

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2017-126376

(P2017-126376A)

(43) 公開日 平成29年7月20日(2017.7.20)

(51) Int.Cl.	F I	テーマコード (参考)
GO6F 3/03 (2006.01)	GO6F 3/03 400A	5E555
GO6F 3/0488 (2013.01)	GO6F 3/0488 130	

審査請求 有 請求項の数 13 O L (全 161 頁)

(21) 出願番号	特願2017-75945 (P2017-75945)	(71) 出願人	000139403
(22) 出願日	平成29年4月6日 (2017.4.6)		株式会社ワコム
(62) 分割の表示	特願2016-219334 (P2016-219334) の分割	(74) 代理人	100115738
原出願日	平成26年11月19日 (2014.11.19)		弁理士 鷲頭 光宏
(31) 優先権主張番号	61/906,334	(74) 代理人	100121681
(32) 優先日	平成25年11月19日 (2013.11.19)		弁理士 緒方 和文
(33) 優先権主張国	米国 (US)	(74) 代理人	100130982
(31) 優先権主張番号	61/908,647		弁理士 黒瀬 泰之
(32) 優先日	平成25年11月25日 (2013.11.25)	(72) 発明者	ペクトフ プラメン
(33) 優先権主張国	米国 (US)		ブルガリア共和国 1303 ソフィア
(31) 優先権主張番号	61/973,161		オソゴボ通り 38 ワコム ヨーロッパ
(32) 優先日	平成26年3月31日 (2014.3.31)		ゲーエムペーハー内
(33) 優先権主張国	米国 (US)		

最終頁に続く

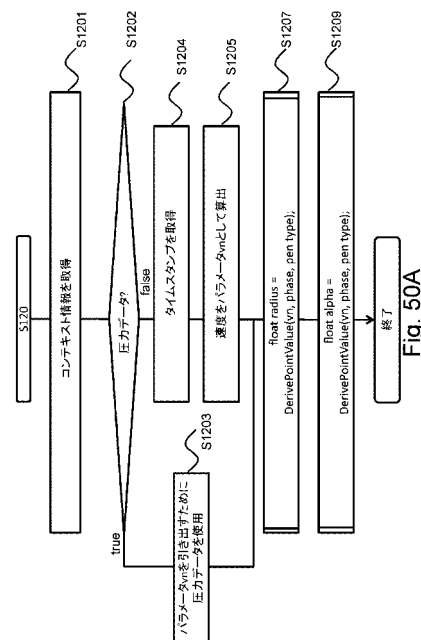
(54) 【発明の名称】 インクデータ出力方法及びインクデータ出力装置

(57) 【要約】

【課題】リアルな外観を有するストロークをレンダリング (描画) するために、非装置依存型のインクデータを生成し、使用可能とする。

【解決手段】インクデータを出力する方法であって、指示体の動きに伴い検出された位置データに基づいて、動きの軌跡を所定の曲線補間により示す制御点データを生成するステップと、位置データが筆圧データを伴うか否かを判定し、位置データが筆圧データを伴う場合には筆圧データに基づいて、位置データが筆圧データを伴わない場合には位置データの変位量に基づいて、制御ポイントの形成する曲線の透明度データを導出する導出ステップ (ステップS1202～S1209) と、制御点データに透明度データを関連付けてポイントデータを出力する出力ステップとを含む。

【選択図】 図50A



【特許請求の範囲】

【請求項 1】

インクデータを出力する方法であって、

指示体の動きに伴い検出された位置データに基づいて、前記動きの軌跡を所定の曲線補間により示す制御点データを生成するステップと、

前記位置データが筆圧データを伴うか否かを判定し、

前記位置データが筆圧データを伴う場合には前記筆圧データに基づいて、前記位置データが筆圧データを伴わない場合には前記位置データの変位量に基づいて、前記制御点データの形成する曲線の透明度データを導出する導出ステップと、

前記制御点データに前記透明度データを関連付けてポイントデータを出力する出力ステップと

10

を含むインクデータ出力方法。

【請求項 2】

前記出力ステップは、前記制御点データの示す位置座標と前記透明度データとを 1 つの前記ポイントデータに含めて出力する

請求項 1 記載のインクデータ出力方法。

【請求項 3】

前記出力ステップは、複数の前記ポイントデータを 1 つのストロークデータに含めて出力する

請求項 2 記載のインクデータ出力方法。

20

【請求項 4】

出力された前記ストロークデータは、前記透明度データに基づいて透明度が変化する可変透明度のストロークデータである

請求項 3 記載のインクデータ出力方法。

【請求項 5】

前記導出ステップは、前記位置データが筆圧データを伴わない場合に、前記位置データの変位量と前記位置データに関する時刻情報とに基づいて前記透明度データを導出する

請求項 1 に記載のインクデータ出力方法。

【請求項 6】

前記導出ステップは、前記位置データ及び前記位置データに関する時刻情報に基づいて、前記ストロークデータを構成する前記複数のポイントデータのそれぞれにおける速度に対応するパラメータを導出し、前記速度に対応するパラメータが増加すると前記透明度データの透明度の値が増加する関数または不透明度の値が減少する関数を用い、前記透明度データを決定する関数ステップを含む

30

請求項 3 に記載のインクデータ出力方法。

【請求項 7】

前記関数ステップは、前記指示体のタイプに応じて、異なる特徴を有する複数の関数の中から前記関数を選択する

請求項 6 に記載のインクデータ出力方法。

【請求項 8】

前記位置データが筆圧データを伴う場合に、前記位置データの変位量及び前記筆圧データに基づいて前記透明度データを導出する

40

請求項 1 に記載のインクデータ出力方法。

【請求項 9】

前記所定の曲線補間は、キャットマルーロム曲線補間である

請求項 1 に記載のインクデータ出力方法。

【請求項 10】

前記制御点データは、2次元座標または3次元座標で示される

請求項 1 に記載のインクデータ出力方法。

【請求項 11】

50

インクデータを出力するインクデータ出力装置であって、
指示体の動きに伴い検出された位置データに基づいて、前記動きの軌跡を所定の曲線補間により示す制御点データを生成するステップと、
前記位置データが筆圧データを伴うか否かを判定し、
前記位置データが筆圧データを伴う場合には前記筆圧データに基づいて、前記位置データが筆圧データを伴わない場合には前記位置データの変位量に基づいて、前記制御点データの形成する曲線の透明度データを導出する導出ステップと、
前記制御点データに前記透明度データを関連付けてポイントデータを出力する出力ステップと
を実行するインクデータ出力装置。

10

【請求項 1 2】

指示体の動きの軌跡を再現するためのストロークデータを含むインクデータを出力する方法であって、
前記指示体の動きに伴い検出された位置データに基づいて、前記動きの軌跡をキャットマルーロム曲線を用いる補間により示すための制御点データを生成するステップと、
前記位置データに伴い取得された筆圧データに基づいて、前記制御点データの位置における透明度データを導出する導出ステップと、
前記制御点データの位置と前記透明度データとを含むポイントデータを複数含むストロークデータを出力することで、可変透明度の前記ストロークデータを出力する出力ステップと
を含むインクデータ出力方法。

20

【請求項 1 3】

前記制御点データの位置での曲線の透明度データは、前記制御点データの位置の透明度又は不透明度の値として示される

請求項 1 2 に記載のインクデータ出力方法。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は、手書き（フリーハンド）のストロークデータのパスを再生し、該パスをスタイルとともに描写するインクデータを生成し、レンダリングし、操作し、伝達するための方法及びシステムに関する。

30

【背景技術】**【0002】**

各種の手書き入力システムが知られている。手書き入力システムは、ユーザがペン形の装置を利用することによって手書き（フリーハンド）のデータを入力することを可能にするものである。例えば、電磁共鳴タイプのペンタブレット入力システムが知られており、これは、筆圧及びペンの傾きデータに関連するデータを含む手書きデータの入力を可能にする。さらなる例としては、静電タイプのペン入力システムが知られており、これは、指とセンサ表面との間に静電容量が生成されるのと同じように、道具と（タブレット）センサ表面との間に静電容量を生成する。さらに、測定された位置のコレクションから導出されるジェスチャー情報のような相対的にシンプルな情報を出力する入力システムもまた知られている。

40

【0003】

典型的には、ペン型の道具によって入力された手書きデータまたはストローク（パス又はトレース）データは、ピクセルデータや画像データのようなラスターデータを生成するために、単一の描画アプリケーション内で使用可能である。インクメッセージング、インク保管・回復アプリケーション、電子メール、写真注釈、遠隔ビデオ会議アプリケーションなどのような様々なタイプの装置及びアプリケーションを操作することによって生成された手書きデータまたはストロークが各種装置間で共有されることを可能にする方法及びシステムのニーズが存在する。デジタルインク又はインクデータ（以下、「インクデータ

50

」と称する)は、そのようなニーズに向けたものとして提案される。典型的には、使用されるアプリケーションの設定に従って生成される直接的なピクセルデータや画像データのようなラスターデータは、入力装置におけるユーザのストローク入力操作をサポートするために使用される。一方で、インクデータは中間データであり、ストロークデータのラスター化に先駆けて存在し、様々なアプリケーションによって使用可能なベクトルデータの形態を有している。インクデータの種類のサンプルが以下の非特許文献 D 1 ～ D 4 に記述されている。

【先行技術文献】

【非特許文献】

【0004】

【非特許文献 1】W3C、レコメンデーション 20、2011 年 9 月、「インクマークアップ言語(Ink Markup Language) (InkML)」、URL: <http://www.w3.org/TR/2011/REC-InkML-20110920/>

【非特許文献 2】マイクロソフトコーポレーション他、「インク・シリアライズド・フォーマット仕様書(Ink Serialized Format Specification)」、2007 年、URL: [http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/InkSerializedFormat\(ISF\)Specification.pdf](http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/InkSerializedFormat(ISF)Specification.pdf)

【非特許文献 3】W3C ワーキングドラフト 11 (W3C Working Draft 11)、2014 年 2 月、「スケーラブルベクトルグラフィックス(Scalable Vector Graphics) (SVG) 2」、URL: <http://www.w3.org/TR/SVG2/>、W3C レコメンデーション、2011 年 8 月 16 日、「スケーラブルベクトルグラフィックス(Scalable Vector Graphics) (SVG) 1.1 (第 2 版)」、URL: <http://www.w3.org/TR/2011/REC-SVG11-20110816/>

【非特許文献 4】W3C、「HTML5・HTML 及び XHTML のための語彙及び関連する API (A vocabulary and associated APIs for HTML and XHTML)・W3C レコメンデーション 2014 年 10 月 28 日 (W3C Recommendation 28 October 2014)」、URL: <http://www.w3.org/TR/html5/>

【非特許文献 5】スレート・コーポレーション他、「JOT-インクの格納及び交換フォーマットのための仕様書(A Specification for an Ink Storage and Interchange Format)」、バージョン 1.0、1996 年 9 月

【発明の概要】

【発明が解決しようとする課題】

【0005】

手短に言うと、InkML (非特許文献 1) と ISF (非特許文献 2) のデータ構造は、色々なアプリケーション間で共有できる態様で、ペン型の装置によって入力されるストロークデータを表す。SVG (非特許文献 3) は、入力装置としてどのタイプのペン装置が利用されたかということに関わらず、ユーザによって入力された制御点によって定義されるパスをベクトルデータとして描画することを可能にするウェブ標準を提供する。

【0006】

非特許文献 1 ～ 4 に記載されたインクデータはいずれも、ペン又は指の動きによって形成されるトレース (又はパス) を再生するために必要な幾何学的情報を定義するものである。そのような情報を、ここでは「ストロークデータ」と総称する。

【0007】

非特許文献 1 は、現在最も広く知られたインクデータを記述している。非特許文献 1 は、「トレース」と呼ばれるオブジェクトを次のように定義する。すなわち、「トレース」はユーザがデジタルインクを書いたときのペンの軌跡を記録するために使用される基本要素である。

【0008】

例えば、

```
<ink><trace>x1 y1, x2 y2, .....xn yn </trace></ink>
```

は、点 x_1 , y_1 から点 x_2 , y_2 まで延在するストロークオブジェクトのパスを記述

する。

【0009】

非特許文献2は、マイクロソフト（登録商標）ウィンドウズ（登録商標）のアプリケーション上で使用可能なインク機能によって生成されるインクデータを記述している。非特許文献2は、「ストローク」と呼ばれるオブジェクトを次のように定義する。すなわち、「単純な例で先に述べたように、ストロークは、ISFにおいて最も根本的でありかつ重要なプロパティである。ストロークは、ストローク内の個々の点を構成するパケットデータを含み、潜在的には、他のストロークごとのプロパティも同様に含む。」と定義する。

【0010】

非特許文献3は、ペン入力を仮定するものではないが、各種のブラウザ及び描画ソフトによってサポートされるベクトルデータの標準を記述している。非特許文献3は、「パス」と呼ばれる情報を次のように定義する。すなわち、「パスは、埋められることができ、ストロークされることができ、クリッピングパスとして使用されることができ、或いは、これら3つの任意の結合であり得る形状の輪郭を表す。」と定義する。SVG（非特許文献3）では、パスオブジェクトは、当該分野ではよく知られたポリベジェ（キュービクベジェ、二次ベジェ）曲線のような補間曲線に基づいて補間される。

【0011】

例えば、

```
<path stroke="green" stroke-width="5" d="M100,200 C100,100 300,100 300,200"/>
```

は、開始制御点（100，200）から始まり、終了制御点（300，200）までのパスを記述する。このパスは、2つの制御点（100，100）及び（300，100）を利用するものであり、パス幅「5」及びカラー「グリーン」を有している。

【0012】

非特許文献4は、補間曲線を生成するために例えば二次曲線コマンド及びベジェ曲線コマンドを使用することができる「カンバスパス」と呼ばれるクラスを定義する。

【0013】

本記述においては、用語「ストロークオブジェクト」は、上記非特許文献1～4の「トレース」、「ストローク」、「パス」、及び「カンバスパス」を包含する一般用語として使用される。

【0014】

ストロークオブジェクトは、そのデータ構造がペン又は指の動きによって形成されるトレース（又はパス）を再生するために集合的に使用される一組のポイント又は制御点座標を含むベクトルデータ情報である。様々な実施の形態によれば、本発明は、様々なアプリケーション上で、又は、様々なアプリケーション間で手書き（フリーハンド）のストロークデータを表すインクデータを生成し、操作し（例えば、スライスし）、レンダリングし、伝達するための方法及びシステムを提案する。各実施の形態は、上記非特許文献1～4の従来技術では利用可能でなかった技術的解決法を提供する。以下の記述は全体として発明の4つの実施の形態を開示するように組み立てられる一方、当業者にとっては明らかなことであるように、各実施の形態の様々な側面が、各実施の形態の間で結合され、補足され、交換され、切り替えられ、変更され得、それによってさらなる実施の形態が生み出され得ることに注意すべきである。例えば、各実施の形態の様々な方法及びシステムは、インクデータの定義はもちろん、インクデータを生成し、再生し、描画し（レンダリングし）、操作し、及び伝達する方法、並びに、インクデータの構造（データオブジェクト及びデータフォーマット）について、本明細書に開示される一以上の他の実施の形態に関して説明されるものを採用することができる。

【0015】

以下の第1乃至第4の実施の形態のそれぞれは、様々な例において、以下に記述される一以上の側面を扱う。

【課題を解決するための手段】

【0016】

〔側面 1〕いくつかのコンピュータにおいて既存のストロークオブジェクトを部分的又は全体的に変形する操作オブジェクトの導入。

【0017】

一側面によれば、本発明は、操作オブジェクトを提供することに向けられる。上述した以前から知られているインクデータモデルは、1つのストロークオブジェクトを1つの集合として処理するために、静的なストロークデータを処理するためにのみ使用される意味(semantics)又は文法(syntax)を含む。したがって、以前から知られているインクデータモデルは、ストロークオブジェクトの部分を選択し又はスライスするという能力を有しない。また、以前から知られているインクデータモデルは、1つのプロセッサ上でのストロークオブジェクトの操作を可能とするものの、ストロークオブジェクト上で実行される操作(例えば、編集)オペレーションを複数のプロセッサが共有することを可能にする能力を有しない。

10

【0018】

図91は、本発明の一実施の形態による操作オブジェクト270の一例である「スライス」オブジェクトを図示している。ストロークオブジェクトの一部を操作する(スライスする)能力を有するスライスオブジェクト274が生成され、転送される。図示した例においては、1つのコンピュータ上で1つのストロークオブジェクト9101の一部がスライスされ、そのスライスされた部分を示す操作データ9103が他のコンピュータによって、その他のコンピュータ上においてもストロークオブジェクト9101が同じように操作され得るように、共有される。ストロークオブジェクトの一部を変形又は操作(例えばスライスすること)は、本発明の第1及び第4の実施の形態において後に詳細に記述されるであろう。インクデータが編集され、更新された状態を複数のコンピュータの間で共有するためにそれらの間で1つの操作オブジェクト270を共有することは、本発明の実施の第1、第2、及び第4の実施の形態において後に詳細に記述されるであろう。

20

【0019】

〔側面 2〕装置の違いを吸収するためにペンイベント入力情報の定義を抽象化すること(及び、SVGのペン入力表現能力を改善するためにSVGをよりペン入力由来志向とすること)。

【0020】

さらなる側面によれば、本発明は、異なる入力装置の間に存在する任意の違いを吸収するために、手書き入力データを抽象化することに向けられる。これは、筆圧やペン角度の情報のようなストロークの既存の入力属性を、新規のモデル内に定義されるより高いレベルの概念による属性に抽象化することによって達成される。一般的に、手書きの入力データに基づいて再生される必要のある情報は、ペン(スタイラス)がどの角度で保持されていたか、どの座標の時点でどの点が取得されたか、どれだけの筆圧が適用されたか、というような手書きデータが「どのように」入力されたかという情報ではない。代わりに、キャプチャされる必要のある情報は、ある筆圧、ペン速度などで実施されたそのようなペンオペレーション又は描画オペレーションの「結果」を再生することのできるベクトルデータである。

30

【0021】

現在、筆圧、ペン角度、ペン回転角データなどを取得可能な高性能入力装置(例えば、図92の9202C)から、指による入力を受け付ける能力を有する一方で筆圧やペンの傾き角を取得する能力を有しない、広く使用されている静電タブレット又は他のよりシンプルな装置など(例えば、図92の9202A)まで、様々な手書き入力装置が存在している。したがって、手書き入力データの装置依存型の任意の属性を、ペンイベントの「結果」を再生するために使用され得る非装置依存型の抽象化されたベクトルデータ(図92の9204)に変換することが望まれる。そのような抽象化された形式で定義されるインクデータは、最終的に図92の9208に示されるようなラスタデータ(画像データ)を生成するために、ベクトルデータ内にまとめられる。上で議論したSVG1.1(非特許文献3)はベクトルデータを定義するもので、図92の9206に示されている。SV

40

50

G 1. 1 (非特許文献 3) は、ストロークの幅、色、及び透明度 (不透明度) を変えたり調節したりすることを許容しておらず、結果として、ペンイベントの「結果」を再生することに特には適していない。また、S V G は、ベジエ曲線を生成するために使用される制御点のようなストロークオブジェクトのパス座標データ以外のデータを含んでおり、したがって、特殊化された描画アプリケーション以外の様々なアプリケーション 9 2 2 0 での使用には適していない。

【0 0 2 2】

ラスター画像データ (図 9 2 の 9 2 0 8) を生成することに加えて、ベクトル内のより抽象化された形式にインクデータをまとめることは、サイン認証アプリケーションや注釈アプリケーション内での使用という点から、また望ましいことである。その際、抽象化は画像由来であり過ぎないことが好ましいが、ラスター形式およびベクトル形式の両方においてインクデータを定義するために使用され得る抽象化属性とはなるべきである。タイプ 1 (筆圧を含む) 及びタイプ 2 (筆圧を含まない) の装置依存型ペンイベントデータ 9 2 0 2 を一般化されたインクデータ、すなわち図 9 2 の中間データ 9 2 0 4 に抽象化することは、本発明の第 1 及び第 3 の実施の形態において以下に詳しく記述されるであろう。

【0 0 2 3】

〔側面 3〕フォーマットから言語 (情報モデル) を分離することによって、インクデータのライフサイクルを延ばすこと。

【0 0 2 4】

例えば、デジタル写真のようなラスターデータの内容は、しばしば、単一の装置や単一のアプリケーションだけでなく、複数の装置及びアプリケーションによって使用され、特定の「エコシステム」において、連鎖的にすべての間で共有され又は転送される (もっとも、それらは J P E G、G I F、T I F F などのような様々なフォーマットで処理され得る)。これらの様々なフォーマットが使われ得るのは、ラスターデータがピクセル値の集合を概念的に記述する一般的な情報モデルを含んでいるからである。

