上述した第1実施形態では、クライアント毎にサムネイル画像のサイズを指定できるため、サーバはリクエスト受信ごとに、サムネイル画像を作成する必要があった。第2実施形態では、サーバノクライアント共に本発明に対応しているシステムであることが予め分かっている場合に、サムネイル画像サイズおよび返送データ形式を予め決めておくことを特徴とする。10

#### 【0057】

第2実施形態では、ROIの名称が"ganon-thumbnail"である場合に、 $320 \times 320[\text{pixel}]$ に納まるサイズを有するJPEG画像に対し、第1実施形態と同様に、オリジナルのJPEG2000符号化データから抜き出されたmain header部をアpendしたデータが返送されるものとする。この場合、サムネイル画像を必要とするクライアントが発行するリクエストは、以下のようになる。

#### 【0058】

<http://www.image.com/JPIP.cgi?target=sample.jp2&roi=ganon-thumbnail>

このとき、サーバの処理として第1実施形態で示した図5と異なるのは、ステップS505～S507の部分である。

#### 【0059】

以下、第2実施形態におけるサーバの処理について、図6のフローチャートを用いて説明する。なお、図6において図5と同様の動作を行うステップには同一ステップ番号を付し、説明を省略する。

#### 【0060】

まず、ステップS501～S504では第1実施形態と同様の動作を行い、ステップS504においてROIの要求がサムネイルの要求であると判断されると、ステップS601へ進む。30

#### 【0061】

ステップS601では、予め用意されたサムネイル画像を取得する。このサムネイル画像は、第1実施形態で示したように、サムネイル画像データの後ろに、オリジナルのJPEG2000符号化データから抜き出したmain header部がアpendされているデータである。このサムネイル画像は同じファイル内に格納されていても良いし、別ファイルとして用意し、何らかの仕組みによってオリジナル画像と対応付けられていても良い。別ファイルとオリジナル画像を対応付ける仕組みとは、例えば、"sample.jp2"に対しては"sample.jpg"というように、同一名称で拡張子を異ならせたファイルでも良いし、全く関係のないファイル名とオリジナルのJPEG2000画像データとをテーブルを用いて関連付けても良い。

#### 【0062】

第2実施形態の例においては、"sample.jp2"に対応するサムネイル画像データが取得された後、処理はステップS510へ進む。以下、ステップS508, S509, S510については第1実施形態と同様の動作を行うため、その説明は省略する。

#### 【0063】

ここで、本発明に対応していないサーバに対して、本発明に対応するクライアントから第2実施形態のようなリクエストが発行された場合、該サーバは、一般的なJPIPサーバと同様の振る舞いをする。つまり、"fsiz"フィールドの無いリクエストを受信したJPIPサーバは、画像データを返送せず、main headerデータとファイルのヘッダデータのみを返送する。したがってクライアント側においては、少なくともmain headerデータを取得することができるため、本発明に対応していないサーバノクライアントシステムと同様の動作50

をすることになる。

#### 【0064】

以上説明したように第2実施形態によれば、サーバ側において予めサムネイルデータを用意しておくことにより、上述した第1実施形態よりもサーバの処理負荷が少なく、データを返送するまでの時間を短縮することができる。

#### 【0065】

<第3実施形態>

以下、本発明に係る第3実施形態について説明する。

#### 【0066】

第3実施形態では、ROIの名称が"ganon-thumb nail"である場合に、main headerとresolution level 0のデータをオリジナルのJPEG2000符号化データから抜き出し、JPEG2000符号化データとして返送するものとする。 10

#### 【0067】

この場合、上述した第1、第2実施形態とは異なり、オリジナルのJPEG2000符号化データから抜き出されたmain headerが既に存在するので、その後ろに、更にオリジナルのJPEG2000符号化データから抜き出したmain headerを再度挿入する必要は無い。したがって、第2実施形態と異なるのは、図6のステップS601で用意されるthumb nail画像である。

#### 【0068】

第3実施形態で返送されるサムネイル画像のデータを、図12の左側に示す。図12の右側は、オリジナルのJPEG2000符号化データである。図12の右側に示されるように、各タイルのresolution level 0に相当するデータを、タイルパート番号0のデータとしてまとめ、各タイルのタイルパート0のみをJPEG2000符号化データの先頭部分にまとめておくことで、容易に、図12の左側に示されるようにサムネイル画像を抜き出し、返送することができる。図12の右側に示すような符号化データは、解像度方向のスケーラビリティを持つよう、progression orderをRLCPまたはRPCLとすれば、容易に作り出すことが可能である。 20

#### 【0069】

また、図12左側のJPEG2000データは、図12右側のオリジナルのJPEG2000符号化データから一部分を切り取った状態であるので、main headerの示す符号化条件とその後ろに続くデータとを比較すると、データが不足している状態になってしまう。しかし、一般的に、JPEG2000のクライアントは、符号化データの先頭部分を切り取ったような状態の符号化データであっても、デコード可能である。 30

#### 【0070】

第3実施形態によれば、クライアント毎に画像サイズを変更することができない。さらに、返送されるサムネイル画像は、エンコード時のresolution level 0のサイズに依存するため、一律にあらかじめ決められた画像サイズに合わせておくこともできない。

#### 【0071】

しかしながら第3実施形態によれば、サーバ側において別符号化データとしてサムネイルデータを用意する必要がない。さらに、オリジナルのJPEG2000符号化データの先頭部分を抜き出して、そのまま返送データとすることができますので、データを返送するまでの時間を短縮することができる。さらに、返送されるJPEG2000符号化データのmain headerは、オリジナルのJPEG2000符号化データのmain headerであるため、クライアントは、オリジナルのJPEG2000符号化データの符号化条件を知ることができ、サムネイル取得後のインタラクティブな通信にスムーズに移行することができる。 40

#### 【0072】

以上説明したように第3実施形態によれば、JPEG2000コードストリームで返送したり、JP2ファイル形式等に変換して返送したりすることによって、クライアント側のデコーダを1種類にすることが可能である。