【0 0 2 5】

さらなる側面によれば、本発明は、共通言語 (ストローク言語 (S L)) の採用に基づき、異なるフォーマット間でのインクデータの交換及び転送を促進することに向けられる。ストローク言語 (S L) は、本発明のインクデータの意味を定義する情報モデルであり、インクデータのフォーマットと対立するものである。すなわち、例えば抽象化された属性によって定義されるインクデータは、色々なラスター画像フォーマット (P N G、J P E G など) に処理され、色々なベクトルグラフィックフォーマット (S V G、I n k M L、H T M L 5 など) に置換され、ストローク構造を定義する色々なストリームフォーマット (I S F、I n k M L など) で生産される。図 9 3 は、概念的に本発明のこの側面を描写するものである。入力フォーマットタイプとともに出力フォーマットタイプに柔軟性を加え、多様な出力及び入力のフォーマットタイプに適応させるために、共通言語 (又は共通言語を定義する情報モデル) は、言語を生成するデバイスドライバのレベルと、生成された言語がファイルやパケットに出力される際の出力レベルとの中間に位置することが好ましい。特に、本発明の様々な実施の形態によるインクデータ処理部 1 0 0 は、2 つの分離された構成要素として、抽象化された言語 (ストローク言語) に基づいてインクデータを生成するインクデータ生成部 1 2 0 と、インクデータの入出力を扱うインクデータ整形部 1 4 0 とを含む。インクデータ生成の機能と、入力／出力のためのインクデータ整形の機能とが分離されていることから、インクデータ処理部 1 0 0 は、様々な装置の間にインクデータの使用を広げるためのインクデータエコシステムの基礎的要素として使用されることに適している。本発明のこの側面は、第 4 の実施の形態にて以下に詳細に記述されるであろう。

【0 0 2 6】

図 9 1 ~ 図 9 3 に描写されたこれら本発明の 3 つの側面は、本発明の第 1 ~ 第 4 の実施の形態を以下で記述した後、再度、議論されるであろう。

【図面の簡単な説明】

【0027】

【図1】本発明の様々な実施の形態による、インクデータが生成され、利用されるシステムの全体を説明する図である。

【図2】本発明の実施の形態における使用に適したインクデータ構造の実体関連図である。

【図3A】複数のポイントオブジェクトによって定義されるストロークオブジェクトを説明する図である。

【図3B】2つの異なる描画スタイルオブジェクトによる、図3Aに示したストロークオブジェクトの2つのレンダリング（描画）結果のうちの1つを説明する図である。

【図3C】2つの異なる描画スタイルオブジェクトによる、図3Aに示したストロークオブジェクトの2つのレンダリング（描画）結果のうちの他の1つを説明する図である。

【図4A】ストロークオブジェクトを選択し変形する（例えば回転する）ために使用される「選択」操作オブジェクトの動作を説明する図である。

【図4B】ストロークオブジェクトをスライスするために使用される「スライス」操作オブジェクトの動作を説明する図である。

【図5】本発明の第1の実施の形態によるインクデータ処理装置の機能ブロック図である。

【図6】本発明の第1の実施の形態による図5のインクデータ処理装置のインクデータ処理部（100）の機能ブロック図である。

【図7】本発明の第1の実施の形態による図6のインクデータ処理部のストロークオブジェクト操作部（122）の機能ブロック図である。

【図8】図7のストロークオブジェクト操作部内のポイント「A」から「D」で実行される処理を説明する図である。

【図9】図6のインクデータ処理部によって実行されるサンプルルーチンを説明するフローチャートである。

【図10】図5のインクデータ処理装置のポイント「E」に出力され得る、インターフェイス定義言語（IDL）で書かれたストロークファイルフォーマット（SFF）ファイルのサンプルを説明する図である。

【図11】ストロークファイルフォーマット（SFF）で書かれ、図5のインクデータ処理装置のポイント「E」に出力され得るストロークオブジェクトファイルのサンプルを説明する図である。

【図12】図5のインクデータ処理装置のポイント「F」に出力され得る、ストロークメッセージフォーマット（SMF）による3つのメッセージと、図5のインクデータ処理装置のポイント「G」に出力され得る1つのパケットとを説明する図である。

【図13A】図5のインクデータ処理装置のポイント「D」に出力され、グラフィック処理部（300）又はインクデータ整形部（140）に入力され得る、キャットマルローム曲線補間動作を受けたストロークオブジェクトを説明する図である。

【図13B】図5のインクデータ処理装置のポイント「H」にグラフィック処理部（300）から出力され得る、図13のストロークオブジェクトのレンダリング（表示）結果を説明する図である。

【図14】本発明の第1の実施の形態によるストロークオブジェクトに適用されるスライス操作のフローチャートである。

【図15A】図14のステップS1409で実行される、2つのストロークの間の単一の（中間の）交点（P__i n t e r s e c t __M i d）を決定する処理を説明する図である。

【図15B】図14のステップS1413で実行される、スライスしようとしている有幅のストロークオブジェクトと既存のストロークオブジェクトとの間の2つの（端）交点（P__i n t e r s e c t __L及びP__i n t e r s e c t __R）を得る処理を説明する図である。

【図16A】図14のステップS1415で導出される、ストロークオブジェクトをスラ

10

20

30

40

50

イスしてなる2つのスライスのうちの第1のものを説明する図である。

【図16B】図16Aに示した第1のスライスを定義するパラメータのデータ構造を説明する図である。

【図16C】新たに生成された第1のストロークオブジェクトの、表現されたパスを説明する図である。

【図17A】図14のステップS1415で導出される、ストロークオブジェクトをスライスしてなる2つのスライスのうちの第2のものを説明する図である。

【図17B】図17Aに示した第2のスライスを定義するパラメータのデータ構造を説明する図である。

【図17C】新たに生成された第2のストロークオブジェクトの、表現されたパスを説明する図である。

【図18A】図16Aに示した第1のスライスのために新たな終了点を得る処理と、図17Aに示した第2のスライスのために新たな開始点を得る処理とを説明する図である。

【図18B】本発明の第1の実施の形態による、ホールセグメントオブジェクトを定義するパラメータのデータ構造を説明する図である。

【図19】ストロークファイルフォーマット(SFF)で書かれ、1つのストロークオブジェクトをスライスしてなる2つのスライスを表す2つの新規生成ストロークオブジェクトを含むファイルのサンプルを説明する図である。

【図20】図14のステップS1422で実行されるインクデータ伝送処理の詳細なフローチャートである。

【図21A】スライス操作に関連してインクデータの伝送のために使用され得る色々な伝送メッセージタイプのうちのタイプAを説明する図である。

【図21B】スライス操作に関連してインクデータの伝送のために使用され得る色々な伝送メッセージタイプのうちのタイプBを説明する図である。

【図21C】スライス操作に関連してインクデータの伝送のために使用され得る色々な伝送メッセージタイプのうちのタイプCを説明する図である。

【図21D】スライス操作に関連してインクデータの伝送のために使用され得る色々な伝送メッセージタイプのうちのタイプDを説明する図である。

【図22】本発明の第1の実施の形態による、ネットワークを介してインクデータをリモート受信するように構成されたインクデータ受信装置の機能ブロック図である。

【図23】本発明の第1の実施の形態による受信側での、操作(スライス)オブジェクトの受信処理を説明するフローチャートである。

【図24A】従来技術に関連する技術的問題を説明する図である。

【図24B】従来技術に関連する技術的問題を説明する図である。

【図25】本発明の第2の実施の形態における使用に適したインクデータ構造の実体関連図である。

【図26】第2の実施の形態における使用に適した通信システム図の全体である。

【図27】図26に示した通信システムの伝送装置(10-1)を説明する図である。

【図28】本発明の第2の実施の形態における、共通描画領域(キャンバス)の更新状態を保管するのに適した記録フォーマットのサンプルを説明する図である。

【図29】図26に示した通信システムの中継サーバ(10-2)を説明する図である。

【図30A】伝送装置の通信及びグラフィック環境を連帯して描写又は定義する通信パラメータ、描画パラメータ、及びユーザポリシーパラメータのうち通信パラメータを説明する図である。

【図30B】伝送装置の通信及びグラフィック環境を連帯して描写又は定義する通信パラメータ、描画パラメータ、及びユーザポリシーパラメータのうち描画パラメータを説明する図である。

【図30C】伝送装置の通信及びグラフィック環境を連帯して描写又は定義する通信パラメータ、描画パラメータ、及びユーザポリシーパラメータのうちユーザポリシーパラメータを説明する図である。

10

20

30

40

50

【図 3 1】図 2 6 に示した通信システムの受信装置（1 0－3）を説明する図である。

【図 3 2】本発明の第 2 の実施の形態による、伝送装置（1 0－1）、中継サーバ（1 0－2）、及び受信装置（1 0－3）の間で行われるインクデータ通信を説明するシーケンス図である。

【図 3 3】インクデータを伝送するために規定の単位 T を発見する処理のサンプルを示すフローチャートである。

【図 3 4】本発明の第 2 の実施の形態による、複数の装置の間でインクデータを伝送する（通信する）のに適した通信パケット及びメッセージの伝送フォーマットのサンプルを説明する図である。

【図 3 5 A】データ再送スキームを含む通信プロトコルにおいて使用される通信パケットを説明する図である。

10

【図 3 5 B】データ再送メカニズムを含まない通信プロトコルにおいて使用される通信パケットを説明する図である。

【図 3 6 A】データ再送メカニズムを含まない通信プロトコルでの使用に適した、通し ID を用いるデータ再送処理のサンプルを示すシーケンス図である。

【図 3 6 B】データ再送メカニズムを含まない通信プロトコルでの使用に適し、かつ、データ再送が実行されないデータ伝送処理を示すシーケンス図である。

【図 3 6 C】メッセージの制御位置を計算する方法を説明する図である。

【図 3 6 D】メッセージの制御位置を計算する方法を説明する図である。

【図 3 6 E】メッセージの制御位置を計算する方法を説明する図である。

20

【図 3 6 F】図 3 6 B のシーケンスでの使用のための、図 3 6 C～図 3 6 E で計算された制御位置を使用する補間（エラー隠蔽）処理を説明する図である。

【図 3 6 G】図 3 6 B のシーケンスでの使用のための、図 3 6 C～図 3 6 E で計算された制御位置を使用する補間（エラー隠蔽）処理を説明する図である。

【図 3 7】伝送装置によって発行された共通描画領域の更新要求が中継サーバによって拒否される場合のインクデータ通信を説明するためのシーケンス図である。

【図 3 8】中継サーバが伝送装置からストロークオブジェクトの断片化データを受け取り、これを受信装置に対して中継するために、ストロークオブジェクトごとにすべての断片化データを結合するようにした、図 3 2 に示したシーケンス図の第 1 の変形例である。

【図 3 9 A】図 3 8 に示した第 1 の変形例での使用のためのフォーマットであって、その中に 1 つのストロークオブジェクトのためのすべてのインクデータが結合された状態で含まれるデータ伝送フォーマットである。

30

【図 3 9 B】ストローク ID が知られているときにストローク全体にかかるストロークオブジェクトデータを要求するメッセージである DATA__INK__ALL__REQ のデータ伝送フォーマットのサンプルを説明する図である。

【図 4 0】ストロークオブジェクトが「そのまま」（すなわち、断片化されずに）伝送装置から中継サーバを介して受信装置に伝送されるようにした、図 3 2 に示したシーケンス図の第 2 の変形例である。

【図 4 1】中継サーバが、伝送装置からストロークオブジェクトを受け取り、受け取ったストロークオブジェクトを、受信装置に対して中継するために複数の断片化データに断片化するようにした、図 3 2 に示したシーケンス図の第 3 の変形例である。

40

【図 4 2】共通描画領域の原点に関連してユーザ特有のストローク開始ポイントの概念を説明する図である。

【図 4 3】ストローク単位より大きい意味単位でのインクデータの伝送にかかる第 2 の実施の形態を説明する図である。

【図 4 4】ストローク単位より大きい意味単位でのインクデータの伝送にかかる第 2 の実施の形態を説明する図である。

【図 4 5】本発明の第 3 の実施の形態による、一方では、インクデータ処理部のデータ入出力であって生成する方法によるものを説明し、他方では、インクデータ処理部のデータ入出力であって再生する方法によるものを説明する図である。

50

【図４６Ａ】本発明の第３の実施の形態によるインクデータ生成方法の３つの構成例のうちの１番目を説明する図である。

【図４６Ｂ】本発明の第３の実施の形態によるインクデータ生成方法の３つの構成例のうちの２番目を説明する図である。

【図４６Ｃ】本発明の第３の実施の形態によるインクデータ生成方法の３つの構成例のうちの３番目を説明する図である。

【図４７Ａ】本発明の第３の実施の形態によるインクデータ再生方法の２つの構成例のうちの１番目を説明する図である。

【図４７Ｂ】本発明の第３の実施の形態によるインクデータ再生方法の２つの構成例のうちの２番目を説明する図である。

【図４８Ａ】本発明の第３の実施の形態によるインクデータモデル（ストローク言語（ＳＬ））に従う、インクデータ構造の実体関連図である。

【図４８Ｂ】図４８Ａに示したインクデータ構造の詳細な実体関連図である。

【図４８Ｃ】ストロークオブジェクトのグラフィック表現である。

【図４８Ｄ】キャンバスオブジェクトを説明する図である。

【図４８Ｅ】メタデータオブジェクトを説明する図である。

【図４８Ｆ】画面上で見られる、３つの異なる描画スタイルオブジェクトの描画結果を説明する図である。

【図４８Ｇ】操作（スライス）オブジェクトの動作を説明する図である。

【図４８Ｈ】操作（消去）オブジェクトの動作を説明する図である。

【図４８Ｉ】既存のストロークオブジェクトに適用される操作（選択及び変形）オブジェクトの動作を説明する図である。

【図４８Ｊ】本発明の第３の実施の形態によるストロークファイルフォーマット（ＳＦＦ）内に並べられたインクデータ構造の構文を説明する図である。

【図４８Ｋ】本発明の第３の実施の形態によるストロークファイルフォーマット（ＳＦＦ）内に並べられたインクデータ構造の構文を説明する図である。

【図４８Ｌ】本発明の第３の実施の形態によるストロークファイルフォーマット（ＳＦＦ）内に並べられたインクデータ構造の構文を説明する図である。

【図４９】本発明の第３の実施の形態によるインクデータ処理部の機能ブロック図である。

【図５０Ａ】本発明の第３の実施の形態による、図４９の「インクデータ生成部」内で実行され、ポイントオブジェクトの属性として半径及びアルファの情報を出力する処理を説明するフロー図である。

【図５０Ｂ】ペンイベントデータに関してコンテキスト情報を定義するために使用され得るアプリケーション又はオペレーティングシステムのＧＵＩのサンプルを説明する図である。

【図５１】本発明の第３の実施の形態による、図５０のステップＳ１２０５で速度を導出する処理を説明する図である。

【図５２】本発明の第３の実施の形態による、図５０のステップＳ１２０７で半径を導出する処理を説明するフロー図である。

【図５３】本発明の第３の実施の形態による、図５２のステップＳ１２０７__０１で使われるものとしてのストロークの「フェーズ」の定義を説明するフロー図である。

【図５４】本発明の第３の実施の形態による、図５２のステップＳ１２０７__０５及びステップＳ１２０７__０７で使われるものとしての、パラメータ（速度）から半径を導出するための３つの関数を説明するグラフである。

【図５５】本発明の第３の実施の形態による、図５０のステップＳ１２０９において透明度（又は不透明度）を示すアルファを導出する処理を説明するグラフである。

【図５６】本発明の第３の実施の形態による、図５５のステップＳ１２０９__０５及びステップＳ１２０９__０７で使われるものとしての、パラメータ（速度）からアルファ（透明度／不透明度）を導出するための２つの関数を説明するグラフである。

10

20

30

40

50

【図 5 7】本発明の第 3 の実施の形態による、X 及び Y 座標とともに半径及びアルファの値をインクデータフォーマット（データ構造）に整形する処理を説明するフロー図である。

【図 5 8】本発明の第 3 の実施の形態による、図 5 7 のステップ S 1 4 1 1 及びステップ S 1 4 1 3 の実施例を説明する図である。

【図 5 9】本発明の第 3 の実施の形態による、図 5 7 のステップ S 1 4 1 1 及びステップ S 1 4 1 3 で使われる整数データ型への浮動データ型の変換を説明する図である。

【図 6 0】本発明の第 3 の実施の形態による、図 5 9 のデータ型変換の結果としての圧縮の効率化を説明する図である。

【図 6 1】本発明の第 3 の実施の形態による、生成されたインクデータを圧縮するために図 4 9 に示した「インクデータ整形部」で実行され得る処理を説明するフロー図である。

【図 6 2】本発明の第 3 の実施の形態による、半径情報をインクデータ属性として出力するため、図 4 9 に示した「インクデータ生成部」において（図 5 2 に代えて）実行される処理を説明するフロー図である。

【図 6 3】本発明の第 3 の実施の形態による、インクデータ属性としてアルファ情報を出力するため、図 4 9 に示した「インクデータ生成部」において（図 5 5 に代えて）実行される処理を説明するフロー図である。

【図 6 4】本発明の第 3 の実施の形態による、インクデータ処理部と様々なアプリケーションとの間の関係を説明する図である。

【図 6 5】本発明の第 3 の実施の形態による、インクデータ内の半径及びアルファの情報を X 及び Y 座標とともに取り出す（再生する）ためのインクデータ再生処理と、取り出した情報及びデータを描画アプリケーションからの要求に応じて出力することとを説明するフロー図である。

【図 6 6】本発明の第 3 の実施の形態による、図 6 5 のステップ S 2 0 1 1 及びステップ S 2 0 1 3 の実施例を説明する図である。

【図 6 7】本発明の第 3 の実施の形態による、選択された描画スタイルオブジェクトを描画対象のストロークオブジェクトに適用する描画処理を説明するフロー図である。

【図 6 8】本発明の第 3 の実施の形態による、図 5 4 に示した減衰（ダンピング）関数に基づいて生成されたインクデータの入力の結果としての描画レンダリング例を説明する図である。

【図 6 9】本発明の第 3 の実施の形態による、図 5 6 に示した冪関数に基づいて生成されたインクデータの入力の結果としての描画レンダリング例を説明する図である。

【図 7 0】本発明の第 3 の実施の形態による、図 5 4 に示した減衰関数及び図 5 6 に示した冪関数の両方に基づいて生成されたインクデータの入力の結果としての描画レンダリング例を説明する図である。

【図 7 1】本発明の第 3 の実施の形態による、図 5 4 の他の関数（S 字及び周期関数）の効果を示す描画レンダリング例を説明する図である。

【図 7 2】本発明の第 3 の実施の形態による、描画対象のストロークの開始点及び終了点の半径として特別な値を用いることの効果を示す描画レンダリング例を説明する図である。

【図 7 3】本発明の第 4 の実施の形態による、インクデータが使用されるシステムの全体を説明する図である。

【図 7 4】本発明の第 4 の実施の形態によるインクデータ処理部の機能ブロック図である。

【図 7 5】図 7 4 に示した本発明の第 4 の実施の形態によるインクデータ処理部のより詳細な機能ブロック図である。

【図 7 6】図 7 5 に示したストロークオブジェクト操作部（1 2 2）の機能ブロック図である。

【図 7 7 A】ストロークオブジェクトを生成する方法を説明するフローチャートである。

【図 7 7 B】ストロークオブジェクトを生成する方法を説明するフローチャートである。

10

20

30

40

50

【図 7 8】図 7 5 に示したメタデータオブジェクト操作部（1 2 4）の機能ブロック図である。

【図 7 9】メタデータオブジェクトを生成する方法を説明するフローチャートである。

【図 8 0】図 7 5 に示したレンダリング（描画スタイル）オブジェクト操作部（1 2 6）の機能ブロック図である。

【図 8 1】（描画）スタイルオブジェクトを導出する方法及びそのカスケーディングプロパティを説明するフローチャートである。

【図 8 2】図 7 5 に示した操作オブジェクト操作部（1 2 8）の機能ブロック図である。

【図 8 3 A】スライスオブジェクトのような操作オブジェクトを導出する方法を説明するフローチャートである。

【図 8 3 B】スライスオブジェクトのような操作オブジェクトを導出する方法を説明するフローチャートである。

【図 8 4】図 7 5 に示したインクデータ整形部（1 4 0）の機能ブロック図である。

【図 8 5】図 8 4 に示したインクデータ整形部において実行される処理を説明するフローチャートである。

【図 8 6】ストロークファイルフォーマット（S F F）のデータを出力する方法を説明するフローチャートである。

【図 8 7】J P E G フォーマットのデータを出力する方法を説明するフローチャートである。

【図 8 8】ストロークメッセージングフォーマット（S M F）のデータを出力する方法を説明するフローチャートである。

【図 8 9】様々なファイルフォーマット及びメッセージングフォーマットで出力されたデータ（S F F / J P E G 及び S M F）の入力処理を説明する機能ブロック図である。

【図 9 0 A】S F F ファイル内に配置されたオブジェクトを解釈して再生するための処理のフローチャートである。

【図 9 0 B】I n k M L での入力に基づくオブジェクトを解釈して再生するための処理のフローチャートである。

【図 9 0 C】S M F による操作（スライス）オブジェクトを受信して実行する処理のフローチャートである。

【図 9 1】側面 1 に関して、図 7 5 のインクデータ処理装置（1 0 1）を使用することの効果の説明する図である。

【図 9 2】側面 2 に関して、図 7 5 のインクデータ処理装置（1 0 1）を使用することの効果の説明する図である。

【図 9 3】側面 3 に関して、図 7 5 のインクデータ処理装置（1 0 1）を使用することの効果の説明する図である。

【発明を実施するための形態】

【0 0 2 8】

詳細な説明

本明細書で使用される以下の用語は、特定の実施の形態にのみ適用され得ると特定の文脈で特に定義されない限り、本明細書に記述された各種の実施の形態を通じて、次の意味を有する。

【0 0 2 9】

「ペンイベントデータ」（入力 1）は、ユーザの手書き描画動作に基づいて入力されるデータを意味する。ペンイベントデータは、所与の入力装置によって入力される生データであってもよいし、該生データの処理によって得られるデータであってもよい。すべてのペンイベントデータは、ユーザによって描かれた各ストロークの位置情報（例えば X Y 座標）を少なくとも有することが期待される一方、装置に依存しており、入力装置の各タイプを特定する属性（例えば、筆圧データ、ペン回転又は傾斜角のデータ）を含む。例えば、筆圧を検出可能な入力装置から受信されるペンイベントデータは、筆圧を検出できない入力装置から受信されるペンイベントデータとは異なるものとなる。

10

20

30

40

50

【0030】

「インクデータ」(200)は、ペンイベントデータから導出されるオブジェクトの集合を意味する。インクデータ200は、ペンイベントデータに基づいて形成されるパス(ストローク)をキャプチャするものであって、各パスの性質(色、ペンタイプなど)を記述する中間データの種類であるベクトルデータの形式をとる。インクデータ200は装置に依存するものではなく、筆圧及び／又はペン回転／傾斜角といった属性に対応する装置、並びに、これらの属性に対応しない装置のいずれによっても利用され得る。本発明の実施の形態によるインクデータ200は、ストロークオブジェクト210、メタデータオブジェクト250、描画スタイルオブジェクト230、及び操作オブジェクト270を含む。インクデータ200については、図2、図3A～図4B、図25、図48A～48I等を参照しつつ、以下で詳細に説明されるであろう。

【0031】

「ストロークオブジェクト」(210)は、インクデータ200内に含まれるオブジェクト又はデータの種類である。上掲した非特許文献1～4に記述される「ストローク」「パス」「トレース」「カンバスパス」はすべて、ストロークオブジェクト210である。ストロークオブジェクト210は、入力装置におけるユーザ操作によって取得されるパス(ストローク)の形を表している。

【0032】

「メタデータオブジェクト」(250)は、インクデータ200内に含まれるオブジェクトの種類であり、著者、ペンID、ローカルに取得される日時情報、GPSによって取得される位置情報などのような、ストロークオブジェクト210を説明する非描画関連情報を含む。

【0033】

「描画スタイルオブジェクト」(230)は、インクデータ200に含まれるオブジェクトの種類であり、ストロークオブジェクト210が画面上にレンダリングされる(描かれる、表現される、ラスター化される)際に、その形(ストローク幅、ストロークのスタイル／パターン)や色を制御するために必要となる情報を含む。一口で言えば、描画スタイルオブジェクトは、ストロークオブジェクト210のレンダリング(描画)を制御する。

【0034】

「操作オブジェクト」(270)は、インクデータ200に含まれるオブジェクトの種類であり、既存の1以上のストロークオブジェクト210それぞれの全部又は一部に対し、操作／変更の動作(例えば、スライス操作)を実行する。ストロークオブジェクトの一部に対する操作オブジェクト270のアプリケーションについては、第1の実施の形態にて後に詳細に記述されるであろう。

【0035】

「ストローク言語(SL)」は、インクデータ200を構成する様々なオブジェクトの属性及び意味を定義する情報モデルである。

【0036】

「ストロークファイルフォーマット(SFF)」は記録フォーマットの種類であり、その中で出力対象のインクデータ200がシリアル化される。SFFの詳細は、図10、図11、図19、図28、図48J、図48K、図48L、図57、図65、及び図86を参照して、後に記述されるであろう。

【0037】

「記録フォーマット」は、SFFフォーマットやSVGフォーマットのような、インクデータ200を永続的なものとするために適したフォーマットを意味する。記録フォーマットによるインクデータ200は、ファイル又はデータベースとして記憶装置(HDD、ネットワークストレージなど)内に記録される。そしてそこから、そのシリアル化されたデータストリームが引き出され、かつ、デシリアル化され得る。

【0038】

10

20

30

40

50

「ストロークメッセージフォーマット（SMF）」は、伝送パケット又はフレームに含まれるメッセージ伝送フォーマットの一種であり、定義された伝送プロトコルを用いてインクデータ200を伝送する際に使用するためのものである。SMFの詳細は、図12、図21、図34、図35A、図35B、図39A、図39B、及び図88を参照して、後に記述されるであろう。

【0039】

「伝送フォーマット」は、SMFフォーマットのような、ネットワークを介してインクデータ200を伝送する（メッセージを送る）のに適したメッセージフォーマットを意味する。

【0040】

「インクデータ」は、GIFやJPEG画像のようなラスター化された画像を意味し、ピクセルデータを含み、インクデータ200に基づいて生成（描画）され得る。中間物でない画像フォーマットデータをインクデータ200に戻すことは、不可能である。

【0041】

以下の用語は、例えば図5に示すように、インクデータ200を処理するために使用される主な構造及び構成要素のそれぞれを記述するために使用される。

【0042】

「インクデータ処理部」（100）は、インクデータ200を生成し、格納し、処理するプロセッサを意味する。本明細書では、ペンイベントデータに基づいてインクデータ200を生成するため、かつ、定義されたフォーマットにインクデータ200を配列するために使用されるインクデータ処理部を、参照符号100Tによって表す。一方、定義されたフォーマットで生成され配列されたインクデータ200を再生産するためにコンピュータ内で使用されるインクデータ処理部を、参照符号100Rによって表す。インクデータ処理部100の詳細は、図5及び図75を参照し、さらに図6、図22、図27、図31、図49、及び図74も参照して、後に記述されるであろう。インクデータ処理部100は一般に、次の3つの情報を受け付け／含み／受信する。1つ目はペンイベント（入力型）情報（入力1）であり、2つ目はコンテキスト情報（入力2）であり、3つ目は操作情報（入力3）である。

【0043】

「インクデータ生成部」（120）は、インクデータ200を生成し、インクデータ200を展開する。本明細書では、入力センサから受信される入力信号に基づいてインクデータ200を生成するインクデータ生成部を参照符号120Tによって表し、既に生成されたインクデータ200を展開し、メモリ内にリストアするインクデータ生成部を参照符号120Rによって表す。

【0044】

「インクデータ整形部」（140）は、入力及び出力の目的で記録フォーマット又は輸送フォーマット内に配列されたインクデータ200を処理する。本明細書では、定義されたフォーマットでインクデータ200を出力するインクデータ整形部を参照符号140Tによって表し、定義されたフォーマットのインクデータ200の入力を受け付けるインクデータ整形部を参照符号140Rによって表す。

【0045】

第1の実施の形態

本発明の第1の実施の形態は、インクデータ200を構成するストロークオブジェクト210を生成し、レンダリングし、操作し（例えば、スライスし）、及び通信することに向けられたものである。特に、ストロークオブジェクト210の一部分の操作が、図1を参照して上述したように、複数のプロセッサの間での操作オペレーションの共有（伝送）とともに記述されるであろう。

【0046】

第1の実施の形態の背景

非特許文献1～4に記述されるストロークオブジェクトは、所定の補間曲線アルゴリズム

10

20

30

40

50

ムを用いて補間曲線又はパスを生成するために必要な複数の点又は複数の制御点を含む。

【0047】

非特許文献1, 2は、特別な補間曲線アルゴリズムを何も明示していない。言い換えれば、任意の適当な補間曲線アルゴリズムが使用され得る。

【0048】

非特許文献3, 4は、ポリベジエ（キュービックベジエ）曲線を用いている。ポリベジエ曲線では、1つの曲線セグメント（パスセグメント）の開始点 P_i と終了点 P_{i+1} が制御点として利用される。加えて、点 P_i と点 P_{i+1} の間の曲線セグメントの曲率を定めるために、少なくとももう1つの制御点が必要とされる。その制御点は、 P_i 及び P_{i+1} のいずれとも異なり、かつ、曲線セグメントを含む曲線上にない（すなわち、曲線の外にある）ものとなる。例えば、キュービックベジエ曲線は、曲線セグメントを定めるために、曲線の外に位置する2つの制御点を必要とする。

【0049】

例えば、キュービックベジエ曲線のために使用されるXML表記`<stroke-width="5" d="M 100, 200 C100, 100 300,100 300,200"/>`は、開始点（100, 200）が制御点として使用され、終了点（300, 200）が他の制御点として使用され、さらに、（開始点と終了点の間であって曲線セグメントの外にある）2つの制御点（100, 100）及び（300, 100）が曲線セグメントを定めるために使用される、ということ意味する。

【0050】

近年、W3CでSVG（非特許文献3）を担当しているSVGワーキンググループでは、補間曲線についてキャットマルローム曲線を利用することの可能性が議論されている。ポリベジエ曲線とは異なり、キャットマルローム曲線は、曲線の外にある（すなわち、曲線上にない）制御点を有しない。キャットマルローム曲線は、開始点（ P_i ）、終了点（ P_{i+1} ）、開始点の「前」の点（ P_{i-1} ）、終了点の「後ろ」の点（ P_{i+2} ）という4つの制御点により各曲線セグメントを定義する。すべての制御点は、曲線上にある。別の言葉で言えば、キャットマルローム曲線は、そのすべての制御点を通る（もっとも、各曲線セグメントが曲線セグメントの「前」と「後ろ」にある2つの制御点を必要とすることから、ストロークオブジェクトの一番端にある2つの点では曲線セグメントが定義されない）。

【0051】

第1の実施の形態の概要

図24A及び図24Bは、非特許文献1～5に記載された従来技術におけるインクデータの定義が遭遇する技術的問題を説明している。図24Aは、ストロークオブジェクトによって表現され、スライス操作2403が適用される曲線2401Sを示している。曲線2401Sを表すストロークオブジェクトは、入力センサを介して入力される一組の点座標（ $p_1 \sim p_{10}$ ）を含む。

【0052】

図24Aでは、ストロークオブジェクトの曲線セグメントをスライスするために、交点2405を通過する分割線に沿って、点座標 p_5 と p_6 の間にスライス操作2403が適用されている。図24Bは、スライス操作2403の結果として得られる2つの分割曲線2411S1, 2415S2を示している。曲線2411S1は点座標 $p_1 \sim p_5$ を含み、曲線2415S2は点座標 $p_6 \sim p_{10}$ を含む。示されるように、実線で示される分割曲線2411S1は点座標 p_5 で終わっており、したがって、交点2405まで延在する実際上の分割曲線よりも短いものとなっている。同様に、実線で示される分割曲線2415S2は点座標 p_6 で始まっており、交点2405で始まる実際上の分割曲線よりも短いものとなっている。図24Bでは、破線で示される不完全な曲線セグメント2413が、スライス操作2403によって失われた曲線の部分を示している。

【0053】

交点2405に位置する新たな制御点を加え、さらに、 p_5 と交点2405の間、及び

、交点 2 4 0 5 と p 6 の間のそれぞれに、新たに生成される部分的な曲線セグメント 2 4 1 3 を定義するための制御点を加えることは可能ではある。しかし、部分的な曲線セグメント 2 4 1 3 の終了点 2 4 0 5 までの正確な形状を表すための新たな制御点の位置を計算することは計算集約的であり、簡単なタスクではない。例えば、キュービックベジエ曲線のような補間曲線を用いるとすると、新たなセグメントを定義するためには、曲線（又はパス）の外にある 2 つの制御点を計算する必要がある。キャットマルローム曲線を用いる場合には、新たなセグメントを定義するために曲線を通る 2 つの制御点を計算する必要があり、その結果、実際の曲率を維持するために、それまでのすべての制御点を芋蔓式に再計算しなければならない。いずれのタイプの計算も高度に複雑であり、時間を要するため、グラフィック又は描画アプリケーションにおいてリアルタイムなスライス操作をサポートすることが不可能になる。

10

【0054】

ユーザが、スライス操作の結果として得られる 2 つのセグメントのそれぞれが元のストロークオブジェクトから切り出される実際の分割曲線を表すように、インクデータを形成するストロークオブジェクトをスライスすることを可能にする方法及びシステムに対するニーズが存在する。そして、そのような方法及びシステムは、補間曲線のために使用される新たな制御点の位置の計算を必要としないものであることが好ましい。そのような計算は複雑であり、しばしば計算集約的であるため、リアルタイムなアプリケーションをサポートすることができなくなるからである。

【0055】

20

一側面によれば、本発明は、ストロークオブジェクト 2 1 0 を含むインクデータを生成し、描画し、操作し（例えばスライスし）、伝達する方法及びシステムを提供する。この方法及びシステムにおいて、ストロークオブジェクト 2 1 0 は、レンダリング対象となるストロークオブジェクト 2 1 0 の特定の部分を定義する範囲情報を含み、又は、該範囲情報と関連付けられる。範囲情報が全部表示を示す場合にはストロークオブジェクト 2 1 0 の全体が表示され、範囲情報が部分表示を示す場合には、ストロークオブジェクト 2 1 0 の一方又は両方の端部が表示されないものとして部分的に指定される。スライス操作が元のストロークオブジェクト 2 1 0 に適用され、2 つの新しいストロークオブジェクト 2 1 0 が生成されるとき、1 つ目の新たなストロークオブジェクト 2 1 0 は、該 1 つ目の新たなストロークの終わりをラスタ化する（若しくは、レンダリングする、又は、結果として表示する）点となる新たな「終了」点を指定する範囲情報と関連付けられる。同様に、2 つ目の新たなストロークオブジェクト 2 1 0 は、該 2 つ目の新たなストロークの表示が始まる点となる新たな「開始」点を指定する範囲情報と関連付けられる。1 つ目及び 2 つ目の新たなストロークオブジェクト 2 1 0 はいずれも元のストロークオブジェクトと同じ構造及び制御点（たとえ部分的であっても）を維持し、それ故、1 つ目及び 2 つ目の新たなストロークオブジェクト 2 1 0 の表示は、正確に元のストロークオブジェクト 2 1 0 の形を辿ることとなる。また、新たな制御点の計算は不要となる。

30

【0056】

他の側面によれば、インクデータ 2 0 0 を形成するためにストロークオブジェクト 2 1 0 を出力する方法及びシステムが提供される。ストロークオブジェクトは、複数の座標位置を表す複数のポイントオブジェクトを含む。この複数のポイントオブジェクトのうちの少なくともいくつかは、補間曲線セグメントを生成するために使用される制御点として機能し、全体としてストロークオブジェクト 2 1 0 のパスを形成する。ストロークオブジェクト 2 1 0 は、ストロークオブジェクト 2 1 0 の表示が始まる開始曲線セグメント内の開始点と、ストロークオブジェクト 2 1 0 の表示が終わる終了曲線セグメント内の終了点とを定義する範囲情報をさらに含む。元のストロークオブジェクト 2 1 0 が 2 つの新しいストロークオブジェクト 2 1 0 を生むためにスライスされるとき、2 つの新たなストロークオブジェクト 2 1 0 はそれぞれ、自身の範囲情報、すなわち自身の開始点と自身の終了点とを示すパラメータの他に、元のストロークオブジェクト 2 1 0 のものと全く同じポイントオブジェクトの部分的なセットを含む。

40

50

【0057】

他の側面によれば、上記のようにして組み立てられたインクデータを描画する（レンダリングする、又は、表示する）方法及びシステムが提供される。この方法及びシステムにおいて、各ストロークオブジェクト210は、複数のポイントオブジェクトと、範囲情報とを含む。複数のポイントオブジェクトのうちの少なくともいくつかは、補間曲線セグメントを生成するために使用される制御点である。この方法及びシステムは、制御点に基づいて曲線セグメントを補間することでストロークオブジェクト210のパスを生成し、そして、範囲情報によって指定されたストロークオブジェクト210の一部を表示することによって、画面上に各ストロークオブジェクト210を描く。他の言葉で言えば、この方法及びシステムは、範囲情報内に指定される開始点でストロークオブジェクト210の表示を開始し、範囲情報内に指定される終了点でストロークオブジェクト210の表示を停止する。

10

【0058】

さらなる側面によれば、上記のようにして組み立てられたインクデータのストロークオブジェクト210をユーザがスライスすることを可能にする方法及びシステムが提供される。ユーザがストロークオブジェクト210上でスライス操作を実行するとき、この方法及びシステムは、スライシングパスとストロークオブジェクト210の交点の位置を算出する。（図24Aの2405を参照）。この方法及びシステムは、スライス操作の結果として、第1のストロークオブジェクト210及び第2のストロークオブジェクト210という2つの新しいストロークオブジェクト210を生成する。第1のストロークオブジェクト210は、ポイントオブジェクトの第1のセットと、表示開始点と表示終了点とを示す第1の範囲情報とを含む。この場合において、表示終了点は、算出された交点から導出される。第2のストロークオブジェクト210は、ポイントオブジェクトの第2のセットと、表示開始点と表示終了点とを示す第2の範囲情報とを含む。この場合において、表示開始点は、算出された交点から導出される。典型的には、第1のストロークオブジェクト210の第1の範囲情報は、元のストロークオブジェクト210のものと同一表示開始点を維持し、第2のストロークオブジェクト210の第2の範囲情報は、元のストロークオブジェクト210のものと同一表示終了点を維持する。

20

【0059】

上記のようにして組み立てられたインクデータは、共通の描画領域（共通の「キャンバス」）上でストロークを描画し、操作する（スライスする）という経験を複数のユーザがリアルタイムで共有できるよう、インクデータを処理する能力を有する色々な装置又はアプリケーションの間で容易に伝達され得る。

30

【0060】

本発明の様々な方法及びシステムによれば、上記のようにして組み立てられたインクデータは、生成され、出力され、画面上に表示され、ユーザがストロークオブジェクト210をスライスすることを可能にするために使用され、異なるユーザ間で共有される。ストロークオブジェクト210に含まれる実際の曲線セグメントの一部のみを表示するために範囲情報を使用することは、元のストロークオブジェクト120の形状をその末端まで正確に辿るスライスされた（新たに生成された）ストロークオブジェクト210を表示することを可能にする。また、スライスされたストロークオブジェクト210が元のストロークオブジェクト210と同じ構造及び制御点（たとえ部分的であっても）を維持することから、スライス操作に関連して新たな制御点を計算ないし再計算する必要がなくなる。

40

【0061】

本発明の方法及びシステムは、インクデータであって、その内部で曲線セグメントが例えば公知のポリベジエ曲線（キュービックベジエ、二次ベジエ）アルゴリズムやキャットマルローム曲線アルゴリズムのような様々なタイプの曲線補間アルゴリズムによって補間されるインクデータに適用され得る。

【0062】

第1の実施の形態の記述

50

図1は、本発明の実施の形態による、インクデータ200が利用されるシステムの全体を説明する図である。図1では、破線で輪郭が描かれたクラウド部1がインターネットのようなインフラを表しており、本発明のインクデータ200を利用するシステムは、このクラウド部1上で動作することができる。例示のインフラとしてのインターネットは、インターネットプロトコルスイート（例えば、IP、TCP、HTTP）と、ライブラリと、様々なウェブ及びメールデータフォーマット（HTML、MIME）を実行するソフトウェアと、それらの通信方法（HTTP、SMTP）との標準化されたセットを基にするもので、各ベンダーが独自に構成するハードウェア構成及びオペレーティングシステムの間の違いを吸収する。図1では、インフラ部1を通過する破線が、これらのインフラ技術に基づいて発生するデータ交換を説明している。図1では、実線で輪郭が描かれたクラウド部10がインクデータ200を交換するためのインフラを表しており、クラウド部10は、インクデータ200に関する共通の情報モデル（言語）を確立することによって実現される。インクデータ200は、様々なアプリケーションサービス（又はエコシステム）及び様々な装置によって共通に使用可能となるように一般化される。例えば、図1に示したアプリケーションサービス#1及びアプリケーションサービス#2はともに、インクデータ交換インフラ10を介して、インクデータ200を利用し、交換することができる。これは、いくつかの種類のコンピュータ、例えばモバイル端末及びサーバ、によって分散的にサポートされるインクデータ処理部100のために必要なライブラリとして、実現され得る。データ交換インフラ10を通過する実線は、インクデータ200を利用するための一群のライブラリを利用するいくつかのアプリケーションのために提供される様々なアプリケーションの間におけるインクデータ200の交換を説明する。インクデータ200の領域（ドメイン）において共通の情報モデルを確立することにより、様々なタイプのアプリケーション及び装置がインクデータ200を共有ないし交換できるようになる。

【0063】

図1では、装置10-1-1は、入力センサとして、筆圧データを出力できるペンタブレット型の入力装置を含み、第1のプロバイダ/ソフトウェアベンダーによって提供されるアプリケーションサービス#1のために提供されるアプリケーション#1を用いてインクデータを生成する。生成されたインクデータ200はその後、宛先であるメディア（例えば、ネットワーク）に対応する適切な出力フォーム（例えば、パケット内のSMF）で出力され得る。

【0064】

装置10-1-2は、ユーザの指によってなされる手書き入力を受け取る能力を有するタブレット型の入力装置である。装置10-1-2のセンサは、筆圧データを出力する能力を有さず、アプリケーションサービス#2のために提供されるアプリケーション#2を用いて筆圧情報を利用せず、又は、宛先であるメディアに対応する適切な出力フォームでインクデータ200を生成する。