#### 【0073】

<第4実施形態>

以下、本発明に係る第4実施形態について説明する。

**【 0 0 7 4 】**

上述した第1～第3実施形態では、サムネイル画像を一度のリクエストで取得する方法について説明したが、ROI領域が表示画面に収まるサイズの画像データを取得する際にも、第1～第3実施形態と同様の手法を適用することによって、一度のリクエスト発行での取得が可能となる。

**【 0 0 7 5 】**

第4実施形態においては、既に生成済みの複数のJPEG2000ファイルを保有しているサーバが、クライアントからの要求を受けて該要求を解析し、指定されたJPEG2000画像からROI領域を含む画像を適切なサイズにリサイズし、クライアントに送信する方法について説明する。クライアントのユーザは、携帯端末を使用してホームページを開き、そこに記載されたJPEG2000画像を取得するために、画像名に対応する文字列が記載されているリンク情報をクリックし、表示画面にROI領域がちょうど納まるサイズにリサイズされた画像の要求をサーバに送信し、JPIPの仕組みを使用して画像を取得し、表示する。10

**【 0 0 7 6 】**

第4実施形態においてサーバが保有しているJPEG2000画像データは3つの解像度からなり、それぞれは以下のようなサイズである。

**【 0 0 7 7 】**

Resolution level 0 ... 400 × 300[pixel]

Resolution level 1 ... 800 × 600[pixel]

Resolution level 2 ... 1600 × 1200[pixel]

20

さらにこの画像データは、Resolution level 2において、256 × 256[pixel]のタイルに分割されているものとする。図7に、第4実施形態における画像データの構成例を示す。同図において斜線で示される領域701が、ROI領域として定義されているものとする。このROI領域701は、ROI description boxでROIとして定義されていても良いし、符号化データ自体がROIとしてエンコードされていても良いが、ここでは、ROI description boxの最初にROIとして記述されている領域とする。

**【 0 0 7 8 】**

このような画像データを有するサーバに対して、クライアントは、画像内でROIとして定義されている部分を320 × 240[pixel]の領域に納まるサイズにリサイズした時の画像全体を取得するべく、JPIPの定義に従って、次のようなコマンドをサーバに送信したものとする。30

**【 0 0 7 9 】**

<http://www.image.com/JPIP.cgi?target=example.jp2&roi=ganon-download&fsiz=320,240&type=image/jpeg>

30

このリクエストをJPIPの定義に従って解釈すると、クライアントは"sample.jp2"画像の"ganon-download"という名称のROI領域の画像データを要求しており、その際、画像全体が320,240[pixel]程度の大きさであり、返送されるデータ形式はJPEGである旨が指示されている。

**【 0 0 8 0 】**

しかしながら、本発明に対応するサーバ／クライアント間では、"ganon-download"というROIは、特別な意味を含むROIとして定義された特別コマンドであり、このROI要求を含むリクエストの場合には、"ganon-download"で指定されたROIを、"fsiz"で指定された大きさに収まるように画像全体をリサイズし、指定した返送画像データに変換し、さらに、その返送画像データの後ろにオリジナルのJPEG2000符号化データのmain headerをappendしたデータを、クライアントへ送り返すように定義されている。40

**【 0 0 8 1 】**

以下、第4実施形態におけるサーバの動作を、図8のフローチャートを用いて説明する。なお、上述した第1実施形態で示した図5と同様の動作を行うステップには同一ステップ番号を付し、説明を省略する。

**【 0 0 8 2 】**

50

まず、ステップS501～S503では第1実施形態と同様の動作を行い、ステップS801において、ステップS502の解析で得られたROI要求が特別なROI要求であるか否かを判断する。特別なROI要求であればステップS802へ、それ以外のROI要求であればステップS508へ進む。第4実施形態においては、受信したクライアントからのリクエストに特別なROI要求としての"ganon-download"が含まれているため、処理はステップS802へ進む。

#### 【0083】

ステップS802では、要求された特別なROIがステップS502の解析によって得られた指定サイズに収まるように画像全体をリサイズし、指定された返送画像データ形式に変換する。第4実施形態の場合、fsiz=320,240であるから、ROI領域が $320 \times 240[\text{pixel}]$ に収まるように画像をリサイズして、JPEG形式に変換する。具体的には、ROI領域は図7の701で示される領域であるので、領域701を含む4タイル(タイル9, 10, 16, 17)が $320 \times 240[\text{pixel}]$ に収まるようにリサイズされる。したがって、タイル9, 10, 16, 17の4タイルが $240 \times 240[\text{pixel}]$ になるように、画像全体を $750 \times 563[\text{pixel}]$ にリサイズし、JPEGに変換する。10

#### 【0084】

ステップS506では、第1実施形態と同様に、オリジナルのJPEG2000符号化データからmain header部を抜き出す。そしてステップS803において、ステップS506で抜き出したmain header部をステップS802で作成した画像データの後ろにアpendする。第4実施形態においては、ステップS802で作成した $750 \times 563[\text{pixel}]$ のJPEG画像データの後ろに、ステップS803で抜き出したexample.jp2のmain headerをアpendする。

#### 【0085】

そしてステップS804において、JPIPレスポンスヘッダを作成する。すなわち、JPIP-roiヘッダとして、返送画像データの全体のサイズおよび、ROIとして特定されるべき領域の位置を示す。具体的には、ステップS802で作成された画像と、その中のROIの位置関係が図9に示すようになるので、レスポンスヘッダは以下のように作成される。20

#### 【0086】

JPIP-roi: roi=ganon-download; fsiz=750,563; rsiz=240,240; roff=240,120

ここで、"roi="に続く文字列はROIの名称、"fsiz="に続く2つの整数は返送画像データの全体のサイズ、"rsiz="に続く2つの整数はROI領域のサイズ、"roff="に続く2つの整数はROI領域の位置を、それぞれ示す。