【0065】

装置10-3は、アプリケーションサービス#2を使用する、さらに他のタイプのコンピュータ（例えば、デスクトップ型のPC）である。装置10-3は、装置10-1-1及び装置10-1-2のそれぞれから提供されるインクデータ200を結合する（統合すること）ことができる。装置10-3は、装置10-1-1及び装置10-1-2から出力されたインクデータ200を、互いに重ねた状態で、その画面上にレンダリングする（描画すること）ことができる。

【0066】

図2は、インクデータモデルの実態関連図である。本発明の実施の形態によるインクデータ200は、ストロークオブジェクトセット202と、スクリーン又はディスプレイ上にレンダリングする（描画する、表現する、ラスター化する）際にストロークオブジェクト210の形及び色を制御するために必要となる情報を含む描画スタイルオブジェクト（セット）230と、ストロークオブジェクト210を説明する非描画関連情報を含むメタデータオブジェクト250（例えば、著者）と、既存のストロークオブジェクト210を

操作する（例えば、スライスする、回転する）ために必要となる情報を含む操作オブジェクト（セット）270とを含む。

【0067】

ストロークオブジェクトセット202内のストロークオブジェクト210は、ポインタ（指、ペンなど）の動きによって形成されるストローク210（又はトレース、パス）を再生するために必要な情報を含む。ストロークは、複数の（「N」個の）ポイントオブジェクト212（ポイント1～ポイントN）を含む（217）。他の言葉で言えば、ストロークは、ポインタの動きによって生成されるペンイベントデータ（ポインタ操作）をサンプリングすることから得られる複数のポイントオブジェクトの座標によってサポートされる。ポイントオブジェクトは、それが2次元、3次元、・・・N次元空間におけるポイントオブジェクトの位置を示す限り、絶対的又は相対的な座標値の形式、又は、ベクトル形式などの任意の形式をとり得る。様々な実施の形態において、複数のポイントオブジェクトは、それらの間に曲線セグメント210を挿入し、それによってストロークオブジェクト210のパスを形成するために使用され得る制御点として機能する。

【0068】

本発明の実施の形態によれば、ストロークオブジェクト210はさらに、表示の対象となるストロークオブジェクト210の部分を実際範囲情報を含む。図示した実施の形態では、範囲情報は、ストロークオブジェクト210の開始曲線セグメント内の開始点を定義する第1のパラメータである「開始パラメータ」301と、ストロークオブジェクト210の終了曲線セグメント内の終了点を定義する第2のパラメータである「終了パラメータ」303とを含む。範囲情報は、ポイントオブジェクトが生成された後、ストロークオブジェクト210のために生成される。例えば、ストロークオブジェクト210上でスライス操作のような操作オペレーションが実行され、2つの新たなストロークオブジェクト210が生成されるとき、2つの新たなストロークオブジェクト210のそれぞれを構成する二組のポイントオブジェクトが得られ、2つの新たなストロークオブジェクト210のそれぞれに範囲情報が付加される。

【0069】

ここで、開始曲線セグメント及び終了曲線セグメントは、描画（表示）動作が開始する又は終了するセグメントを意味する。したがって、全く表示されないと指定されたストロークオブジェクト210の真に1つ目の曲線セグメントは、ここで使用される「開始」曲線セグメントではない。同様に、全く表示されないと指定された真に最後の曲線セグメントは、「終了」曲線セグメントではない。

【0070】

一般的に、複数のポイントオブジェクトの（x，y）座標を生成するためには2つの方法がある。第1に、ペンイベントデータが入力されている間、ペンイベントデータ（ペン操作）から導出される座標点が「生値型」のポイントとして出力され得る。第2に、完全なストロークを形成するためのすべてのポイントが入力された後、ストロークの近似曲線を示すベジエ曲線のような三次スプライン関数又はより高次の関数（例えば、ラグランジェ多項式）が生成され、その近似曲線を表現するために必要な最小数のポイントオブジェクトが「最適化ポイント型」として取得され得る。以下の説明では、ポイントオブジェクトは、第1の方法による「生値型」のものとして生成される。ただし、本発明はまた、第2の方法による「最適化ポイント型」のポイントオブジェクトを使用することも可能である。

【0071】

描画スタイルオブジェクト（セット）230は、ストロークオブジェクト210の形（ストローク幅、ストロークのスタイル／パターン）及び色を制御するために必要な情報を含む。要するに、描画スタイルオブジェクト230は、ストロークオブジェクト210のレンダリングを制御する。図示した例の描画スタイルオブジェクト230は、シェイプフィル（塗りつぶし）オブジェクト232及び粒子散布オブジェクト234を含む。

【0072】

10

20

30

40

50

図 3 B 及び図 3 C はそれぞれ、図 3 A と同じストロークオブジェクト 2 1 0 を 2 つの異なる描画スタイルオブジェクトによってレンダリングしてなる（描画してなる）2 つの結果を示す。

【0073】

図 3 B は、シェイプフィルオブジェクト 2 3 2 のレンダリング（描画）結果を示すもので、様々な半径及び幅を有する円 3 2 1 の集合としてストロークオブジェクト 2 1 0 を表現している。各円の中心は、ストロークオブジェクト 2 1 0 によって表されるトレースに沿って一列に整列しており、円の集合の外周は、包絡線 3 2 3, 3 2 5 を生成する（算出する）ために使用される。包絡線 3 2 3, 3 2 5 はその後、スクリーン又はディスプレイ上に図 3 A のストロークオブジェクト 2 1 0 を描画するために使用される。

10

【0074】

図 3 C は、粒子散布オブジェクト 2 3 4 のレンダリング（描画）結果を示すもので、ポイントスプライトの集合として図 3 A のストロークオブジェクト 2 1 0 を描画している。ポイントスプライトは形作られた粒子 3 4 1（薄片）であり、中心を持ち、サイズ、及び、薄片の規定の軸に対する回転角 3 4 5 θ が様々に変化する。様々なサイズの薄片がそれぞれの規定の軸に対して θ だけ回転し、その中心は、トレースからトレース方向に対して垂直な方向にオフセット 3 4 3 の分だけ移動する。オフセット 3 4 3 は、所定の種から導出される乱数値である。

【0075】

メタデータオブジェクト 2 5 0（図 2 参照）は、著者、ペン I D、ローカルに取得される日時情報、G P S によって取得される位置情報などのような、ストロークオブジェクト 2 1 0 を説明する非描画関連情報を含む。

20

【0076】

操作オブジェクト（セット）2 7 0 は、既存のストロークオブジェクト 2 1 0 を全体的又は部分的に操作する（例えば、選択する、変形する／回転する、スライスする、など）ために必要な情報を含む。そのような情報は、それぞれがストロークオブジェクト 2 1 0 に対して所望の操作をもたらすためにストロークオブジェクト 2 1 0 の全体又は一部に対して実行可能である複数の操作オブジェクトの形式で整理される。各操作オブジェクト 2 7 0 は、特定の操作オペレーションを定義し制御するパラメータを含む。例えば、選択オブジェクト 2 7 2 は、図 4 A に示すようにストロークオブジェクト 2 1 0 を選択し、変形する（例えば、変換行列によって回転する）ために使用されるパラメータを含む。スライスオブジェクト 2 7 4 は、図 4 B に示すようにストロークオブジェクト 2 1 0 をスライスするために使用されるパラメータを含む。

30

【0077】

図 4 A は、選択オブジェクト 2 7 2 の動作を説明する。選択及び変形のターゲットは既存のストロークオブジェクト 2 1 0「ストローク i」であり、図 4 A では、これが他のストローク j（ $j > i$ ）によって選択される。ストローク j は、新規かつ連続的に入力されるペンイベントデータに基づいて新規に入力されたもので、ポイントオブジェクト P 1 ~ P n を含む。ストローク j は、既存のストローク i を囲む領域を定義し、それによって既存のストローク i を選択するように入力される。選択オブジェクト 2 7 2 は、図 4 A の矢印 4 0 5 に図示するように、選択されたストローク i を変形する（回転する）ため、規定の変換行列を用いることができる。ストローク i がストローク j によって選択されるか否か、また、どのように選択されるか、を決定する方法には様々なものがある。例えば、もしストローク j が p 1 と p 2 の間の単一の位置（P _ i n t e r s e c t _ M i d）でストローク i と交差するのであれば、ストローク i の右側部分のみを選択し、4 0 5 によって変形することができる。ストローク i の残りの左側部分は選択されず、変形 4 0 5 によって変形されることなく維持される。これは、ストローク j を用いてストローク i に対し同時にスライス操作を適用することによって実現され得る（すなわち、ストローク j が選択オブジェクト 2 7 2 とストローク i 4 0 1 のためのスライスオブジェクト 2 7 4 との両方の生成を起動するために使用される）。このケースでは、ストローク i は 2 つの新たに

40

50

生成されたストロークに分割される。これら新たに生成されたストロークのうちの一方は、ストローク j によって完全に囲まれており、したがって選択される。

【0078】

図4Bは、スライスオブジェクト274の動作を説明する。スライスオブジェクト274は、ストロークi401の部分的な操作であり、ポイントオブジェクトP1～P4を含む新たなストロークオブジェクト403（ストロークj）によって生成される。ストロークオブジェクト403は、通常のストロークオブジェクト210ではないが、定義された操作オペレーションを既存のストロークオブジェクト210上で行うように構成された操作オブジェクトであることを示す型情報に関連付けられる。例えば、ストロークオブジェクト403（ストロークj）は、図5を参照して以下でより完全に記述されるように、「入力3」（操作オブジェクト）型であるとして分類されることができる。図4Bに図示するように、（「入力3」型のストロークj内に表現される）スライスオブジェクト274は、既存のストロークオブジェクト401（ストロークi）をスライスするように入力される。これを実現するため、スライスオブジェクト274は、既存のストロークオブジェクト401（ストロークi）を2つのスライス、すなわちスライスi1407及びスライスi2409、にスライスするために必要なパラメータを含む。スライスオブジェクト274は、様々なアプリケーションにおいて、スライサー、イレーザ、部分抽出器などとして機能するものであり得る。スライス操作が実行され、2つの新しいスライス407、409が生成された後、これらのスライスは、2つの完全に定義されたストロークオブジェクト210となるべく「コミット」（又はファイナライズ）され得る。この点において、元のストロークオブジェクト401（ストロークi）だけでなく、（コミットされていない）スライス407、409及びスライスを生成するために使用されたスライスオブジェクト274（ストロークj）自身についても、維持される必要はない。

【0079】

図5は、本発明の実施の形態による、インクデータを出力し、操作し、描画し、及び伝達する（送信する／受信する）ことのできるインクデータ処理装置の機能ブロック図である。この装置は一般的に、図1の装置10-1-1（センサタイプ1）又は装置10-1-2（センサタイプ2）に対応する。

【0080】

この例の装置は、入力センサ110と、入力処理部111と、インクデータ処理部100と、アプリケーション部300-1と、グラフィック処理部300と、ディスプレイ113と、通信部112（「Tx, Rx」）とを含み、これらはすべて、記憶装置と結合してCPUにより実行されるオペレーティングシステム400-1によって制御される。この装置は、ペンタブレットセンサの形態をとる入力センサ110を含むか又はこれと結合するパーソナルコンピュータ（PC）又はモバイル端末装置であってよい。

【0081】

入力センサ110は、ユーザの手書き動作を（ペンや指などのポイントを経して）検出し、検出した手書き動作を表現する入力データ信号を生成する。例えば、静電センサ、感圧センサ、電磁共鳴（EMR）ベースのセンサが使われ得る。

【0082】

入力処理部111は、各入力センサに依存するタイプの入力データを入力センサ110から受け取り、この入力データを、インクデータ200を生成するためのさらなる処理を受けることのできる「ペニイベントデータ」に変換する。生成された「ペニイベントデータ」は、インクデータ処理部100に「入力1」（図5のポイント「A」を参照）として入力される。ペニイベントデータ（「入力1」）は、感知した座標位置を少なくとも含み、入力センサ110が圧力／傾きの検出能力を有するか否かに応じて追加で、筆圧データやペンの傾きデータなどを含み得る。したがって、入力処理部111から出力されるペニイベントデータもまた、装置／センサに依存する。入力処理部111は、典型的には、アンドロイド（登録商標）オペレーションシステム上で動作する入力サブシステムのような入力センサ110のドライバであるソフトウェアプログラムとして実現される。入力セン

サ 1 1 0 と入力処理部 1 1 1 の構成は、図示したものに限定されない。例えば、入力センサ 1 1 0 及び入力処理部 1 1 1 のいくつか又は全部は、ペン形装置のようなデジタル文具装置として提供され得る。

【0083】

インクデータ処理部 1 0 0 は、インクデータ生成部 1 2 0 及びインクデータ整形部 1 4 0 を含む。インクデータ処理部 1 0 0（より具体的にはインクデータ生成部 1 2 0）は、入力処理部 1 1 1 から受信されるペンイベントデータと、アプリケーション部 3 0 0-1 から受信されるコンテキスト情報（「入力 2」）及び操作情報（「入力 3」）とに基づいてインクデータ 2 0 0 を生成する責任を負う。インクデータ処理部 1 0 0 は、典型的には、アプリケーション部 3 0 0-1 に動的及び／又は静的にリンクされた一組のライブラリとして実現される。

10

【0084】

コンテキスト情報（「入力 2」）は、ペンイベントデータ（「入力 1」）のコンテキスト又は環境を記述する情報であり、例えば、使用されるペン先タイプ（例えば、ブラシ、クレヨン、鉛筆）や、使用されるペン色（赤、緑、青）などを示すことができる。コンテキスト情報は、典型的には、入力センサ 1 1 0 への入力データの入力に先立ち、アプリケーション部 3 0 0-1 によって選択される。

【0085】

操作情報（「入力 3」）は、入力センサ 1 1 0 からの次の入力が典型的なペンイベントデータ（通常のストロークオブジェクト 2 1 0）として取り扱われるべきでなく、既存のストロークオブジェクト 2 1 0 に対していくつかの操作オペレーション（例えば、スライスする、消去する、展開する、削除する、コピーする、拡大する、など）を適用するためのコマンドであることを明示する。入力 3 が受信されると、インクデータ生成部 1 2 0 は、新しいストロークオブジェクト # j と、この新しいストロークオブジェクト # j が原因となって既存のストロークオブジェクト # 0 ~ # i に適用される操作オブジェクトとを生成する。操作情報（「入力 3」）は、アプリケーション部 3 0 0-1 をサポートするアプリケーション内で定義されたスイッチ、ボタンなどをユーザが選択することによって生成され、インクデータ生成部 1 2 0 に入力される。

20

【0086】

インクデータ生成部 1 2 0 は、ペンイベントデータ（「入力 1」）、コンテキスト情報（「入力 2」）、及び操作情報（「入力 3」）を受信し、（図 5 のポイント「D」に）ストロークオブジェクト 2 1 0、描画スタイルオブジェクト 2 3 0、操作オブジェクト 2 7 0、及びメタデータオブジェクト 2 5 0 を含む「インクデータ」（インクデータ 2 0 0）を生成する。インクデータ生成部 1 2 0 のさらなる詳細は、図 6 を参照して後に記述されるであろう。

30

【0087】

さらに図 5 を参照すると、インクデータ処理部 1 0 0 のインクデータ整形部 1 4 0 は、インクデータ生成部 1 2 0 からポイント「D」を介してインクデータを受信し、アプリケーション部 3 0 0-1 から受信されるフォーマット選択情報（F m t - S e l）に従って選択されるフォーマットにて、該インクデータを出力する。

40

【0088】

具体的には、インクデータ整形部 1 4 0 は、インクデータ通信部 1 4 4 及び記録フォーマットデータ処理部 1 4 2 を含む。インクデータ通信部 1 4 4 は、ストロークメッセージフォーマット（S M F）のインクデータ 2 0 0 を（図 5 の「F」を介して）送信し、（図 5 の「F _ i n」を介して）受信するよう構成される。ストロークメッセージフォーマット（S M F）は、ネットワーク上の他の（遠隔地にある）装置に対して、インクデータ 2 0 0 を（例えばリアルタイムで）伝達することに適したフォーマットである。記録フォーマットデータ処理部 1 4 2 は、インクデータをストロークファイルフォーマット（S F F）に整形するように構成される（図 5 の「E」を参照）。ストロークファイルフォーマット（S F F）は、インクデータ 2 0 0 をより永久的な記憶媒体に記憶するのに適したフォ

50

ーマットである。

【0089】

グラフィック処理部300は、ストロークオブジェクト210、描画スタイルオブジェクト230、操作オブジェクト270、及びメタデータオブジェクト250を含むインクデータ200を「D」を介して受信し、「H」を介して、定義された解像度レベルであって、ピクセルごとの色（例えば、RGB）の値を含む一組のピクセル値を出力する。例えば、グラフィック処理部300は、ストロークオブジェクト210を構成するポイントオブジェクト（p1～pn）を受信し、曲線補間アルゴリズムに従う制御点として使用されるポイントオブジェクトの間に曲線を挿入し、結果として得られるストロークオブジェクト210のパスをディスプレイ113上に、Direct X（登録商標）やOpenGL（登録商標）ライブラリのような関連するGPUライブラリを用いて描画（レンダリング）する。

10

【0090】

様々な実施の形態によれば、グラフィック処理部300は、当業者に知られたキャットマルローム曲線アルゴリズムやポリベジエ曲線アルゴリズムのような適切な曲線補間アルゴリズムに従う曲線を挿入するために、受信されたストロークオブジェクト210内に含まれるポイントオブジェクトを使用する。

【0091】

さらに、本発明の代表的な実施の形態によれば、グラフィック処理部300は、ストロークオブジェクト210内に含まれる「開始パラメータ」の値301及び「終力パラメータ」の値303を参照して、ストロークオブジェクト210の表示を行う。別の言葉で言えば、グラフィック処理部300は、ストロークオブジェクト210のうち、「開始パラメータ」の値301及び「終力パラメータ」の値303によって描写された（バインドされた）部分のみをレンダリング（表示）する。ここで使用される表示（される）とは、最終的に表示されることを意味する。定義された部分が表示対象であるか否かを設定するために、様々な技術が使用され得る。例えば、GPUライブラリに供給する頂点情報に表示対象でない所定の部分のための頂点情報を含まないようにすることで、該所定部分のためのピクセルデータを生成しないようにする方法が使われ得る。他の例としては、100%完全にストロークオブジェクト210を再生しつつも、表示対象でない所定の部分を透明に設定する方法が使われ得る。

20

30

【0092】

アプリケーション部300-1は、図1のアプリケーション#1のような、インクデータ処理部100と動的又は静的にリンクする1以上のユーザアプリケーションを含む。例えば、アプリケーション部300-1は、いずれも本発明の実施の形態によるインクデータ200を使用するリアルタイム会議アプリケーション、書類生成アプリケーション、描画アプリケーションなどを含むことができる。アプリケーション部300-1内の各アプリケーションは、例えば、ユーザがインクデータ処理部100に対して操作情報（「入力3」）を入力することを可能にするユーザインターフェイス（UI）を提供する。

【0093】

図6は、インクデータ生成部120及びインクデータ整形部140を含むインクデータ処理部100の機能ブロック図である。

40

【0094】

インクデータ生成部120は、ストロークオブジェクト操作部122、メタデータオブジェクト操作部124、描画スタイルオブジェクト操作部126、及び操作オブジェクト操作部128を含む。これらはそれぞれ、集合的に本発明の実施の形態によるインクデータ200を構成するストロークオブジェクト210、メタデータオブジェクト250、描画スタイルオブジェクト230、及び操作オブジェクト270を操作し、生成する。

【0095】

ストロークオブジェクト操作部122は、ペンイベントデータ（「入力1」）を受信し、ストロークオブジェクト210（図6の「D」を参照）を生成する。ストロークオブジ

50

ェクト操作部122は、上記したように生値型のポイントオブジェクトを生成し、ストロークオブジェクト210を形成する。代表的な実施の形態では、ストロークオブジェクト操作部122は、ポイントオブジェクトの生成を開始する前にペンイベントデータの全体が受信されるのを待つ代わりに、ペンイベントデータが入力されるに従って連続的にポイントオブジェクトを生成する。ストロークオブジェクト操作部122は、図7を参照して以下により完全に詳細に記述されるように、グラフィック処理部300（図5を参照）又はインクデータ整形部140に対し、「D」を介して、生成したポイントオブジェクトを連続的に出力する。アプリケーション300-1は、生値型のインクデータ200を出力することと、例えばアプリケーション300-1がリアルタイム通信を実施するか、或いは、高圧縮のベクトルデータを必要とするかに応じて最適化された値型のインクデータ200を出力することとを切り替えるよう、ストロークオブジェクト操作部122を制御することができる。

10

【0096】

メタデータオブジェクト操作部124は、ペンストロークの開始（すなわち、「ペンダウン」）を示すペンイベントデータ（「入力1」）の受信、又は、（例えばスライスすることに応じた）新しいストロークオブジェクト210の生成に応じてコンテキスト情報（「入力2」）を処理し、著者情報、日時情報などのような非描画関連情報を抽出する。メタデータオブジェクト操作部124は、抽出したメタデータを含むメタデータオブジェクト250を、対応するストロークオブジェクト210と関連付けて生成する。

20

【0097】

描画スタイルオブジェクト操作部126は、ペンダウンを示すペンイベントデータ（「入力1」）の受信、又は、新しいストロークオブジェクト210の生成に応じてコンテキスト情報（「入力2」）を処理し、ストロークオブジェクト210をディスプレイ上に表現するために必要となる描画関連情報を抽出する。描画スタイルオブジェクト操作部126は、描画スタイルオブジェクト230（例えば、シェイプフィルオブジェクト232及び粒子散布オブジェクト234）を、対応するストロークオブジェクト210と関連付けて生成する。

【0098】

操作オブジェクト操作部128は、操作情報（「入力3」）の受信に応じて、既存のストロークオブジェクト210の全体又は一部に対して適用されるべき操作又は変形のオペレーション（例えば、図2に示した「選択（変形）」オブジェクト272及びスライスオブジェクト274）を定義する操作オブジェクト270を生成する。

30

【0099】

図6では、2つの破線矢印「M1（ローカル）」及び「M2（リモート）」が、操作オペレーションの方向、すなわち、ターゲットであるストロークオブジェクト210が操作されるべきなのか、或いは、変形されるべきなのか、を示している。示されるように、操作オブジェクト270によって定義される操作オペレーションは、ストロークオブジェクト操作部122内に存在するストロークオブジェクト210に対してローカル（M1）に適用されてもよいし、ネットワークに結合された遠隔地の受信装置内のような外部ネットワーク上に存在するストロークオブジェクト210に対して、インクデータ整形部140を介してリモート（M2）に適用されてもよい。

40

【0100】

図6では、ストロークオブジェクト210、メタデータオブジェクト250、及び描画スタイルオブジェクト230は、インクデータ整形部140の記録フォーマットデータ処理部142及びインクデータ通信部144に対して入力され、一方で、操作オブジェクト270はインクデータ通信部144のみに入力され、142には入力されないように説明されている。初めの3つは、永久的又は半永久的に記憶されることが好ましく、それ故、ストロークファイルフォーマット（SFF）、SVGフォーマット、InkMLフォーマットなど、記憶装置に適した形に整形される。また、新たなストロークオブジェクト210が生成されるとき、ストロークオブジェクト210と、関連するメタデータ及び描画ス

50

タイルオブジェクトは、ネットワークを介して受信側に伝達され、したがって、インクデータ通信部 144 及び記録フォーマットデータ処理部 142 の両方で処理される。これに対し、操作オブジェクト 270 は、既存のストロークオブジェクト 210 に適用されるためのいくつかの操作的オペレーションを定義するものであることから、本来的に一時的なものである。一旦、操作的オペレーションが既存のストロークオブジェクト 210 に適用される（コミットされる）と、その操作オブジェクト 270 はメモリから追い出される。したがって、操作オブジェクト 270 は、典型的には、ネットワークを介する送信に適したストロークメッセージフォーマット（SMF）に整形され、ストロークファイルフォーマット（SFF）には含まれない。

【0101】

インクデータ整形部 140 の記録フォーマットデータ処理部 142 は、それぞれインクデータ 200 を異なる記録フォーマット（SFF、非特許文献 1 の InkML、非特許文献 3 の SVG、非特許文献 4 の HTML5 など）で出力するための複数の処理部 142-1, 142-2 を含む。例えば、処理部 142-1 は、SFF のインクデータ 200 を出力するように構成され、SFF 形式でファイルフォーマットされたデータをシリアル化するために、グーグルのプロトコルバッファ (<https://developers.google.com/protocol-buffers/>) 及びメッセージを採用することとしてもよい。処理部 142-2 は、例えば、非特許文献 1 の「トレース」の定義と、非特許文献 3 の「パス」又は非特許文献 4 の「キャンバスパス」の定義との間の違いのような、SFF ファイル、非特許文献 1 の InkML、及び非特許文献 3 の SVG の間の違いを吸収するためのフォーマット変換処理を行うこととしてもよい。

【0102】

図 7 は、インクデータ処理部 100 内のストロークオブジェクト操作部 122 の機能ブロック図である。ストロークオブジェクト操作部 122 は、ますます多くのペンイベントデータが入力されるに従い、グラフィック処理部 300 又はインクデータ整形部 140 に対し、生値型のストロークオブジェクト 210 を構成するポイントオブジェクトを連続的に出力する能力を有する。ストロークオブジェクト操作部 122 は、記憶装置 770 を含むか、又は記憶装置 770 と結合される。

【0103】

ストロークオブジェクト操作部 122 は、開始点と終了点を設定する開始／終了パラメータ設定部 122A と、装置のタイプに基づいて適切なパスビルダを選択するパスビルダ設定部 122B と、いくつかのポイントオブジェクトが部分的に形成されたストロークオブジェクト 210 に付加されるべきかを制御する対ストローク追加部 122D と、ストロークオブジェクト 210 の末尾にあるギャップ（図 8 の「遅延」）を埋めるサフィックス付加部 122E とを含む。

【0104】

生成され、メモリ 770 に記憶される断片化データは、ストロークオブジェクト 210 の断片として、リアルタイム伝送のために使用される。断片化データは、第 2 の実施の形態において以下により完全に記述されるように、バイト又は時間の単位ごとに送信される。

【0105】

開始／終了パラメータ設定部 122A は、ペンダウンイベント（ペンストロークの開始）又はペンアップイベント（ペンストロークの終了）の検出に応じて、開始パラメータ 301 及び終了パラメータ 303 をそれぞれの初期デフォルト値に設定する。例えば開始パラメータ 301 は、ペンダウンイベントに応じてそのデフォルト値「0」に設定され、終了パラメータ 303 は、ペンアップイベントに応じてそのデフォルト値「1」に設定される。これらのパラメータの初期値が記憶装置 770 に記憶される必要はなく、例えば、これらのパラメータがデフォルト値に設定されていることを暗示するフラグの形態で設定されればよい。

【0106】

パスビルダ部 1 2 2 B は、コンテキスト情報（「入力 2」）に含まれる入力ダイナミクス設定の値 7 0 1 に基づいて、入力処理部 1 1 1 から出力されるペンイベントデータのタイプに適したパスビルダを選択するように構成される。例えば、もしペンイベントデータが筆圧値を含むのであれば、筆圧に基づいてストローク幅（W）及び透明度（A）を導出する能力を有する第 1 の（圧力）関数 f 1 を含む圧力パスビルダ 1 2 2 B 1 が選択される。一方、もしペンイベントデータが筆圧値を含まないのであれば、速度パスビルダ 1 2 2 B 2 が選択される。速度パスビルダ 1 2 2 B 2 は、ペンの移動スピードに基づいてストローク幅（W）及び透明度（A）を導出する能力を有する第 2 の（速度）関数 f 2 を含む。ペンの移動スピードは、ペンイベントデータに含まれる点座標又はタイムスタンプの変化量から決定される。言い換えれば、速度パスビルダ 1 2 2 B 2 は、圧力パスビルダ 1 2 2 B 1 で用いられる圧力値の代わりに、速度値を用いている。すべてのペンイベントデータは、圧力情報を含むタイプ（タイプ 1）と、圧力情報を含まないタイプ（タイプ 2）のいずれかに分類され得るので、すべてのタイプのペンイベントデータが圧力パスビルダ 1 2 2 B 1 又は速度パスビルダ 1 2 2 B 2 のいずれかによって処理されるであろう。この理由、並びに、圧力パスビルダ 1 2 2 B 1 及び速度パスビルダ 1 2 2 B 2 がどのように動作するのか、については、以下で第 3 の実施の形態を参照しながら記述されるであろう。

【0 1 0 7】

ストロークオブジェクト操作部 1 2 2 はまた、ストロークオブジェクト 2 1 0 が生成され、完全なものとなる前に、コンテキスト情報（「入力 2」）に含まれる平滑化使用情報に基づき、該ストロークオブジェクト 2 1 0 に対して平滑化オペレーションの適用を開始する平滑器 1 2 2 C を有する。加速度平均、加重平均、指数平滑法、二重指数平滑法などのような任意の適切な平滑化オペレーションが使用され得る。

【0 1 0 8】

図 8 は、列 A 及び列 B において、平滑器 1 2 2 C により行われる平滑化オペレーションを説明する。列 A は図 7 のポイント「A」（平滑化前）に対応し、列 B は図 7 のポイント「B」（平滑化後）に対応する。列 A において、x 1 ~ x 1 0 は、ペンイベントデータから得られる 1 0 点の X 座標を表している。図 8 では、x 1 ~ x 1 0 は（1 0, 2 0, 3 0, 4 5, 6 0, 8 0, 9 0, 1 0 0, 1 1 0, 1 1 5）である。

【0 1 0 9】

列 B において、x 1' ~ x 1 0' は、平滑化オペレーションが適用された後の上記 1 0 点の X 座標を表している。図示した例では、次の指数平滑関数が適用される。ただし、フィルター強度 $\alpha = 0.5$ である。

【数 1】

$$X_i' = \alpha * X_{(t-1)} + (1-\alpha) * X_{(t-1)}' \quad (\text{Eq. 1})$$

【0 1 1 0】

図 8 では、x 1' ~ x 1 0' は（1 0, 1 5, 2 3, 3 4, 4 7, 6 3, 7 7, 8 8, 9 9, 1 0 8）である。

【0 1 1 1】

平滑器 1 2 2 C によって実行される平滑化オペレーションは、各点の修正された（平滑化された）位置を継続的に出力するべく、各点の座標が導出される都度、回転ベースで各点に適用される。したがって、ストロークオブジェクト操作部 122 は、全体としてのペンアップイベントが検出されるのを待つ代わりに、ペンダウンイベントが検出されたときから、「生値型」のストロークオブジェクト 2 1 0 の生成と出力を開始する。

【0 1 1 2】

列 B のポイント x 0' は、一対の制御点の間に補間曲線を定義するためにキャットマルローム曲線が使用される場合に加えられるものである。上で議論したように、キャットマルローム曲線は、スタートポイント（P i）、エンドポイント（P i+1）、スタートポ

10

20

30

40

50

イント (P_i) の「前」の点、エンドポイント (P_{i+1}) の「後ろ」の点を含む 4 つの制御点を使用して各曲線セグメントを定義する。それゆえ、ポイント $x_{1'}$ と $x_{2'}$ の間に開始曲線セグメントを定義するためにスタートポイント $x_{1'}$ が複製され、それによって、 $x_{1'}$ と $x_{2'}$ の間に曲線セグメントを定義するための点 $x_{1'}$, $x_{2'}$, $x_{3'}$ とともに使用され得る新しいポイント $x_{0'}$ が ($x_{1'}$ と同じ位置に) 生成される。新たなポイント $x_{0'}$ の一は、 $x_{1'}$ と $x_{2'}$ の間のキャットマルローム曲線が入力されたストロークに最もフィットする位置となるよう調整され得る。シンプルに $x_{1'}$ (p_1) の値を複製することにより $x_{0'}$ (p_0) の値を生成することにより、プロセスは直ちに $x_{0'}$ の位置を定義し、 $x_{0'}$ (p_0) から $x_{1'}$ (p_1) へのベクトルの構成要素をゼロにセットすることができる。そのようなプロセスは、リアルタイム実行のために適しており (p_0 を生成するために p_2 を待つ必要がない)、かつ、 $x_{1'}$ (p_1) と $x_{2'}$ (p_2) の間の曲線セグメントの曲率に過度に影響する (例えば、一方側又は他方側に押し出す) こともない。

10

【0113】

再び図 7 を参照すると、対ストローク追加部 122D は、いくつかのポイントオブジェクトが確立されて記憶装置 770 に格納され、そしてそれ故、出力されるストロークオブジェクト 210 の部分的なデータに加えられ得るのかを決定する。図示した実施の形態では、ポイントオブジェクト $P_1 \sim P_{n-1}$ が記憶装置 770 内に確立され、グラフィック処理部 300 又はインクデータ整形部 140 (ストロークオブジェクトの全体が完成するのを待つ代わりに) に出力されるストロークオブジェクト 210 の部分的なデータに加えられることが決定されている。グラフィック処理部 300 は、部分的に形成されたストロークオブジェクトを表示することができる。別の言葉で言えば、グラフィック処理部 300 は、初期の点から始まるストロークオブジェクトを、その成長とともに表示する。インクデータ整形部 140 (又はインクデータ通信部 144) は、確立された部分的なデータを、ストロークオブジェクトの断片化データとして伝送フォーマットにフォーマットし、そして送信することができる。断片化データの送信方法は、第 2 の実施の形態でより詳しく説明されるであろう。

20

【0114】

図 7 及び図 8 を参照すると、サフィックス付加部 122E は、入力されたストロークオブジェクト 210 の終端位置 (列 A の x_{10}) と、平滑化されたストロークオブジェクト 210 の終端位置 (列 B の $x_{10'}$) との間のギャップ (又は「遅延」) を埋める。図 8 では、列 C が「遅延」を埋める (サフィックスを付加する) オペレーションを図示している。図示した例では、平滑化オペレーションの後、ストロークオブジェクト 210 の終端において、もともと入力された位置 x_{10} (115) と、平滑化された位置 $x_{10'}$ (108) との間に「遅延」が生成されている。ストロークオブジェクト操作部 122 は、コンテキスト情報 (「入力 2」) に含まれる平滑化使用情報 705 の内容に基づいて、サフィックスを付加する動作を行うか行わないかを決定する。サフィックス付加オペレーションはまた、新たなポイントオブジェクトがストロークオブジェクト 210 に加えられる都度、発動され得る。この場合、サフィックス付加オペレーションは、グラフィック処理部 300 に対し、一時的な視覚上のプレビューとして使用され得るポイントオブジェクトを用意する。サフィックス付加オペレーションによって新たに生成されたポイントは、まだ最終的なストロークオブジェクト 210 の一部にはなっておらず、そしてそれ故、明確に加えられるまでの間、インクデータ整形部 140 (又はインクデータ通信部 144) によって無視される。

30

40

【0115】

図示した実施の形態においては、サフィックス付加オペレーションが実行されると、ストロークオブジェクト操作部 122 は $x_{11'}$ 、 $x_{12'}$ 、及び $x_{13'}$ に新たなポイントオブジェクトを加えている。ポイント $x_{12'}$ は、列 A のオリジナル入力の最新位置 x_{10} (115) と同じ位置に加えられる。ポイント $x_{11'}$ は、ポイント $x_{10'}$ と $x_{12'}$ の間の平滑化されたポイントに加えられる。最終的に、この例では制御点の各対の間に補間曲線を定義するためにキャットマルローム曲線が使用されることから、 $x_{11'}$ と $x_{12'}$

50

2'の間にキャットマルローム曲線である終了曲線セグメントを定義するために必要となる新たなポイント x_{13}' を生成するために、終端のポイント x_{12}' が(x_{12}' と同じ位置に)複製される。新たなポイント x_{13}' の位置は、 x_{11}' と x_{12}' の間のキャットマルローム曲線が入力されたストロークに最もフィットする位置となるよう調整され得る。また、サフィックス付加オペレーションが実行されないときでさえも、もしキャットマルローム曲線が使用されるならば、列Bの最新の平滑化ポイント x_{10}' は、新たなポイント x_{10}' (新)を生成するために複製されることができ、そしてそれは、この場合、 x_9' と x_{10}' の間の最新の曲線セグメントを定義するための制御点として x_8' 、 x_9' 、 x_{10}' とともに使用され得る。 x_{10}' の値を生成するために単純に x_9' の値を複製することにより、 x_8' (p_8)と x_9' (p_9)との間の曲線セグメントの曲率に過度に影響を与える(例えば、一方側又は他方側に押し出す)ことなく、プロセスは迅速に中立の位置に x_{10}' の位置を定義することができる。

【0116】

図8では、列Dは、平滑化され(列Aから列B)、終端にサフィックスが付加され(列Bから列C)、そして対ストローク追加部122Dの制御の下で継続的に出力されるストロークオブジェクト210を図示している。この例のストロークオブジェクト210は、各曲線セグメント($P_i \sim P_{i+1}$)が4つの制御点(P_{i-1} 、 P_i 、 P_{i+1} 、 P_{i+2})で定義され、結果としての曲線がすべての制御点を通るキャットマルローム曲線アルゴリズムによる補間曲線を生成するために定義される。したがって、ストロークオブジェクト210は、4つの制御点 x_0' 、 x_1' 、 x_2' 、 x_3' によって定義される開始曲線セグメント($x_1' \sim x_2'$)を含み、4つの制御点 x_{10}' 、 x_{11}' 、 x_{12}' 、 x_{13}' によって定義される終了曲線セグメント($x_{11}' \sim x_{12}'$)を含む。ストロークオブジェクト210はまた、開始曲線セグメント($x_1' \sim x_2'$)のための開始パラメータ301を含み、この開始パラメータ301は、開始/終了パラメータ設定部122Aによって「0.0」のデフォルト値に設定される。「0.0」のデフォルト値は、開始曲線セグメント($x_1' \sim x_2'$)が最初のポイント x_1' から完全に表示される(レンダリングされる、表現される)ことを意味する。ストロークオブジェクト210はまた、終了曲線セグメント($x_{11}' \sim x_{12}'$)のための終了パラメータ303を含み、この終了パラメータ303は、開始/終了パラメータ設定部122Aによって「1.0」のデフォルト値に設定される。「1.0」のデフォルト値は、終了曲線セグメント($x_{11}' \sim x_{12}'$)が最後のポイント x_{12}' まで完全に表示されることを意味する。

【0117】

図9は、インクデータ200を生成するために図6のインクデータ生成部120によって実行されるサンプル処理を説明するフローチャートである。処理は、インクデータ生成部120がペンイベントデータ(「入力1」)を受信するところから始まる。ステップS901において、ストロークオブジェクト操作部122は、入力1として、位置座標(x 、 y)と、以下に示す3種類のタイミングのうちの1つを示すタイミング情報とを含むペンイベントデータを受信し、決定されたタイミングに従って処理を実行する。

- 1) 指又はポインティングデバイス(例えば、ペン型装置)のようなポイントが他の(センシングする)物体とコンタクト状態に入ったときであるペンドアウンタイム。
- 2) ペンドアウンタイムとペンアップタイムの間であるペン移動タイム。
- 3) ポインタが他の(センシングする)物体から離れた(引き離された)ときであるペンアップタイム。

【0118】

<1. ペンドアウンタイムにおける処理フロー>

イベントタイプがペンドアウンイベントを示す「ダウン動作」である場合、ステップS910においてストロークオブジェクト操作部122が、新たに生成されるストロークオブジェクト210の開始曲線セグメントの開始パラメータ301をデフォルト値(「0,0」)にセットする。上述したように、デフォルト値(「0,0」)は、開始曲線セグメントがその最初の値から完全に表示されるべきことを定義する。同時に、ストロークオブジ

ェクト操作部 1 2 2 はまた、ストロークオブジェクト 2 1 0 の終了曲線セグメントの終了パラメータ 3 0 3 を、終了曲線セグメントもまたその最後の値まで完全に表示されるべきことを定義するデフォルト値（「1, 0」）にセットしてもよい。

【0 1 1 9】

ステップ S 9 1 2 では、ストロークオブジェクト操作部 1 2 2 は、コンテキスト情報（図 7 の「入力 2」）に含まれる入力ダイナミクス設定パラメータ 7 0 1 に基づき、ストロークオブジェクト 2 1 0 をビルドするために使用される複数のパスビルダの中から一のパスビルダ（例えば、図 7 の 1 2 2 B 1 又は 1 2 2 B 2）を選択する。

【0 1 2 0】

ステップ S 9 1 4 では、ストロークオブジェクト操作部 1 2 2 は、コンテキスト情報（「入力 2」）に含まれるチャンネル数パラメータ 7 0 3 に基づき、上記ステップ S 9 1 2 で選択されたパスビルダから出力されるべき一組のパラメータを決定する。出力されるべき一組のパラメータは、例えば、 (x, y, W, A) 、 (x, y, W) 、 (x, y, A) 、又は (x, y) であってよい。ただし、 (x, y) はポイントオブジェクトの x 、 y 座標であり、 W はストローク幅の値であり、 A は透明度（又は不透明度）を示すアルファ（ α ）である。2 次元座標 (x, y) に「 z 」値を加え、3 次元座標を生成してもよい。

【0 1 2 1】

ステップ S 9 1 6 では、ストロークオブジェクト操作部 1 2 2 が、コンテキスト情報（「入力 2」）に含まれる平滑化使用パラメータ 7 0 5 に基づき、選択されたパスビルダから出力される一組のパラメータに平滑化オペレーションが適用されるべきか否かを決定する。平滑化使用パラメータ 7 0 5 はまた、複数のパラメータのうちのどれに平滑化オペレーションが適用されるかを示すこととしてもよい。

【0 1 2 2】

ステップ S 9 1 8 で平滑化の適用が示されたとき（ステップ S 9 1 6 の YES）、指示された平滑化処理が実行される。図 7 は、平滑化処理が (x, y, W) パラメータに適用される一方、「 A 」（アルファ）パラメータに適用されない場合を図示している。コンテキスト情報（「入力 2」）は、ストロークオブジェクト操作部 1 2 2 が所望の強度レベルの平滑化処理を選択するために使用し得る入力センサ 1 1 0 のサンプリングレートの情報を追加で含むことができる。例えば、入力センサ 1 1 0 のサンプリングレートを低下させる（例えば、1 秒当たり 1 0 0 サンプルに対する 1 秒当たり 1 0 サンプル）とき、より大きな平滑度の値（効果）を有するより強い平滑化処理が選択され得る。