#### 【0087】

以下、ステップS508, S509, S510については第1実施形態と同様に、一般的なJPIPと同様の動作を行うため、その説明は省略する。30

#### 【0088】

このように第4実施形態のリクエストによれば、"ganon-download"が特別なROI領域の返送形式を意味するROIであると認識できないサーバであっても、通常のROI要求として処理することができる。例えば、"ganon-download"が図7の701で指定される領域であると認識されれば、画像全体を $320 \times 240[\text{pixel}]$ 程度に収めた場合に、"ganon-download"で表されるROI領域を抜き出して返送する。つまり、main headerを伴わないROI領域のみのJPEG画像が返送される。

#### 【0089】

また、サーバがROI要求自体を処理できない、あるいは、"ganon-download"という名前のROIを特定できない場合でも、ROIを処理できなかったことを通知するJPIPヘッダ"JPIP-roi:no-roi"と、roiフィールド以外のリクエスト文字列より $320 \times 240[\text{pixel}]$ 程度の大きさに画像全体をリサイズしたJPEG画像を返送するので、やはり、クライアントにおいて所望の画像を表示することが可能になる。40

#### 【0090】

なお、ROIが複数ある場合には、複数のROIを包含する領域を"ganon-download"に対応する領域として定義しても良い。

#### 【0091】

また、クライアントが本発明に対応しておらず、サーバから返送されたJPEG画像データ

50

の後ろにmain header部がアペンドされていることを認識できない場合であっても、送信されたデータをデコードして画像を表示することが可能である。さらに、JPIPレスポンスヘッダの記述により、返送画像データのどの位置にROI領域が含まれているかを判断することができる。

#### 【0092】

もちろん、本発明に対応しているクライアントであれば、サーバから返送されたJPEG画像データの後ろにアペンドされた、オリジナルのJPEG2000符号化データのmain header部を抜き出し、これを解析することによって、受信したJPEGデータの表示後に拡大操作等を行う際にも、JPIPを用いたインタラクティブな通信をスムーズに開始することができる。

#### 【0093】

以上説明したように第4実施形態のサーバ／クライアントシステムによれば、クライアントは一度の通信で適切なROIサイズを表示可能とする画像データを取得でき、なおかつ、オリジナルのJPEG2000符号化データのmain headerも同時に取得できる。そのため、画像表示後にJPIPを用いたインタラクティブな通信を行う際にも、既に受信済みであるmain headerを解析することで、表示したいJPEG2000画像のデコード条件を知ることができ、JPIP通信によって必要な部分データを要求することができる。

#### 【0094】

なお、第4実施形態においても上述した第2実施形態のように、サーバとクライアント間で予めROIのサイズを決定しておき、サーバ側で予め"roi=ganon-download"に対応する画像データを作成しておくことも有効である。この場合、画面サイズが異なるクライアントには対応できないものの、返送データを再エンコードによって作成する必要がないため、サーバのレスポンス速度の向上が期待できる。

#### 【0095】

さらに、上述した第3実施形態のように、サーバが画像毎にROIのサイズを決定しておき、サーバ側で予め"roi=ganon-download"に対応するresolution level Rを決定し、resolution level Rまでのデータをタイルパートとして抽出し、オリジナルJPEG2000符号化データの先頭に配して置くことも有効である。つまり、"roi=ganon-download"に対応する画像データとして、resolution level Rのタイルパートまでを、オリジナルのJPEG2000符号化データの先頭から抜き出し、返送データとしてクライアントに送信する。この場合、画像サイズはエンコード時の各resolution levelのサイズに依存するため、一律にあらかじめ決められた画像サイズに合わせておくことはできないが、サーバ側において別符号化データとしてサムネイルデータを用意する必要がないため、サーバの記憶容量の削減につながる。さらに、データを返送するまでの時間を短縮することができる。

#### 【0096】

##### <第5実施形態>

以下、本発明に係る第5実施形態について説明する。

#### 【0097】

上述した第4実施形態では、ROI領域が表示画面に収まるサイズの画像全体のデータを一度のリクエスト発行によって取得する例を説明したが、同様の手法を適用して、ROI領域のデータのみを取得することも可能である。

#### 【0098】

第5実施形態では、ROIの名前が"ganon-download"である場合に、対応するROI領域を抜き出し、 $320 \times 240[\text{pixel}]$ に納まるサイズを持つJPEG画像に、上述した各実施形態と同様に、オリジナルのJPEG2000符号化データから抜き出されたmain headerをアペンドしたデータが返送されるものとする。この場合、ROI画像を必要とするクライアントが発行するリクエストは、以下のようになる。

#### 【0099】

<http://www.image.com/JPIP.cgi?target=example.jp2&roi=ganon-download&rsiz=320,240&type=image/jpeg>

上記のリクエストにおいて、"rsiz"フィールドは、画像全体から抜き出す領域の大きさ

10

20

30

40

50

を指定するフィールドである。一般にJPIPでは、"fsiz"フィールドによって画像全体の大きさを指定し、その大きさを基準にして、抜き出す矩形領域の大きさを"rsiz"フィールドによって、抜き出す矩形領域の左上の点を"roff"フィールドによって指定することができる。

#### 【0100】

以下、第5実施形態におけるサーバの動作を、図10のフローチャートを用いて説明する。なお、上述した第4実施形態で示した図8と同様の動作を行うステップには同一ステップ番号を付し、説明を省略する。

#### 【0101】

まず、ステップS501～S503およびステップS801では、第4実施形態と同様の動作を行う。次いでステップS1001において、特別なROIに対応する領域を画像データから抜き出す。ここで特別なROI領域とは、第4実施形態と同様に図7の領域701であるものとする。すなわち、領域701を含む4タイル(タイル9, 10, 16, 17)が抜き出される。

10

#### 【0102】

次にステップS1002では、ステップS1001で抜き出したROI領域を、ステップS502の解析によって得られた指定サイズに収まるようにリサイズし、指定された返送画像データ形式に変換する。第5実施形態の場合、rsiz=320,240であるから、タイル9, 10, 16, 17より構成される、抜き出した領域701を320×240[pixel]に収まるサイズに変更し、JPEG形式に変換する。具体的には、タイル9, 10, 16, 17の4タイルの領域が240×240[pixel]サイズのJPEG画像に変換される。