【0 1 2 3】

ステップ S 9 1 9 では、ここまでで使用されたセッティングパラメータが、インクデータ 2 0 0 の属性として出力される。パラメータは、例えば、ストロークオブジェクト 2 1 0 内に含まれるポイントオブジェクトが平滑化されるか否か（S 9 1 6 が YES であったか NO であったか）、又は、使用され得る平滑化フィルターのタイプ若しくは強さを示す。このパラメータに基づき、ストロークオブジェクト 2 1 0 内に含まれるポイントオブジェクトが平滑化されるか否か、（さらに）平滑化されるべきか否か、また、例えば署名検証アプリケーションなどにおいて使用され得る正確な入力データとして扱われ得るかが決定されることができる。

【0 1 2 4】

ステップ S 9 2 0 では、図 8 の列 B を参照して上で説明したように、最初のポイント（制御点） x_1' が、キャットマルローム曲線としての x_1' と x_2' の間の開始曲線セグメントを定義するための新たな制御点 x_0' を生成するために複製される。

【0 1 2 5】

ステップ S 9 7 0 では、インクデータ処理部 1 0 0 が、ローカルコンピュータのインクデータ処理部 1 0 0 によって生成されたインクデータ 2 0 0 を共有している（例えば、受信している、操作している）可能性のある他の（リモートの）ユーザ又はコンピュータが存在するか否かを決定する。

【0 1 2 6】

10

20

30

40

50

もしそのような他のユーザが存在しているなら、インクデータ処理部100はステップS972において、アプリケーション部300-1(図5参照)から受信されるフォーマット選択(Fmt-Sel)パラメータに基づき、インクデータ200がストロークメッセージフォーマット(SMF)内に出力されるようにインクデータ通信部144を制御する。インクデータ通信部144は、最初に、リモートユーザの受信装置が(部分的な)ストロークオブジェクト210を描くために必要な初期の点座標と描画スタイルオブジェクト230を含む部分的な(断片化された)メッセージであるメッセージDATA__INK__BGNO S1201(図12参照)を出力する。メッセージDATA__INK__BGNO S1201を受信した受信装置は、ストロークオブジェクト210の残りの部分を受信する前の段階で直ちに、受信した描画スタイルオブジェクト230を用いて特定される形状、色など、ストロークオブジェクト210の最初の部分のレンダリング(ディスプレイ)を開始することができる。

10

【0127】

<2. ペン移動タイムにおける処理フロー>

フローチャートのステップS901に戻り、インクデータ生成部120は、他の新たなイベントデータ(「入力1」)を受信し、そのタイプがペンダウンイベント、ペン移動イベント、ペンアップイベントのいずれであるかを決定する。イベントタイプが、ストロークオブジェクト210内でペンが移動していることを示す「ムーブ動作」であるとき、インクデータ生成部120は、使用される特定の入力装置依存型のペンイベントデータ内に含まれるものとしてのタイムスタンプ及び/又は筆圧情報とともにx, y座標を受信し、そしてステップS930に進む。

20

【0128】

ステップS930では、ストロークオブジェクト操作部122が、コンテキスト情報(「入力2」)に含まれる平滑化使用パラメータ705に基づき、例えば受信された一組のパラメータ(x, y, W, A)に平滑化オペレーションが適用されるべきか否かを決定する。平滑化パラメータ705は、追加で、複数のパラメータのうちのどれに平滑化オペレーションが適用されるかを示すこととしてもよい。ステップS930の動作は、上述したステップS916のものと同じである。

【0129】

平滑化の適用が示される場合(ステップS930のYES)、ステップS932において、示された平滑化処理が実行される。

30

【0130】

ステップS934では、ストロークオブジェクト操作部122が、ポイントオブジェクトのうちのいくつが確立され、出力対象のストロークオブジェクト210の部分的なデータに加えられるために記憶装置770に格納されるかを決定するために対ストローク追加部122Dを使用する。このステップにおいて対ストローク追加部122Dはまた、部分的なデータにポイントオブジェクトの値を追加する前に、その値を変更することができる。例えば、対ストローク追加部122Dは、インクが足りなくなっているボールペンをシミュレートするため、ランダムな基準でアルファパラメータの値を変更することができる。

40

【0131】

ステップS974では、上述したステップS970と同様、インクデータ処理部100が、インクデータ処理部100によって生成されるインクデータ200をリアルタイムで共有している他の(リモートの)ユーザが存在しているか決定する。

【0132】

もしそのような他のユーザが存在しているなら、インクデータ処理部100はステップS976において、上記ステップS972で生成されたメッセージDATA__INK__BGNO S1201に含まれる最初のポイントオブジェクトに続くポイントオブジェクトを含む部分的な(断片化された)メッセージであるメッセージDATA__INK__MVDOS1203(図12参照)を生成して出力するためにインクデータ通信部144を使用す

50

る。メッセージ DATA__INK__MVDOS 1203 に追加されるべきポイントオブジェクトの数は、上記ステップ S 934 において対ストローク追加部 122D により決定される。ストロークオブジェクト 210 のサイズに依存して、複数の DATA__INK__MVDOS メッセージが生成され出力され得る。DATA__INK__MVDOS メッセージを受信するリモートユーザの受信装置は、ストロークオブジェクト 210 の最初の部分に続けて、ストロークオブジェクト 210 の中間部のレンダリング（ディスプレイ）を継続する。

【0133】

＜3. ペンアップタイムにおける処理フロー＞

フローチャートのステップ S 901 に戻り、インクデータ生成部 120 は、他の新たなイベントデータ（「入力 1」）を受信し、そのタイプを決定する。イベントタイプがペンアップイベントを示す「アップ動作」である（すなわち、ストロークオブジェクト 210 の描画が完了し、ポイントが除去される）とき、ストロークオブジェクト操作部 122 はステップ S 950 において、受信した一組のパラメータ、例えば（x, y, W, A）に対して平滑化オペレーションが適用されるか否かを、どのパラメータに平滑化オペレーションが適用されるかということとともに決定する。ステップ S 950 の動作は、上述したステップ S 916 及び S 930 のそれと同じである。

【0134】

平滑化の適用が示される場合（ステップ S 950 の YES）、ステップ S 952 において、示された平滑化処理が実行される。また、平滑化使用パラメータ 705 がそう示すとき、ストロークオブジェクト操作部 122 は追加で、図 8 の列 C に示したサフィックス付加オペレーションを実行する。平滑化使用パラメータ 705 の内容によっては、サフィックス付加オペレーションは必ずしも実行されない。また、平滑化オペレーションが実行されないとき、サフィックス付加オペレーションは必要なく、実行されない。

【0135】

ステップ S 953 では、図 8 の列 C に図示したように、キャットマルローム曲線である x 11' と x 12' の間の終了曲線セグメントを定義するための新たな制御点 x 13' を生成するために終了ポイント（制御点）x 12' が複製される。

【0136】

ステップ S 954 では、ストロークオブジェクト操作部 122 が終了曲線セグメント（x 11' ～ x 12'）の終了パラメータ 303 を、終了曲線セグメントがその終了ポイント x 12' まで完全に表示されるべきであることを示すデフォルト値（「1. 0」）にセットする。このステップは、上述したステップ S 910 において既に終了パラメータ 303 がデフォルト値にセットされているときにはスキップされ得る。

【0137】

ステップ S 978 では、上述したステップ S 970 及び S 970 と同様に、インクデータ処理部 100 が、インクデータ処理部 100 によって生成されたインクデータ 200 をリアルタイムで共有している他の（リモートの）ユーザが存在するか否かを決定する。

【0138】

もしそのような他のユーザが存在しているならば、インクデータ処理部 100 はステップ S 980 において、サフィックスが付加されたストロークオブジェクト 210 のポイントオブジェクトの最後の組を含む部分的な（断片化された）メッセージであるメッセージ DATA__INK__ENDOS 1205（図 12 を参照）を生成及び出力するために、インクデータ通信部 144 を使用する。DATA__INK__ENDOS 1205 を受信するリモートユーザの装置は、それがストロークオブジェクト 210 のための最後のメッセージであることを理解し、ストロークオブジェクト 210 の描画動作を完了する。

【0139】

上述した本発明の実施の形態によるインクデータ 200 を生成し通信するための方法及びシステムは、継続的にペンイベントデータを入力し、部分的なストロークオブジェクト 210 が確立されると同時にこれを出力する。インクデータ処理部 100 からインクデー

タ 200 を受信するリモートユーザの受信装置は、ストロークオブジェクト 210 の全体が受信されるのを待つことなく、各ストロークオブジェクト 210 のディスプレイを開始し、それが成長するとともにストロークオブジェクト 210 のディスプレイを続ける。

【0140】

本発明の様々な実施の形態によれば、平滑化オペレーションは、入力されたペンイベントデータパラメータに選択的に適用される。キャットマルーロム曲線アルゴリズムのような適切な曲線補間アルゴリズムが、平滑化されたポイントオブジェクトを制御点として用いて補間曲線をビルドするために適用される。

【0141】

図 10 は、図 5 のインクデータ処理装置の記録フォーマットデータ処理部 142 からポイント「E」に出力され得る、インターフェイス定義言語（IDL）で書かれたストロークファイルフォーマット（SFF）のプロト（模式）ファイルのサンプルを説明する図である。このプロト（模式）ファイルは、インクデータ 200 がストロークファイルフォーマット内において、バイトのシーケンスとしてどのように並べられているかを記述している。図示したプロトファイルの行 02～07 は、描画領域（又は描画「キャンバス」）に含まれる情報を記述するデータを含む。例えば、破線の矩形で囲んだ行 06 は、ストロークオブジェクト 210 が描画領域内で複数回繰り返されることを定義する。行 04 の「decimal Precision（十進数精度）」は、ストロークオブジェクト 210 のポイントオブジェクトの計算の正確さ／分解能を定義する。

【0142】

行 04 の「decimal Precision」は、好ましくは、望ましい正確さ及び又は分解能を示す対数値である。

【0143】

図示したプロトファイルの行 11～17 は、ストロークオブジェクト 210 を表している。例えば行 12 は、ストロークオブジェクト 210 が、繰り返される複数の「sint 32」型（可変バイトサイズのパケット）のポイントオブジェクトを含むことを示している。

【0144】

いくつかの実施の形態では、2 つ目以降のポイントオブジェクトのパラメータは、最初のポイントオブジェクト又は直前のポイントオブジェクトの対応するパラメータ値に対するオフセット（デルタ）値によって定義される。オフセット（関連）値の使用は、絶対値と比べて、ストロークオブジェクト 210 を形成する 2 つ目以降のポイントオブジェクトを定義するために必要となるデータの量の削減に資することができる。

【0145】

例えば、行 12 のポイントの座標（x, y）は、decimal Precision を利用する以下の処理に基づいて決定される。

【数 2】

1. 以下の変換により浮動型から int32 型に変換する。

$$x_{int} = (int) x_{float} * 10^{decimalPrecision};$$

2. 整数値に対してデルタ符号化が実行される。

$$x_{encoded}[0] = x_{int}[0];$$

$$x_{encoded}[i] = x_{int}[i - 1] - x_{int}[i]; i > 0$$

【0146】

処理 1。x float は、コンピュータ装置に記憶される float である。ポイントオブジェクト 212 の座標は、float 型や double 型のような相対的に多くのビットを有する浮動小数点型の値としてメモリ 770 に記憶される。x float は、10

10

20

30

40

50

の `decimalPrecision` 乗が掛けられる。その結果として得られる値のデータ型としては、`xfloat` は `int` 型に変換され（キャストされ）、それによって `xint` が生成される。

【0147】

処理 2。 `xint` のオフセットが導出される。導出されたオフセットは、「`sint32`」型のデータとして符号化される。

【0148】

破線の矩形で囲んだ行 13～14 は、ストロークオブジェクト 210 の開始パラメータ 301 及び終了パラメータ 303 を定義する。図示するように、これらパラメータ 301 及び 303 は、行 12 で定義されるストロークオブジェクト 210 を形成するポイントオブジェクトとは別に定義される。図示した例では、開始及び終了パラメータ 301 及び 303 は、`float` 型の値として表現され、それぞれ「0」及び「1」のデフォルト値にセットされる。後述するように、操作（スライシング）オペレーションがストロークオブジェクト 210 に適用されるとき、開始及び／又は終了パラメータは新しい値に変更され得る。

【0149】

行 15 の「`variableStrokeWidth`（可変ストローク幅）」は、ストロークオブジェクト 210 に含まれる複数のポイントオブジェクトの幅の値を記憶する。行 12 におけるポイントと同様に「`sint32`」型を使用し、2 つ目以降のポイントオブジェクトの幅はオフセット（デルタ）によって定義される。このパラメータの存在は、ストロークオブジェクト 210 に含まれる複数のポイントオブジェクトのそれぞれが個々にそれ自身の幅の値と関連付けられることを暗示する。別の言葉で言えば、もしこのパラメータが存在していなければ、ストロークオブジェクト 210 は行 16 の「`strokeWidth`（ストローク幅）」プロパティに記憶される固定幅を有する。

【0150】

図 11 は、ストロークファイルフォーマット（SFF）で書かれ、図 5 のインクデータ処理装置の記録フォーマットデータ処理部 142 からポイント「E」に出力され得るストロークオブジェクト 210 のサンプルを説明する図である。図示したストロークファイルフォーマットによるストロークオブジェクト 210 は、描画スタイルオブジェクト 230、フィルターパラメータ（平滑化使用、フィルター強度）、及び、図 8 の列 D に図示した座標 $x_0' \sim x_{13}'$ におけるポイントオブジェクト $p_0 \sim p_{13}$ を含む。開始パラメータフィールド 1101 は開始パラメータ 301 を含み、終了パラメータフィールド 1103 は終了パラメータ 303 を含む。フィールド「`x0`」（1105）は、ストロークオブジェクト 210 の最初のポイントオブジェクトの絶対 x 座標の値を含む。フィールド「`rel_x1`」（1107）は、最初のポイントオブジェクトの絶対 x 座標の値に対するストロークオブジェクト 210 の 2 つ目のポイントオブジェクトの x 座標のオフセット（デルタ）値を含む。

【0151】

図 12 は、図 5 のインクデータ処理装置のインクデータ通信部 144 からポイント「F」に出力され得る、ストロークメッセージフォーマット（SMF）内に格納される 3 つのメッセージと、図 5 のインクデータ処理装置のるっとワーク通信部 112 からポイント「G」に出力される 1 つのパケットとを説明する図である。

【0152】

`DATA__INK__BGNOS1201` メッセージは、図 9 のステップ S972 で出力されるもので、そのメッセージがストロークオブジェクト 210 の最初のメッセージであることを示す情報（例えば `BGNOS` 型。F101）と、インクデータ処理装置及びリモートユーザの受信装置の間で共有される共通描画領域を示す描画領域 ID（F102）と、描画領域内の複数のストロークオブジェクト 210の中からストロークオブジェクト 210 を識別するために使用されるストローク ID とを含む。F101、F102、F103 は、メッセージヘッダを構成する。

【0153】

DATA__INK__BGNOS1201メッセージはさらに、描画スタイルオブジェクト230 (F104) と、適用される平滑化フィルターに関連するフィルターパラメータ (図示せず) と、開始パラメータ及び終了パラメータ (F105__SP__EP) と、オプションで (もしスペースが許せば) ストロークオブジェクト210の部分形成するポイントオブジェクトの最初の断片化データのいくつか (F105__begin) とを含み、最後にCRCエラー訂正値が付加される。例えば、「F105__begin」フィールドは、図8のポイントオブジェクトp0~p3を含むことができる。F104、F105__SP__EP、F105__begin、及びCRCフィールドは、メッセージペイロードを構成する。

10

【0154】

F104がDATA__INK__BGNOS1201に含まれる理由は、第2の実施の形態と関連して後述される。平滑化フィルターに関連するパラメータは、1つ目のメッセージであるDATA__INK__BGNOS1201に含まれており、それにより、ストロークオブジェクト210を受信するデバイスは、ストロークオブジェクト210の受信を開始した時点で直ちに、ストロークオブジェクト210に含まれるポイントオブジェクトに対して平滑化フィルタリング処理を適用すべきか否かを決定することができる。

【0155】

DATA__INK__MVDOS1203メッセージは、図9のステップS976で出力されるもので、メッセージ型フィールド (「MVDOS」) F101、描画領域IDフィールドF102、及びストロークIDフィールドF103を含むメッセージヘッダを含む。DATA__INK__MVDOS1203メッセージはまた、DATA__INK__BGNOS1201内に含まれていたポイントオブジェクトに続くポイントオブジェクトを含む2つ目の断片化データ (F105__moved) を含む。例えば、「F105__moved」フィールドは、図8のポイントオブジェクトp4~p8を含むことができる。1つ目のデータメッセージとは違い、DATA__INK__MVDOS1203メッセージは描画スタイルオブジェクト230 (F104) を含まず、F101において、続く (1つ目でない) タイプのデータメッセージであると識別される。DATA__INK__MVDOS1203メッセージは、1つ目のデータメッセージと同じ描画領域ID (F102) 及び同じストロークID (F103) を含む。

20

30

【0156】

DATA__INK__ENDOS1205メッセージは、図9のステップS980で出力されるもので、ストロークオブジェクト210のための最後のデータメッセージであり、かつ、例えば図8のポイントオブジェクトp9~p13であり得るストロークオブジェクト210の最後の断片化データ (F105__end) を含む。DATA__INK__ENDOS1205メッセージは、F101において、最後のデータメッセージ (ENDOS) であると識別され、1つ目のデータメッセージと同じ描画領域ID (F102) 及び同じストロークID (F103) を含む。最後のデータメッセージは、F109に、著者情報のような非描画関連情報を含むメタデータオブジェクト250を含む。

【0157】

上記した3タイプのデータメッセージは、図5のインクデータ処理装置のインクデータ通信部144からポイント「F」に出力される。図12の最後の列のパケット「G」は、これら3タイプのデータメッセージのすべてをパケットペイロードとして含むパケットであり、図5のインクデータ処理装置のネットワーク通信部112 (Tx, Rx) からポイント「G」に出力される。

40

【0158】

図13Aは、図5のインクデータ処理装置のグラフィック処理部300に対してポイント「D」を介して入力される、キャットマルローム曲線補間動作を受けたストロークオブジェクトを説明する図である (図13のストロークオブジェクト210はまた、上述したインクデータ整形部140に対してもポイント「D」を介して入力され得る)。図13B

50

は、図 1 3 A のストロークオブジェクト 2 1 0 が、図 5 のインクデータ処理装置のディスプレイ 1 1 3 上に描画される（レンダリングされる）ために、どのようにグラフィック処理部 3 0 0 からポイント「H」に出力されるかを説明する図である。

【0 1 5 9】

図 1 3 A では、ストロークオブジェクト 2 1 0 は、上記図 8 に図示した $x_0' \sim x_{13}'$ に対応するポイントオブジェクト $p_0 \sim p_{13}$ を含む。ストロークオブジェクト 2 1 0 は、キャットマルローム曲線補間動作を課され、したがってポイントオブジェクト $p_0 \sim p_{13}$ のすべてが、共同で図 1 3 に示した曲線を形成する補間曲線セグメントを生成するための制御点として使用される。例えば、ポイント $p_0 \sim p_3$ は、 p_1 と p_2 の間の曲線セグメントを生成するための制御点として使用され、ポイント $p_1 \sim p_4$ は、 p_2 と p_3 の間の曲線セグメントを生成するための制御点として使用される、などである。図示するように、結果として得られる曲線は、制御点 $p_0 \sim p_{13}$ のすべてを通過する。

【0 1 6 0】

キャットマルローム曲線の 1 つの特徴は、各曲線セグメントが 4 つの制御点によって完全に定義されるが故に、制御点を動かすことの効果が局地的であるということである。例えば、図 1 3 A は、 p_6 と p_7 の間の曲線セグメント 1 3 0 1 が 4 つの制御点 p_5 、 p_6 、 p_7 、 p_8 で定義されることを図示している。一の制御点を動かすことは、最大で 4 つの曲線セグメントに強い影響を与え得るが、曲線を形成している残りの曲線セグメントには影響しない。例えば、 p_8 を動かすことは、最大で、 $p_6 \sim p_7$ 、 $p_7 \sim p_8$ 、 $p_8 \sim p_9$ 、 $p_9 \sim p_{10}$ という 4 つの曲線セグメントに強い影響を与える可能性がある。キャットマルローム曲線の「局地的制御」という特徴は、結果として得られる 2 つの新たなストロークオブジェクト 2 1 0 が元のストロークオブジェクト 2 1 0 の（形の）情報を最小限のデータ量で必要に応じてできるだけ多く維持することが望まれるストロークオブジェクトのスライス操作をサポートするうえで適している。別の言葉で言えば、「局地的制御」という特徴は、（一端から他端まで曲線の形状を完全に維持するために）結果として得られる各スライスが最小数の制御点で元の形を維持することを可能にする。例えば、図 1 3 A のストロークオブジェクト 2 1 0 が p_6 と p_7 の間の曲線セグメントでスライスされるとき、1 つ目のスライスは制御点 $p_0 \sim p_8$ のみを維持する必要がある、2 つ目のスライスは制御点 $p_5 \sim p_{13}$ のみを維持する必要がある。ポリベジエ曲線もまた「局地的制御」という特徴を有し、したがってスライス操作をサポートするうえで適している。しかしながら、ポリベジエ曲線は、キャットマルローム曲線と違い、曲線に沿っていない制御点を必要とする（すなわち、曲線が制御点のすべてを通過しない）。それら曲線の外にある制御点を計算し、記憶しなければならないことは、記憶装置のスペースを要求する余分な計算であり、ストロークオブジェクト 2 1 0 のポイントオブジェクトによってすべての制御点が用意されるキャットマルローム曲線では要求されないものである。この違いは、計算上の要求が少ないキャットマルローム曲線を、インクデータの生成、操作、描画、及び通信のリアルタイムアプリケーションをサポートするうえでより適したものにしている。