20

#### 【0103】

以下、ステップS803, S804およびステップS508, S509, S510については第4実施形態と同様に、一般的なJPIPと同様の動作を行うため、その説明は省略する。

#### 【0104】

なお第5実施形態においても第4実施形態と同様に、ステップS1002で作成された画像と、その中のROIの位置関係が図9に示すようになるので、第4実施形態と同様の、以下に示すJPIPレスポンスヘッダが返送される。

#### 【0105】

JPIP-roi: roi=ganon-download; fsiz=750,563; rsiz=240,240; roff=240,120

30

このように第5実施形態のリクエストによれば、"ganon-download"が特別なROI領域の返送形式を意味するROIであると認識できないサーバであっても、通常のJPIPの要求として処理することができる。この場合、リクエスト内に"fsiz"フィールドが記載されていないため、これを受信したJPIPサーバは画像データを返送せずに、main headerデータとファイルのヘッダデータのみを返送する。したがってクライアント側においては、少なくともmain headerデータを取得することができるため、本発明に対応していないサーバ／クライアントシステムと同様の動作をすることになる。

#### 【0106】

また、サーバがROI要求自体を処理できない、あるいは、"ganon-download"という名前のROIを特定できない場合でも、roiフィールド以外のリクエスト文字列により作成されたデータを返送するため、ROIを処理できなかったことを通知するJPIPヘッダ"JPIP-roi:noroi"と共に、main headerデータとファイルのヘッダデータのみを返送することになる。

40

#### 【0107】

なお、ROIが複数ある場合には、複数のROIを包含する領域を"ganon-download"に対応する領域として定義しても良い。

#### 【0108】

また、クライアントが本発明に対応しておらず、サーバから返送されたJPEG画像データの後ろにmain header部がアpendドされていることを認識できない場合であっても、送信されたデータをデコードして画像を表示することが可能である。

#### 【0109】

もちろん、本発明に対応しているクライアントであれば、サーバから返送されたJPEG画

50

像データの後にアペンドされた、オリジナルのJPEG2000符号化データのmain header部を抜き出し、これを解析することによって、受信したJPEGデータの表示後に拡大操作等を行う際にも、JPIPを用いたインタラクティブな通信をスムーズに開始することができる。

#### 【0110】

以上説明したように第5実施形態のサーバ／クライアントシステムによれば、クライアントは一度の通信で適切なサイズで表示可能なROI画像データを取得でき、なおかつ、オリジナルのJPEG2000符号化データのmain headerも同時に取得できる。そのため、画像表示後にJPIPを用いたインタラクティブな通信を行う際にも、既に受信済みであるmain headerを解析することで、表示したいJPEG2000画像のデコード条件を知ることができ、JPIP通信によって必要な部分データを要求することができる。

10

#### 【0111】

なお、第5実施形態においても上述した第2実施形態のように、サーバとクライアント間で予めROIのサイズを決定しておき、サーバ側で予め"roi=ganon-download"に対応する画像データを用意しておくことも有効である。この場合、画面サイズが異なるクライアントには対応できないものの、返送データを再エンコードによって作成する必要がないため、サーバのレスポンス速度の向上が期待できる。

#### 【0112】

また、第5実施形態を第4実施形態と組み合わせることも有効である。つまり、"roi=ganon-download"と"fsiz"が指定されている場合には画像全体を返送し、"roi=ganon-download"と"rsiz"が指定されている場合には画像の部分領域を返送するようなシステムが考えられる。

20

#### 【0113】

なお、第4実施形態および第5実施形態共に、ROI領域の抜き出しをJPEG2000画像のタイル境界に沿って行う例を示したが、タイル境界に関わらずROIに接する領域を切り抜いても良い。

#### 【0114】

上述した各実施形態においては、返送データをデコードが簡単なJPEG画像データ形式とする例を説明したが、ロスレス圧縮が可能なPNG形式等、他の形式であっても構わない。

#### 【0115】

また、第1実施形態と第4実施形態、あるいは第2実施形態と第4実施形態を組み合わせたシステム、あるいは、第1実施形態と第5実施形態、第2実施形態と第5実施形態を組み合わせたシステムも、実現可能である。

30

#### 【0116】

##### <第6実施形態>

第1から第5実施形態では、サーバ側の動作について説明した。第6実施形態では、第1実施形態のサーバから特別コマンドによって取得した、JPEG画像データとJPEG2000のメインヘッダを受信したクライアントの、データ受信後の動作について説明する。

#### 【0117】

本実施形態におけるクライアントは、第1実施形態と同様に、リクエスト文字列

<http://www.image.com/JPIP.cgi?target=sample.jp2&roi=ganon-thumbnail&fsiz=320,240&type=image/jpeg>

40

をサーバに送信することで、横が320[pixel]、縦が240[pixel]のJPEG画像と、その後ろにアペンドされたオリジナルのJPEG2000画像データのmain headerデータを受信したものとする。このときのクライアントの動作について、図13のフローチャートを参照して説明する。

#### 【0118】

まず、ステップS1301では、サーバからのレスポンスデータを受信する。ステップS1302では、サーバからのレスポンスデータのうち、JPIPヘッダ部分を解析する。本実施形態では、サーバ側が、特別コマンド"roi=ganon-thumbnail"を含むクライアントからのリクエストを解釈し、320×240[pixel]のサムネイル画像を返送してきたため、JPIPヘッダを送

50

信する必要がない。従って、ステップS1302では、返送データにJPIPヘッダが含まれていないと解析できる。ステップS1303では、ステップS1302の解析結果より、サーバが特別コマンドを処理できたかどうかを判断する。処理できたと判断した場合には、ステップS1304へ進み、処理できなかったと判断した場合には、ステップS1306へ進む。

#### 【0119】

本実施形態では、ROIを処理できなかったことを通知するJPIPヘッダ"JPIP-roi: no-roi"が、返送データになかったため、サーバは、特別コマンドroi=anon-thumbailを理解し処理したものと判断され、ステップS1304へ進む。ステップS1304では、オリジナルのJPEG2000画像データのmain headerを受信画像データから分離する。本実施形態の場合、JPEG画像データの後ろにあるmain headerデータを分離することになる。つまり、受信したバイナリデータの一番最後のバイトから、データの先頭部分に向かってスキャンし、JPEG2000のメインヘッダの開始を示すSOCマーカ(0xFF4F)を探す。そして、SOCマーカより後ろの部分を、JPEG2000のmain header、SOCマーカの一つ前のデータまでをJPEG画像データとして分離する。JPEG2000 main headerの分離方法は、この他にも様々な方法が考えられるが、本件の主眼ではないので、詳細な説明は省略する。10

#### 【0120】

次に、ステップS1305において、ステップS1304で分離したJPEG2000 main headerをキャッシュする。このキャッシュ方法についても様々な方法が考えられるが、本件の主眼ではないので詳しい説明は割愛する。

#### 【0121】

次に、ステップS1306において、受信したJPEG画像をデコードして表示し、このフローを終了する。

#### 【0122】

図13に示す動作によって、クライアントの端末は、アルバムサービスを提供しているホームページの画像のサムネイル一覧を表示することができる。

#### 【0123】

次に、サムネイル一覧の中から、クライアントが一枚選択し、JPEG2000画像データをJPIPを使って取得する際のクライアントの動作について、図14のフローチャートを参照して説明する。

#### 【0124】

まず、ステップS1401では、クライアントが選択した画像に該当するJPEG2000のキャッシュされたmain headerを解析し、JPEG2000のエンコード条件を調べる。これにより、最大画像サイズと、解像度数(Resolution level数)、レイヤ数(画質数)、1コンポーネント当たりのビット数、およびコンポーネント数、タイル分割数など、画像の基本的な情報が分かる。本実施形態の場合、クライアントの選択した画像は、1つのレイヤ、3つの解像度を持ち、各解像度は、30

```
Resolution level 0 ... 400 × 300[pixel]
Resolution level 1 ... 800 × 600[pixel]
Resolution level 2 ... 1600 × 1200[pixel]
```

のような画像サイズを持つものとする。

#### 【0125】

次に、ステップS1402では、クライアントの表示要求を取得する。本実施形態の場合、サムネイル画像から選択した場合には、1024 × 768[pixel]の領域に画像全体が収まるデータを、最高画質で要求するものとする。ステップS1403では、ステップS1401で解析した結果と、ステップS1402で取得したクライアントの表示要求から、JPIPリクエストを作成し、サーバに送信する。JPIPのリクエスト作成方法は、一般的なJPIPクライアントで行われる動作であるので、ここでは詳細な説明は省略する。本実施形態の場合、40

```
http://www.image.com/JPIP.cgi?target=sample.jp2&fsiz=800,600&type=image/jpp-stream&need=r0,r1
```

のようなJPIPリクエストがサーバに発行される。

10

20

30

40

50

**【 0 1 2 6 】**

ステップS1404では、サーバからJPIPレスポンスデータを受信する。ステップS1405では、ステップS1404で受信したレスポンスデータを解析し、キャッシュに書込む。JPIPレスポンスデータの解析方法は、一般的なJPIPクライアントで行われる動作であるので、ここでは詳細な説明は省略する。さらに、キャッシュへの書き込み方法も、様々な方法が考えられるが、本件の主眼ではないので、詳細な説明は省略する。

**【 0 1 2 7 】**

ステップS1406では、キャッシュデータから必要なJPEG2000画像データを作成する。本実施形態の場合、キャッシュから、main headerデータと、resolution level 0, resolution level 1データを抜き出すことで、必要なJPEG2000画像データを作成することができる。ステップS1407では、ステップS1406で作成されたJPEG2000データをデコードし表示する。ステップS1408では、クライアントから新たな表示要求があるかどうか判断する。新たな表示要求があれば、ステップS1402へ処理を戻し、新たな表示要求がなければ、このフローを終了する。

10

**【 0 1 2 8 】**

このようにクライアント側で、オリジナルのJPEG2000 main headerをキャッシュし、その後のJPIPを使った画像要求の際に利用することで、JPEG2000データの部分転送にスムーズに移行することができる。したがって、転送時間の短縮およびレスポンス応答時間の短縮を図ることができる。

**【 0 1 2 9 】**

20

また、本実施形態では、実施形態1のサーバ応答に対するクライアントの振る舞いについて述べたが、実施形態2、実施形態4および実施形態5のサーバ返送データに対して、同様の処理をクライアント側で行うことができる。

**【 0 1 3 0 】**

実施形態3のサーバのレスポンスデータの処理では、図13のステップS1304における、サーバから受信したバイナリデータから、JPEG2000のmain headerデータを抜き出す処理が不要になる。実施形態3のレスポンスデータは、オリジナルのJPEG2000データのresolution level 0を抜き出したデータである。従って、そのまま、受信したデータすべてをキャッシュすることで、その後のJPIPリクエストにスムーズに移行することができる。

**【 0 1 3 1 】**

30

このため、クライアントは2回目以降のリクエストにおいて、main headerとresolution level 0以外のデータを要求すればよいことになる。

**【 0 1 3 2 】**

なお、以上の実施形態では、公知のJPEG2000の符号化データが複数の解像度を表現可能に階層符号化していることに着目して、サーバからクライアントへの1回の返送データに「サムネイル（低解像度画像）」及び「JPEG2000のメインヘッダ」を含ませる場合を説明したが、本発明はこれに限らない。例えばこのJPEG2000は解像度（Resolution level）だけでなく、レイヤ（layer level）においても階層的に符号化されている。従って、上記解像度の概念をレイヤに置き換えて考え、例えば、クライアントからサーバへの要求に応じて、サーバからクライアントへの1回の返送データに「低レイヤで構成された画像データ（S/N比の悪い画像データ）」及び「JPEG2000のメインヘッダ」を含ませる場合も、上記実施形態の場合と同様にその後のインタラクティブな通信が効率的に行える。よって、これら変形例も本発明の範疇に含まれる。