【0 1 6 1】

図 1 3 B は、図 5 のポイント「H」においてグラフィック処理部 3 0 0 から出力されるときの図 1 3 A のストロークオブジェクト 2 1 0 の実際のレンダリング（描画）例を図示している。図 1 3 B は、開始曲線セグメント 1 3 0 5 の開始パラメータ SP （「0. 0」）によって示される開始位置から終了曲線セグメント 1 3 0 7 の終了パラメータ EP （「1. 0」）によって示される終了位置までに及ぶ実際の描画の範囲 1 3 0 9 を図示している。ここで、描画対象となる 1 つ目の曲線セグメント 1 3 0 5 は p_1 と p_2 の間であり、 p_0 と p_1 の間ではないことに注意が必要である。 p_0 は p_1 と p_2 の間の曲線セグメントを定義するための制御点としてのみ使用されるからである。同様に、描画対象となる最後の曲線セグメント 1 3 0 7 は p_{11} と p_{12} の間であり、 p_{12} と p_{13} の間ではない。 p_{13} は p_{11} と p_{12} の間の曲線セグメントを定義するために使用される制御点に過ぎないからである。

【0162】

<インクデータ操作（スライシング）>

以下、本発明のシステム及び方法によって可能とされる様々な実施の形態によるスライス操作について、図14～図19を参照して記述する。

【0163】

図14は、既存のストロークオブジェクト210に適用されるスライス操作のフローチャートである。スライス操作は、ストロークオブジェクト操作部122と操作オブジェクト操作部128によって共同で実行される。図示した例は、新たに描画されたストロークj403が既存のストロークi401をスライスするという、図4Bに示したスライス操作を前提としている。

【0164】

まず、インクデータ生成部120（ストロークオブジェクト操作部122）が新たなペインイベントデータ（「入力1」）を受信する。ステップS1401では、それぞれステップS1403及びステップS1405から始まる2つの処理スレッドが並行して実行される。

【0165】

1つ目の処理スレッドでは、ステップS1403においてインクデータ生成部120が、上記図9に記述した処理によりストロークオブジェクト（ストロークj）を生成する。2つ目の処理スレッドでは並行して、ステップS1405においてインクデータ生成部120（操作オブジェクト操作部128）が操作情報（「入力3」）が、新たなストロークオブジェクト210を生成するために1つ目の処理スレッドによって使用されるペインイベントデータ（「入力1」）に関連付けられているかを判定する。例えば、操作情報（「入力3」）は、関連するストロークオブジェクト210がスライス操作を実行する予定であることを示し得る。

【0166】

もしそのような「入力3」がないのであれば（ステップS1405のNo）、処理は終了に進み、新たに生成されたストロークオブジェクト（ストロークj）は通常のストロークオブジェクト210として取り扱われる。もしストロークオブジェクト210に関連する「入力3」があるのであれば（ステップS1405のYes）、ストロークjは、1以上の既存のストロークオブジェクト210に対してスライス操作を実行するための操作オブジェクト270として取り扱われる。

【0167】

<スライス操作>

ステップS1407から始まりステップS1423に至るループは、既存のストロークオブジェクトのそれぞれ（stroke_1～ストロークj-1）に対して繰り返される。この例では、新たに生成されたストロークjによって表現されるスライス操作は、既存のストロークオブジェクトのそれぞれに対して適用される。操作オブジェクト270に対し、位置という見地から隔離されたストロークをスキップするために適用され得る様々な技術がある。例えば、ストロークオブジェクト操作部122は、領域内のストロークセグメントを含むインデックスを維持することができる。そのようなインデックスは、操作オブジェクト操作部128により、不必要な交差計算をスキップするために使用され得る。

【0168】

ステップS1407は、ループをセットアップする。

【0169】

ステップS1409において操作オブジェクト操作部128は、図15Aに示すように、新しいストロークj403が既存のストロークi401（ $i < j$ ）と交差しているかを判定する。図15は、既存のストロークiの2つの制御点p6とp7の間にある交点P_intersect_Midで既存のストロークiと交差する新たなストロークjを図示している。図示した例では、新しいストロークjが幅を有していることから、交点は、新しいストロークjの中心線（P2, P3, P4を通過する実線で示す）と既存のストロー

10

20

30

40

50

ク i との交差点として計算される。

【0170】

図14に戻り、新しいストローク j が既存のストローク i と交差していないとステップ S1409で判定されるということは、スライス操作が既存のストローク i に適用されないということを意味する。プロセスは i を1インクリメントし、新しいストローク j が次の既存のストローク $i+1$ と交差しているかを判定する。

【0171】

ステップ S1409において新しいストローク j が既存のストローク i と交差していると判定された場合、ステップ S1411のスライス操作が既存のストローク i に適用される。

【0172】

ステップ S1413では、図15Bに示すように、操作オブジェクト操作部128が、計算された交点 $P_intersect_Mid(55, 100)$ と新しいストローク j の「幅」(3.58)とに基づき、新しいストローク j のエッジ(破線)と既存のストローク i との間の2つの交差点 $P_intersect_L$ 及び $P_intersect_R$ をそれぞれ導出する。図15Bでは、既存のストローク i の p_6 は(47, 100)、 p_7 は(63, 100)にあり、一方、新しいストローク j の P_2 は(60, 110)、 P_3 は(50, 90)にある。上述したように、交点 $P_intersect_Mid$ は線 $p_6_p_7$ と線 $P_2_P_3$ の間の交差点として計算される。 $p_6_p_7$ 方向に沿う新しいストローク j の幅は「幅」/ $\sin \theta = 4.0$ (「導出幅」として導出される。ここで、 $\sin \theta = (110 - 90) \div \sqrt{(60 - 50)^2 + (110 - 90)^2}$) である。

【0173】

導出された4.0という幅はその後、導出された幅の半分を交点 $P_intersect_Mid(55, 100)$ に足し、或いは、得られた幅の半分を交点 $P_intersect_Mid(55, 100)$ から引くことによって2つの交差点 $P_intersect_L$ 及び $P_intersect_R$ を計算するために使用される。図示した例では、 $P_intersect_L$ は(53, 100)と算出され、 $P_intersect_R$ は(57, 100)と算出される。S1413で追加の計算を行うこともあり得る。例えば、もしストローク i とストローク j が交差領域で平坦でないのであれば、追加の計算は、 $P_intersect_L$ 及び $P_intersect_R$ の計算において実際の曲率を考慮に入れるべきである。

【0174】

図14では、ステップ S1415において操作オブジェクト操作部128が、図16A及び図17Aに示すように、スライス操作の結果として得られるストロークオブジェクト i の2つのインスタンスをそれぞれ生成する。これら2つのインスタンスは、「スライス」型である。図16Aに示す1つ目のスライスは、 $p_2 \sim p_7$ の間に曲線セグメントを定義する制御点 $p_0 \sim p_8$ を含み、図17Aに示す2つ目のスライスは、 $p_6 \sim p_{12}$ の間に曲線セグメントを定義する制御点 $p_5 \sim p_{13}$ を含む。図16Aの1つ目のスライスは、ストローク i がスライスされた p_6 と p_7 の間に、終了曲線セグメントとしてのホールセグメント1801を含む。一方、図17Aの2つ目のスライスは、開始曲線セグメントとしてのホールセグメント1801を含む。1つ目のスライスは、開始曲線セグメント $p_1_p_2$ のためのデフォルト開始パラメータ「0.0」を維持するが、今や、新たに生成された終了曲線セグメント $p_6_p_7$ のために新たな終了パラメータを必要としている。同様に、2つ目のスライスは、終了曲線セグメント $p_{11}_p_{12}$ のためのデフォルト終了パラメータ「1.0」を維持するが、今や、新たに生成された開始曲線セグメント $p_6_p_7$ のために新たな開始パラメータを必要としている。

【0175】

ステップ S1417において操作オブジェクト操作部128は、図18Aに示すように、図16Aの1つ目のスライスのための新たな終了パラメータ (new_EP1) 値30

10

20

30

40

50

3を導出する。図18Aにおいて、1つ目のスライスの新たな終了パラメータ1803（「new__EP1」）は、1つ目のスライスの終了曲線セグメントp6__p7内に位置するP__intersect__L（53, 100）の位置を示す値である。新たな終了パラメータ1803は、絶対値（例えば（53, 100））であってもよいし、相対値（例えば、p6からX方向に+6）であってもよいし、p6とp7の間の距離に対するp6とP__intersect__Lの間の距離の割合（例えば、 $6/16=0.375$ ）であってもよい。割合は、例えば図5のグラフィック処理部300のような様々なセクションにより実行される補間計算において、さらなる処理なしで使用され得ることから、有用である。

【0176】

図16Bは、1つ目のスライスを定義するパラメータのデータ構造を説明する。このデータ構造は、「0.0」（デフォルト値）である開始パラメータ（float型）と、上記ステップS1417で導出された「0.375」（割合）である終了パラメータ（float型）とを含む。図16Bでは、行11が、そのデータ構造が、ひとたびファイナライズされる（コミットされる）とストロークオブジェクト210となる「スライス」オブジェクトを定義することを示す。行12の「slice__from Index」は、スライスの開始ポイントオブジェクトの番号を示すインデックス値（例えば整数）である。図16の例では、1つ目のスライスオブジェクトがポイントp0から始まることから、「slice__from Index」は「0」である。行13の「slice__to Index」は、スライスの終了ポイントオブジェクトの番号を示すインデックス値（例えば整数）であり、1つ目のスライスオブジェクトがポイントp8で終わることから、図16では「8」である（最後のポイントp8は、終了曲線セグメントp6__p7のための制御点である）。

【0177】

図14に戻り、ステップS1419では、操作オブジェクト操作部128が図18Aに示すように、図17Aの2つ目のスライスのための新たな開始パラメータ（new__SP2）を導出する。図18Aでは、2つ目のスライスの新たな開始パラメータ1805（「new__SP2」）は、2つ目のスライスの開始曲線セグメントp6__p7内に位置するP__intersect__R（57, 100）の位置を示す値である。新たな開始パラメータは、絶対値（例えば（57, 100））であってもよいし、相対値（例えば、p6からX方向に+10）であってもよいし、p6とp7の間の距離に対するp6とP__intersect__Rの間の距離の割合（例えば、 $10/16=0.625$ ）であってもよい。

【0178】

図17Bは、2つ目のスライスを定義するパラメータのデータ構造を説明する。このデータ構造は、「1.0」（デフォルト値）である終了パラメータ（float型）と、上記ステップS1419で導出された「0.625」（割合）である開始パラメータ（float型）とを含む。図17Bでは、行21が、そのデータ構造が、ひとたびファイナライズされるとストロークオブジェクト210となる「スライス」オブジェクトを定義することを示す。行22の「slice__from Index」は、2つ目のスライスオブジェクトがポイントp5から始まることから「5」である（最初のポイントp5は、開始曲線セグメントp6__p7のための制御点である）。行23の「slice__to Index」は、2つ目のスライスオブジェクトがポイントp13で終わることから「13」である。

【0179】

図14に戻り、ステップS1421では、操作オブジェクト操作部128が、図16Bの1つ目のスライス（第1のスライスオブジェクト274）と、図17Bの2つ目のスライス（第2のスライスオブジェクト274）とを、それぞれ第1の新たに生成されたストロークオブジェクト210及び第2の新たに生成されたストロークオブジェクト210にレンダリングするためにファイナライズ（又はコミット）し、それらを記憶装置770に

10

20

30

40

50

格納する。ファイナライズを行うステップ S 1 4 2 1 は、図 6 に示すように、操作オブジェクト操作部 1 2 8 とストロークオブジェクト操作部 1 2 2 の間で発生する「M 1（ローカル）」動作である。ストロークオブジェクト操作部 1 2 2 は、この時点で、第 1 及び第 2 の「スライス」オブジェクト及び／又は元のストロークオブジェクト i を破棄することとしてもよい。

【0180】

図 1 6 C は、ファイナライズされた第 1 のストロークオブジェクト 2 1 0 のレンダリングされた（ディスプレイされた）パスを説明し、図 1 7 C は、ファイナライズされた第 2 のストロークオブジェクト 2 1 0 のレンダリングされた（ディスプレイされた）パスを説明している。図 1 6 C に示すように、p 1 と p 6 の間の曲線セグメントは完全にディスプレイされるが、p 6 と p 7 の間の終了曲線セグメントについては、終了パラメータ 3 0 3（0. 3 7 5）によって示される終了点までの部分のみがディスプレイされる。p 6 から終了パラメータ 3 0 3 によって示される終了点までの部分は、元のストロークオブジェクト 2 1 0 の形状を正確に辿っている。なぜなら、新たに生成された第 1 のストロークオブジェクト 2 1 0 は、元のストロークオブジェクト 2 1 0 に含まれていたものと同じ、p 6 と p 7 の間の終了曲線セグメントを定義する制御点 p 5 ～ p 8 を維持しているからである。同様に、図 1 7 C に示すように、p 7 と p 1 2 の間の曲線セグメントは完全にディスプレイされるが、p 6 と p 7 の間の開始曲線セグメントについては、開始パラメータ 3 0 1（0. 6 7 5）によって示される開始点で始まる部分のみがディスプレイされる。開始パラメータ 3 0 1 によって示される開始点から p 7 までの部分は、元のストロークオブジェクト 2 1 0 の形状を正確に辿っている。なぜなら、新たに生成された第 2 のストロークオブジェクト 2 1 0 は、元のストロークオブジェクト 2 1 0 に含まれていたものと同じ、p 6 と p 7 の間の開始曲線セグメントを定義する制御点 p 5 ～ p 8 を維持しているからである。

【0181】

図 1 4 では、ステップ S 1 4 2 2 において操作オブジェクト操作部 1 2 8 が、新たに生成された第 1 及び第 2 のストロークオブジェクト 2 1 0 をリモートユーザの受信装置に向け、以下で図 2 0 及び図 2 1 を参照してより完全に記述するように、送信することができる。送信ステップ S 1 4 2 2 は、図 6 に示した「M 2（リモート）」動作であり、操作オブジェクト操作部 1 2 8 と、インクデータ整形部 1 4 0 を介して、インクデータ処理部 1 0 0 の外部のネットワークにより結合されたリモートユーザの受信装置との間で発生する。代替として、ステップ S 1 4 2 2 において、操作オブジェクト操作部 1 2 8 が新たに生成したストロークオブジェクト 2 1 0 の代わりに 2 つのスライスオブジェクト 2 7 4 を送信することとしてもよい。このケースでは、ステップ S 1 4 2 1 は受信装置において実行される。これについては、以下で図 2 0 を参照しながら完全に記述する。

【0182】

ステップ S 1 4 2 3 において、複数の既存のストロークオブジェクトのそれぞれに対して繰り返されるループ処理が完了する。

【0183】

ステップ S 1 4 2 7 では、操作オブジェクト操作部 1 2 8 が、上記 S 1 4 1 1 のスライス操作ステップで生成及び使用されたすべてのスライスオブジェクト 2 7 4 を（もしまだであれば）流し去る（破棄する）。

【0184】

上述したスライス操作は、要するに、スライスされる元のストロークオブジェクト 2 1 0 を、スライス操作の結果として得られる 2 つの新たなストロークオブジェクト 2 1 0 に置き換えるというものである。したがって、2 つの新たなストロークオブジェクト 2 1 0 が生成された後には、元のストロークオブジェクト 2 1 0 は破棄されてよい。

【0185】

図 1 4 に記述された実施の形態では、操作情報（「入力 3」）と関連付けられた新たなストロークオブジェクト j が、1 以上の既存のストロークオブジェクト 2 1 0 に対してス

ライス操作を実行するよう構成される操作オブジェクト270として取り扱われる。別の言葉で言えば、ストロークオブジェクトjが操作オブジェクト270として使用される。

【0186】

別の実施の形態では、図14のS1411のスライス操作ステップの間に生成されるスライスオブジェクト274が、既存のストロークオブジェクト210を共有する1以上のコンピュータ(10-1-1, 10-1-2, 10-3, ...)内に存在する1以上の既存のストロークオブジェクト210に対してスライス操作を実行するための操作オブジェクト270として使用され得る。この場合、スライスオブジェクト274は破棄されない。操作オブジェクト270としてのスライスオブジェクト274の使用については、以下で、図20及び図21Aを参照しつつより完全に記述されるであろう。

10

【0187】

さらに別の実施の形態では、図16A、図17A、図18Aに示したホールセグメント1801を定義し、1以上の既存のストロークオブジェクト210に対してスライス操作を実行する操作オブジェクト270として使用されるホールセグメントオブジェクトが生成され得る。図18Bは、ホールセグメントを定義するパラメータのデータ構造を説明している。行01は、このデータ構造が、スライスオブジェクト274やストロークオブジェクト210とは異なる「ホールセグメント」オブジェクトを定義していることを示す。行02の「hole_from Index」は、ホールセグメントの開始ポイントオブジェクトの番号を示すインデックス値(例えば整数)である。この例では、ホールセグメントがp6とp7の間にあることから、「6」となっている。行03の「hole_end Parameter」は、図14のステップS1417で導出された「0.375」(割合。float型)であり、ホールセグメント1801によって生成された1つ目のスライスの終了曲線セグメントの終了点を示している。行04の「hole_to Index」は、ホールセグメントの終了ポイントオブジェクトの番号を示すインデックス値(例えば整数)である。この例では、ホールセグメントがp6とp7の間にあることから、「7」となっている。行05の「hole_start Parameter」は、図14のステップS1419で導出された「0.625」(割合。float型)であり、ホールセグメント1801によって生成された2つ目のスライスの開始曲線セグメントの開始点を示している。操作オブジェクト270としてのホールセグメントオブジェクトの使用については、以下で、図20及び図21Bを参照しつつより完全に記述されるであろう。

20

30

【0188】

上述した実施の形態では、P__Intersect__Lが1つ目のスライスの新しい終了点として使用され、P__Intersect__Rが2つ目のスライスの新しい開始点として使用されているが、他の実施の形態では、同一のポイントが1つ目のスライスの新しい終了点及び2つ目のスライスの新しい開始点の両方として使用されてもよい。例えば、計算の簡単のため、P__Intersect__Midが1つ目のスライスの新しい終了点及び2つ目のスライスの新しい開始点の両方として使用されてもよい。

【0189】

上述した本発明の実施の形態によれば、スライスされた曲線セグメント(例えば、「ホールセグメント」1801)のディスプレイは、1つ目のスライスのディスプレイが終了する終了点を定義する新たな終了パラメータ303と、2つ目のスライスのディスプレイが開始する開始点を定義する新たな開始パラメータ301とによって制御される。1つ目及び2つ目のスライスとともに、ホールセグメントの構造を完全に定義するデータを維持しており、1つ目のスライス及び2つ目のスライスの部分として表示するホールセグメントの部分を制限しているに過ぎない。したがって、レンダリングされる(ディスプレイされる)これらのスライスは、正確に、元のストロークオブジェクト210の形状を辿る。さらに、これらのスライスが元のストロークオブジェクト210内にあるものとしてのホールセグメントを完全に定義する同一のデータ(例えば制御点)を維持していることから、制御点の位置を再計算したり、新たな制御点を計算する必要がない。本発明のシステム及び方法は、好ましくは、キャットマルローム曲線やポリベジェ曲線のような、上述した

40

50

ように「局所的制御」という特徴を有する曲線補間アルゴリズムとともに用いられる。その後、スライス操作によって得られるスライスは、スライス操作から得られるホールセグメントを定義するための最小限のデータ量のデータ（最小数の制御点）を維持する必要がある。

【0190】

図19は、ストロークファイルフォーマット（SFF）で書かれ、図16C及び図17Cに表示した2つの新規生成ストロークオブジェクト210を含むファイルのサンプルを説明している。図19は、元のストロークオブジェクトファイルのサンプルを説明する図11と比較され得る。