40

**【 0 1 3 3 】****[ 他の実施形態 ]**

なお、本発明は、複数の機器（例えばホストコンピュータ、インターフェイス機器、スキーナ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複合機など）に適用してもよい。

**【 0 1 3 4 】**

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラム

50

コードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

10

#### 【0135】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0136】

本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフロー チャートに対応するプログラムコードが格納されることになる。

20

#### 【図面の簡単な説明】

#### 【0137】

【図1】本発明に係る一実施形態におけるコンピュータのシステム構成を示すブロック図である。

【図2】本実施形態におけるサーバ／クライアントシステムの構成を示す図である。

【図3】一般的なJPEG2000の符号化データの構成を示す図である。

【図4】JPEG2000における解像度スケーラビリティを説明する図である。

【図5】第1実施形態におけるサーバ処理を示すフロー チャートである。

【図6】第2実施形態におけるサーバ処理を示すフロー チャートである。

【図7】第4実施形態におけるサーバ側のJPEG2000画像のタイル分割の様子を示す図である。

30

【図8】第4実施形態におけるサーバ処理を示すフロー チャートである。

【図9】第4実施形態における返送画像データとROI領域の位置関係を示す図である。

【図10】第5実施形態におけるサーバ処理を示すフロー チャートである。

【図11】JPEG2000符号化データにおけるタイルとタイルパートの関係を示す図である。

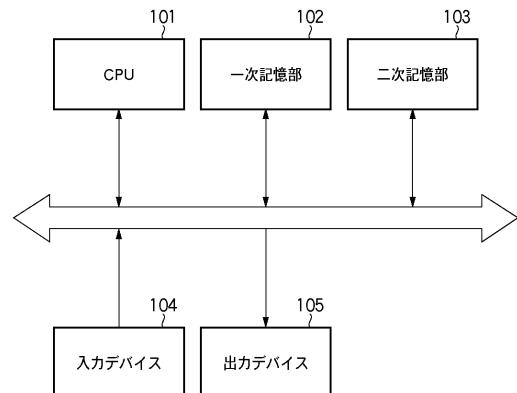
【図12】第3実施形態における返送データとサーバ側のJPEG2000画像データの様子を示す図である。

【図13】第6実施形態におけるクライアントの受信データ処理を示すフロー チャートである。

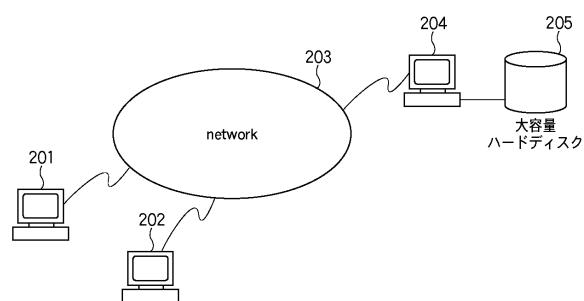
【図14】第6実施形態におけるクライアントのJPIP要求フローを示すフロー チャートである。

40

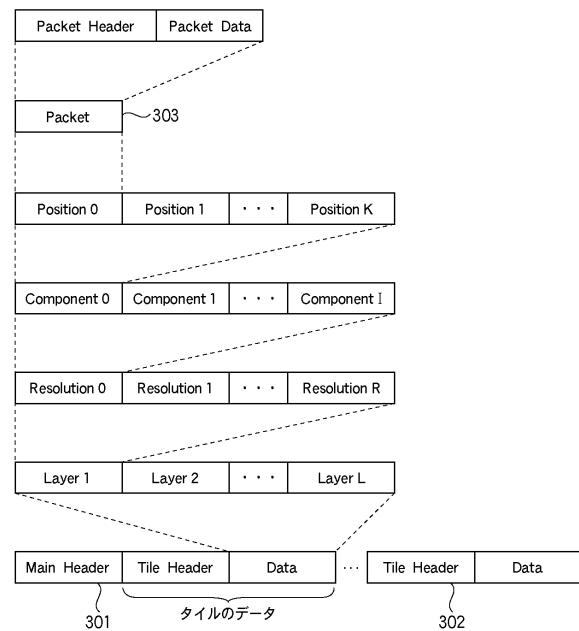
【図1】



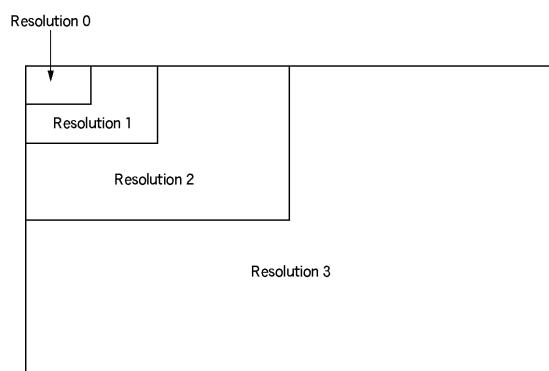
【図2】



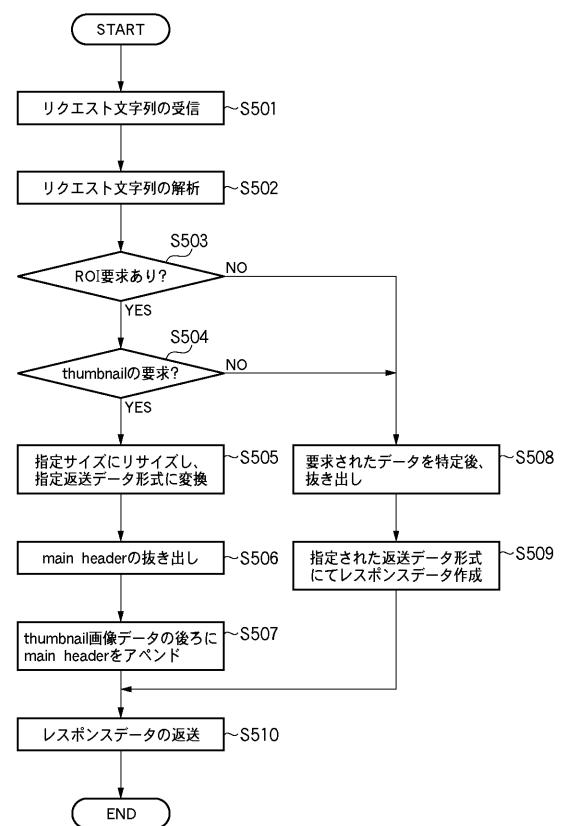
【図3】



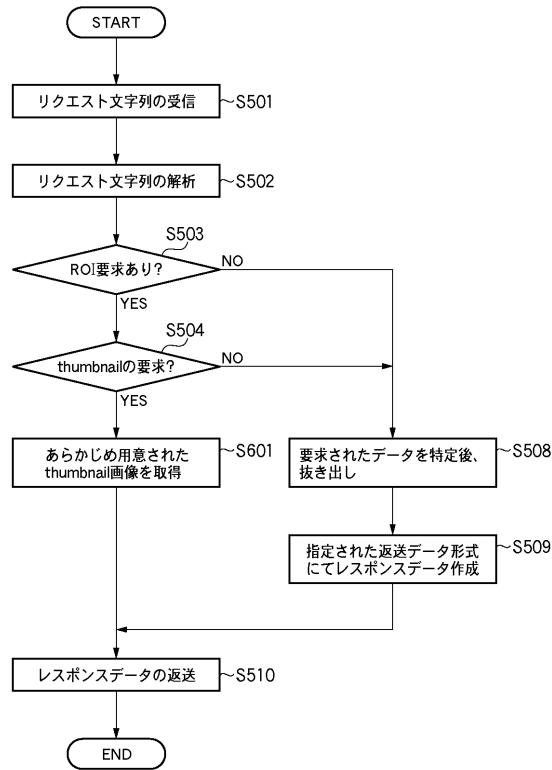
【図4】



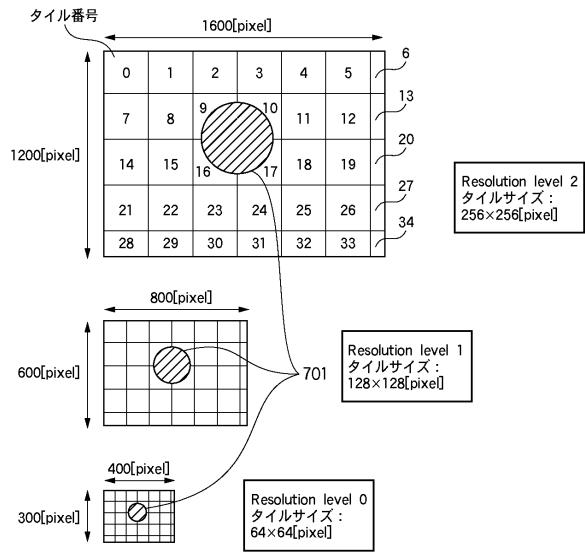
【図5】



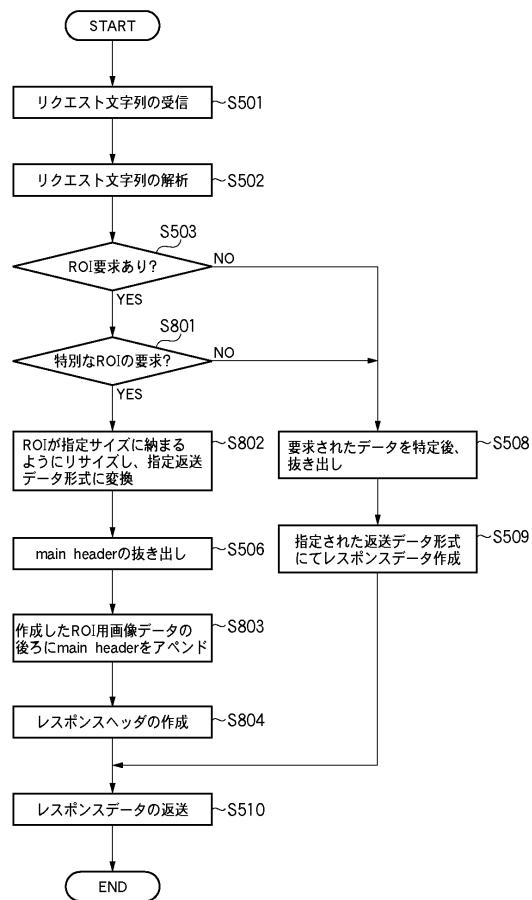
【図6】



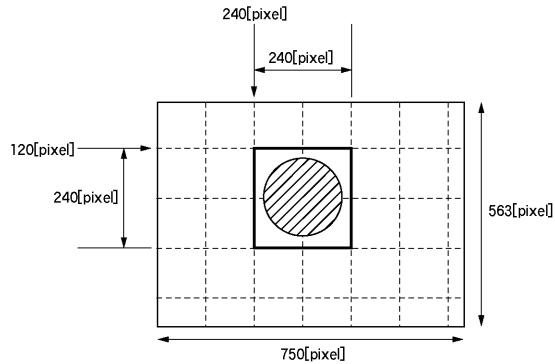
【図7】



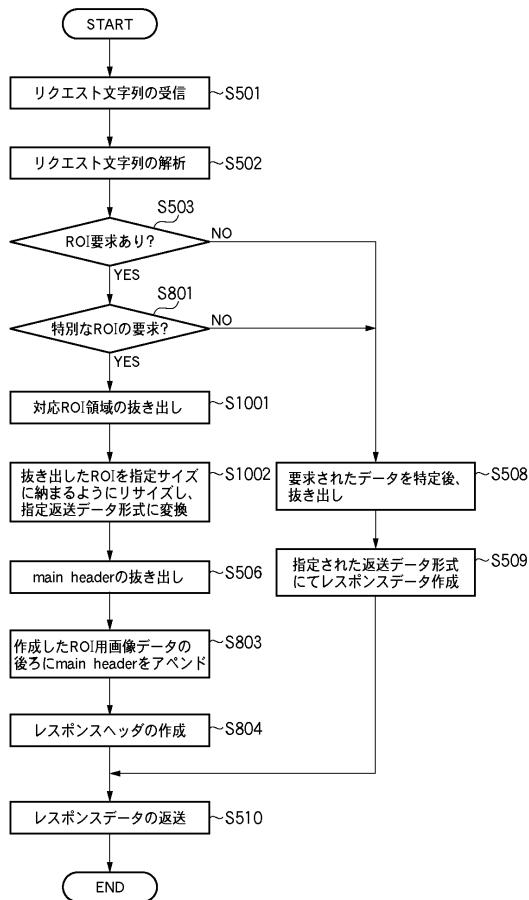
【図8】



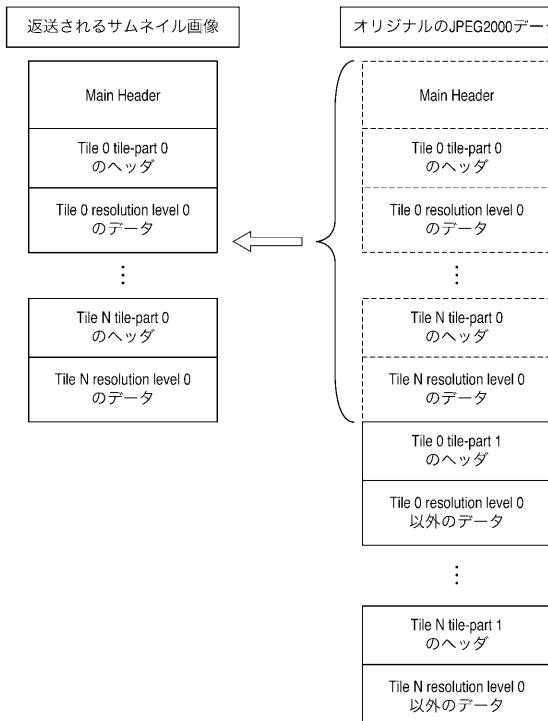
【図9】



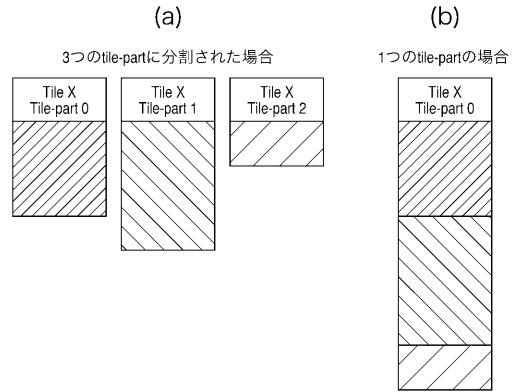
【図10】



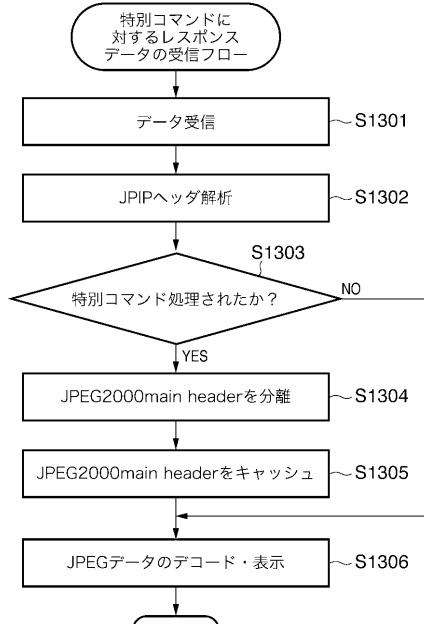
【図12】



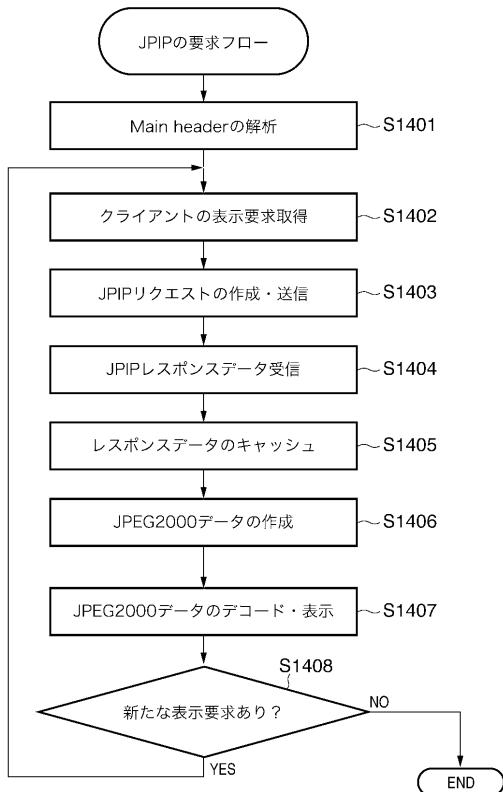
【図11】



【図13】



【図14】



---

フロントページの続き

審査官 松尾 淳一

(56)参考文献 特開平06-062256(JP,A)  
特開平06-233276(JP,A)  
特開平10-105480(JP,A)  
特開平11-196379(JP,A)  
特開2001-136371(JP,A)  
特開2002-185961(JP,A)  
特開2003-023630(JP,A)  
特開2003-024445(JP,A)  
特開2004-040674(JP,A)  
特開2005-027080(JP,A)  
特開2005-086723(JP,A)  
特開2005-101677(JP,A)  
特開2005-303854(JP,A)  
特許第3082958(JP,B2)  
特許第3126484(JP,B2)

(58)調査した分野(Int.Cl., DB名)

H04N 1/00  
H04N 1/21  
H04N 1/41 - 1/419