【0191】

図19では、破線の箱1910が図16Cに示した第1の新規生成ストロークオブジェクト210を含むファイルを示している。第1のストロークオブジェクトファイルは、図16Cのポイントオブジェクトp0～p8に対応するポイントx0～x8を含む。開始パラメータフィールド1912はデフォルト値の「0.0」を含み、終了パラメータフィールド1912は、図14のステップS1417で導出された「0.375」という値を含む。破線の箱1920は、図17Cに示した第2の新規生成ストロークオブジェクト210を含むファイルを示している。第2のストロークオブジェクトファイルは、図17Cのポイントオブジェクトp5～p13に対応するポイントx5～x13を含む。開始パラメータフィールド1922は、図14のステップS1419で導出された「0.675」という値を含み、終了パラメータフィールド1912はデフォルト値の「1.0」を含む。第1のストロークオブジェクトファイル1910は、（図11の）元のストロークオブジェクトファイルの上書き又は変更されたインスタンスであってよく、独立したコピーを形成するために元のストロークオブジェクトファイルから複製されたものであってもよい。ここで使用している「新規生成」ファイルは、これらのファイルタイプのどちらをも意味してよい。第2のストロークオブジェクトファイル1920もまた、図19のフィールド1921に示すように、元のストロークオブジェクトファイルの複製されたコピーであってよい。

【0192】

図20は、図14のステップS1422におけるインクデータ伝送処理の詳細なフローチャートである。ステップS2012において、インクデータ処理部100は、コンテキスト情報（「入力2」）から（i）インクデータ処理部100によって生成されるインクデータ200（一組のストロークオブジェクト210）を共有するリモートユーザが存在するか否か、（ii）リモートユーザに対してインクデータ200を送信するために使用されるメッセージタイプ、を示す情報を取得する。

【0193】

ステップS2014では、インクデータ処理部100のインクデータ整形部140が、上記情報（i）に基づき、ローカルデバイスのインクデータ200が他のリモートのコンピュータに共有されているかを判定する。インクデータ200を使用しているリモートユーザ又はコンピュータが現在のところ存在しないのであれば、ステップS2022においてオプションで、インクデータ整形部140が2つの新規生成ストロークオブジェクト210を、（図19のSFFフォーマットファイルに対応する）図21Dに示すタイプDのメッセージ形式でバッファリングすることとしてもよい。リモートユーザが後にインクデータ処理部100のインクデータ生成セッションに参加する場合に、ファイル整形部140は、リモートユーザにタイプDのメッセージを送ることができる。この時点で、スライシング操作オブジェクトjを送信する必要はない。なぜなら、リモートユーザは、リアルタイムでスライス操作を見ておらず、スライス操作の結果、すなわち2つの新たに生成されたストロークオブジェクト210を受信することのみが必要であるからである。

【0194】

もしステップS2014でリモートユーザが存在すると判定された場合、ステップS2016においてインクデータ処理部100が、上記情報（ii）に基づき、リモートユー

ザに対してインクデータ200を送信するために使用するべきメッセージのタイプを判定する。適したメッセージのタイプは、リモートユーザの受信装置（図22参照）によってサポートされる操作オペレーションのタイプ、特定のアプリケーションにおける時間遅延の許容量、利用可能な送信リソースの量などに応じて選択され得る。例えば、情報（ii）が、リモートユーザの受信装置がいかなる操作（例えばスライス）オペレーションもサポートしていないことを示している場合には、ステップS2022に進み、インクデータ整形部140が、スライス操作の結果、すなわちタイプDのメッセージで書かれた新規生成ストロークオブジェクトのみをバッファリングして、リモートユーザに向けて送信する。

【0195】

ステップS2016において、もし上記情報（ii）により、リモートユーザの受信装置が、受信装置上の1以上のストロークオブジェクト210のスライスするためにストロークオブジェクト210を操作オブジェクト270として実行することをサポートしていることが示されていれば、処理はステップS2020に進む。ステップS2020では、インクデータ整形部140は、図21Cに示すメッセージタイプCで書かれた操作（スライス）ストロークオブジェクトjを送信するためにインクデータ通信部144を使用することができる。

【0196】

図21Cに示すタイプCのメッセージは、型フィールドF101、描画領域IDフィールドF102、及び、ストロークオブジェクトjを識別するストロークIDフィールドF103を含むヘッダを有する。型フィールドF101は、このメッセージが、スライス操作を実施（実行）する「スライサー」タイプであることを示している。この例では、F101における「スライサー」タイプの表示は、関連するストロークオブジェクトjが操作オブジェクト270であることを示す操作情報（「入力3」）を構成する。タイプCのメッセージのペイロード部分は、ポイントオブジェクトp1～pnを含む操作（スライス）ストロークオブジェクトjを含む。操作オブジェクト270としてのストロークオブジェクトjを送信するためにタイプCのメッセージを使うことの利点の1つは、送信側と受信側の両方において、リアルタイムで同時に同じスライス操作を発生させられるという点である。送信側は、ストロークオブジェクトjの生成と実質的に同時に、受信側に対して操作情報（「入力3」）を有するストロークオブジェクトjを送信することができ、したがって、両サイドで同時に、ストロークオブジェクトjによる同じ操作を実行できるからである。

【0197】

図20のステップS2016に戻り、もし上記情報（ii）により、リモートユーザの受信装置が、受信装置上の1以上のストロークオブジェクト210をスライスするためにスライスオブジェクト274又はホールセグメントオブジェクトを操作オブジェクト270として実行することをサポートしていることが示されていれば、処理はステップS2018に進む。ステップS2018では、インクデータ整形部140は、図21Aに示すメッセージタイプAで書かれた操作オブジェクト270としてのスライスオブジェクト274を送信するためにインクデータ通信部144を使用することができる。又は、インクデータ整形部140は、ステップS2018において、図21Bに示すメッセージタイプBで書かれた操作オブジェクト270としてのホールセグメントオブジェクト（図18B）を送信するためにインクデータ通信部144を使用することができる。

【0198】

図21Aに示すタイプAのメッセージは、型フィールドF101、描画領域IDフィールドF102、及び、ストロークオブジェクトjを識別するストロークIDフィールドF103を含むヘッダを有する。タイプAのメッセージのペイロード部分は、ストロークオブジェクトjにより1以上の既存のストロークオブジェクト210をスライスした結果として得られる1以上のスライスオブジェクト210のペアを含む。図21Aは、2つのそのようなペアを図示している。1つは、それぞれフィールドF111_31及びF111

10

20

30

40

50

— 3 2 に示す第 1 のスライス及び第 2 のスライスであり、これらはストロークオブジェクト j により既存のストローク 3 をスライスした結果として得られるものである。もう 1 つは、それぞれフィールド $F111_i1$ 及び $F111_i2$ に示す第 1 のスライス及び第 2 のスライスであり、これらはストロークオブジェクト j により既存のストローク i をスライスした結果として得られるものである（他のペアの記述は省略している）。タイプ A のメッセージの型フィールド $F101$ は、このメッセージが、スライス操作を実施（実行）する「スライサー」タイプであることを示している。この例では、 $F101$ における「スライサー」タイプの表示は、関連するストロークオブジェクト 274 のペアが操作オブジェクト 270 を形成することを示す操作情報（「入力 3」）を構成する。タイプ A のメッセージを受信した受信装置は、スライスオブジェクト 274 の各ペアを取り出し、ディスプレイ上に描画され得る 2 つの新たなストロークオブジェクト 210 を生成するために、スライス対象である元の（既存の）ストロークオブジェクト 210 を参照して、スライスオブジェクト 274 をファイナライズする。操作オブジェクト 270 としてのスライスオブジェクト 274 を送信するためにタイプ A のメッセージを使うことの利点の 1 つは、スライスオブジェクト 274 のデータサイズが一般的に、タイプ C のメッセージに含まれる、スライスするストロークオブジェクト 210（ストロークオブジェクト j ）のデータサイズや、タイプ D のメッセージに含まれる新規生成ストロークオブジェクト 210 のデータサイズより小さいという点である。

【0199】

図 21B のタイプ B のメッセージは、型フィールド $F101$ 、描画領域 ID フィールド $F102$ 、及び、ストロークオブジェクト j を識別するストローク ID フィールド $F103$ を含むヘッダを有する。タイプ B のメッセージのペイロード部分は、ストロークオブジェクト j により 1 以上の既存のストロークオブジェクト 210 をスライスした結果として得られる 1 以上のホールセグメントオブジェクトを含む。図 21B は、2 つのホールセグメントオブジェクトを図示している。1 つは、フィールド $F111_3H$ 内のホールセグメントストローク $_3$ であり、ストロークオブジェクト j により既存のストローク 3 をスライスした結果として得られるものである。もう 1 つは、フィールド $F111_iH$ 内のホールセグメントストローク i であり、ストロークオブジェクト j により既存のストローク i をスライスした結果として得られるものである（他のホールセグメントオブジェクトの記述は省略している）。タイプ B のメッセージの型フィールド $F101$ は、このメッセージが、スライス操作を実施（実行）する「スライサー」タイプであることを示している。この例では、 $F101$ における「スライサー」タイプの表示は、関連するホールセグメントオブジェクトが操作オブジェクト 270 を形成することを示す操作情報（「入力 3」）を構成する。タイプ B のメッセージを受信した受信装置は、上述したタイプ A のメッセージと同様、ディスプレイ上に描画され得る 2 つの新たなストロークオブジェクト 210 を生成するために、ホールセグメントオブジェクトを取り出し、スライス対象である元の（既存の）ストロークオブジェクト 210 を参照して実行する。操作オブジェクト 270 としてのホールセグメントオブジェクトを送信するためのタイプ B のメッセージの利点の 1 つは、ホールセグメントオブジェクトのデータサイズが一般的に、タイプ C のメッセージに含まれる、スライスするストロークオブジェクト 210（ストロークオブジェクト j ）のデータサイズや、タイプ D のメッセージに含まれる新規生成ストロークオブジェクト 210 のデータサイズより小さいという点である。

【0200】

図 22 は、本発明の実施の形態による、ネットワークを介してインクデータ 200 をリモート受信するように構成されたインクデータ受信装置の機能ブロック図である。

【0201】

この受信装置は、ネットワーク通信部 310（Tx, Rx）と、インクデータ処理部 100R と、アプリケーション部 300-2 と、グラフィック処理部 300R と、ディスプレイ 113R と、オペレーティングシステム 400-2 とを含む。代表的な実施の形態では、インクデータ処理部 100R は、送信側のインクデータ処理部 100 を実現するライ

ブラリに統合される。したがって、受信装置は一般に、インクデータ処理部 100 によって実行される機能に対応する相対機能を実行する。

【0202】

図 22 において、ポイント「D」、「E」、「F_in」、「G」、及び「H」は、送信側のインクデータ処理部 100 の各ポイントに対応する。図 22 において、M2'（リモート）は、送信側の M2（リモート）から送信された操作オブジェクト 270 が受信されるポイントを示す。M1'（ローカル）は、送信側の M1（ローカル）に対応するポイントを示し、そこでは、図 4 のステップ S1421 で実行されるスライスオブジェクト 274 をファイナライズする（コミットする）d が実行される。ただし、図 22 では、スライスオブジェクト 274（又は、スライスオブジェクト 274 を生成する操作ストロークオブジェクト 210）は内部生成されず、送信側から受信されるという点を除く。

10

【0203】

ネットワーク通信部 310（Tx, Rx）は、WAN 又は無線／有線の LAN インターフェイスを介してパケットを受信し、図 12 に記述したような様々なインクデータメッセージを取り出す。

【0204】

インクデータ処理部 100R は、インクデータ整形部 140R 及びインクデータ生成部 120R を含む。インクデータ整形部 140R は送信側のインクデータ整形部 140 に対応するもので、それと同様に、ポイント「E」を介して記録のタイプであるストロークファイルフォーマット（SFF）で書かれたインクデータ 200 を受信するように構成される記録フォーマット操作部 142R と、ポイント「F_in」を介して通信のタイプであるストロークメッセージフォーマット（SMF）で書かれたインクデータ 200 を受信するように構成されるインクデータ通信部 144R とを含む。データ通信部 144R は、受信したメッセージ（「入力 3」）のヘッダフィールド F101 に含まれる値に基づき、受信したメッセージが操作オブジェクト 270、すなわち、1 以上の既存のストロークオブジェクトに対して操作（スライス）オペレーションを実行するように構成された特殊なタイプのストロークオブジェクト 210（例えば、スライサタイプのストロークオブジェクト）、スライスオブジェクト 274、又はホールセグメントオブジェクトを含むかを判定する。

20

【0205】

インクデータ生成部 120R は、送信側のインクデータ生成部 120 に対応する。送信側のインクデータ処理装置とは異なり、受信装置は、ストロークオブジェクト 210 を生成するためにペンイベントデータを受信することはない。その代わりに、インクデータ生成部 120R は、ストロークオブジェクト 210、操作オブジェクト 270、描画スタイルオブジェクト 230、及びメタデータオブジェクト 250 のような様々なオブジェクトを受信する。インクデータ生成部 120R は、受信したストロークオブジェクト 210 を処理し（例えば操作し）、操作した（変形した）ストロークオブジェクト 210 を記憶装置 770 に格納するために、ストロークオブジェクト操作部 122 及び操作オブジェクト操作部 128 を使用する。

30

【0206】

グラフィック処理部 300R は、送信側のグラフィック処理部 300 により実行される処理に対応する処理を実行する。図 13A 及び図 13B に示すように、グラフィック処理部 300R は、ストロークオブジェクト 210 内に含まれるポイントオブジェクトを制御点として用い、キャットマルローム曲線アルゴリズムのような曲線補間アルゴリズムに従って補間曲線を生成することにより、ストローク（パス）を再構築する。グラフィック処理部 300R はさらに、ストロークオブジェクト 210 のために定義される開始パラメータ及び終了パラメータを参照し、ストロークオブジェクト 210 の表示範囲（部分）を制御する。例えば、もし開始パラメータが 0.5 であり、終了パラメータが 0.5 であったら、グラフィック処理部 300R は、開始曲線セグメントの midpoint からストロークオブジェクト 210 の描画（ディスプレイ）を開始し、終了曲線セグメントの midpoint でストロークオ

40

50

プロジェクト 210 のディスプレイを停止する。グラフィック処理部 300R はまた、ストロークオブジェクト 210 に関連付けて受信された描画スタイルオブジェクト 230 を参照し、レンダリングしたストロークオブジェクト 210 に幅、色、及び他のグラフィック属性を追加する。

【0207】

ディスプレイ 113R は、送信側のディスプレイ 113 に対応する。アプリケーション部 300-2 は、オペレーティングシステム 400-2 によってサポートされ、動的又は静的に、インクデータ処理部 100R を実現するライブラリにリンクされる。アプリケーション部 300-2 は、送信側のアプリケーション部 300-1 でサポートされるアプリケーションと同じ、又は、類似し、又は、異なるアプリケーションを含むことができる。

10

【0208】

図 23 は、本発明の実施の形態による図 22 の受信装置により実行され得る、操作（スライス）オブジェクトの受信処理を説明するフローチャートである。

【0209】

ステップ S2301 において、インクデータ整形部 140R は、ネットワーク通信部 310 を介して受信されるインクデータメッセージを受信し、メッセージヘッダ（「入力 3」）に含まれる型フィールド F101 から、各メッセージに含まれるメッセージのタイプを取り出す。

【0210】

ステップ S2303 において、インクデータ整形部 140R は、メッセージが標準のストロークオブジェクト 210 と操作オブジェクト 270 とのいずれを含むかを判定する。上述したように、操作オブジェクト 270 は、操作情報（「入力 3」）と関連付けられた、ストロークオブジェクト 210、スライスオブジェクト 274、又はホールセグメントオブジェクトのいずれであってもよい。

20

【0211】

ストロークオブジェクト 210 は、図 12 に示したようなストロークメッセージフォーマット（SMF）ファイルによって書かれることができ、又は、図 11 に示したようなストロークファイルフォーマット（SFF）ファイルによって書かれることができる。操作オブジェクト 270 は、図 21A～図 21C に示したように、SMF で書かれることが好ましい。

30

【0212】

標準のストロークオブジェクト 210 が受信されたと判定された場合、ステップ S2305 でストロークオブジェクト操作部 122 が、受信したストロークオブジェクト 210 を記憶装置 770R に追加する。

【0213】

操作（スライス）オブジェクトが受信されたと判定された場合、操作オブジェクト操作部 128 が、受信したメッセージ内に含まれる 1 以上の操作オブジェクトをステップ S2311 で取り出し、ステップ 2313 で、取り出した操作オブジェクトのリストを生成する。例えば、もし図 21A に示したタイプ A のメッセージが受信された場合、このリストは、フィールド F111__31、F111__32、F111__i1、及び F111__i2 に含まれるスライスオブジェクト 274 の 2 つのペア（他のペアの記述は省略する）を操作オブジェクトとして識別するものとなる。もし図 21B に示したタイプ B のメッセージが受信された場合、このリストは、フィールド F111__3H 及び F111__iH に含まれる 2 つのホールセグメントオブジェクト（他のホールセグメントオブジェクトの記述は省略する）を操作オブジェクトとして識別するものとなる。もし図 21C に示したタイプ C のメッセージが受信された場合、このリストは、ストロークオブジェクト j を操作オブジェクト 270 として識別するものとなる。

40

【0214】

続くステップ S2315～S2318 は、リスト内に含まれる各操作オブジェクトについて繰り返される。

50

【0215】

ステップS2315では、操作オブジェクト操作部128が操作オブジェクト270が適用されるべきターゲットストロークオブジェクト210を決定する。例えば、図21AのタイプAのメッセージのケースでは、フィールドF111__31、F311__32のスライスオブジェクト274のペアの形を取る操作オブジェクト270が既存のストローク3に適用され、フィールドF111__i1、F311__i2のスライスオブジェクト274のペアの形を取る操作オブジェクト270が既存のストロークiに適用されることが決定される。

【0216】

ステップS2317では、操作オブジェクト操作部128が、操作オペレーションを実行するためにステップS2315で識別したターゲットストロークオブジェクト210に対して操作オブジェクト270を適用し、完全に定義されたストロークオブジェクトを生成するために操作オペレーションをコミット（ファイナライズ）する。生成されたストロークオブジェクトはその後、ストロークオブジェクト操作部122の記憶装置770Rに格納される。ステップS2317の動作は一般に、図14示したステップS1421の動作に対応する。一言で言えば、操作オペレーションのコミット又はファイナライズは、操作オペレーションの結果を、結果として得られるストロークオブジェクトに反映させる。

【0217】

ステップS2318では、操作オブジェクト操作部128が、ステップS2313のリスト内に識別された操作オブジェクトのすべてがそれぞれのターゲットストロークオブジェクトに対して実行されたかを判定する。もしされていないならば、次の（残りの）操作オブジェクト270を処理するために、処理はステップS2315に戻る。

【0218】

もしリスト内に識別された操作オブジェクトのすべてがそれぞれのターゲットストロークオブジェクトに対して実行され、それらの操作オペレーションがストロークオブジェクトの新しいセットを生成するためにコミットされ（ファイナライズされ）たとしたら、操作オブジェクト操作部128はステップS2319において、操作オブジェクトとして使用されたスライスオブジェクト274及び（もしあれば）ホールセグメントオブジェクトを流し去る。

【0219】

送信側のインクデータ処理部100及び受信側のインクデータ処理部100Rを、オペレーティングシステム400-1又は400-2上で動作するソフトウェアライブラリとして記述しているが、インクデータ処理部100、100Rは、特定用途向け集積回路（ASIC）又はICのような、それとは異なる態様で実現されてもよい。

【0220】

したがって、第1の実施の形態によるインクデータの処理方法によれば、図91に説明したように、ストロークオブジェクト210の一部を変更又は操作（例えばスライス）し、1以上の他のコンピュータ装置に対してその変更／操作を送信することが可能になる。

【0221】

第2の実施の形態

本発明の第2の実施の形態は、共通描画領域を共有する複数の装置（伝送装置、中継装置、受信装置）の間でインクデータ200を通信する（送信する、中継する、受信して処理する、及びストリーミングする）ための方法及びシステムに向けられたものである。特に、この方法及びシステムは、共通描画領域内に異なる装置によってそれぞれ書き込まれたインクデータ200の複数のレイヤを、リアルタイムで、正しい通信順序で、かつ、適時に重ね合わせることを可能にする。

【0222】

本発明の第2の実施の形態は、複数のユーザが同時かつリアルタイムに共通描画領域（又はキャンバス）に手書き（フリーハンド）入力を行うことのできるリアルタイムコラボレーションアプリケーションの実現に特に適している。

10

20

30

40

50

【0223】

リアルタイムコラボレーションアプリケーションの使用のために、本発明は、インクデータ200の断片の伝送タイミングを制御する新規な通信／再生産スキームにより、知覚可能な時間遅延なしに共有できるフルセットの属性（色、トレース又はストロークの幅、レンダリング（描画）スタイルなど）を有するインクデータを生成する能力を有する方法及びシステムを提供する。

【0224】

一側面によれば、インクデータ200は、それぞれ異なるタイプの装置を用いて生成された（描画された）複数のストロークオブジェクトと、これらのストロークオブジェクトを特徴付ける描画スタイルオブジェクト230（例えば、どのタイプのペン先がストロークオブジェクトを描画するのには使用されるか）とを含み、そしてインクデータ200は、共通描画領域内でレンダリングされる。リアルタイムコラボレーションアプリケーションのようないくつかのアプリケーションは、厳格な（速い）時間に対する要求を有している。一方で、他の種類のアプリケーションは、そのような厳格な時間に対する要求を有しない。本発明の一側面による装置及び方法は、使用中の特定のアプリケーションの要求に従い、インクデータ200を適時に、正しい順序かつ正しいフォーマットで、送信／中継／受信するように構成される。

【0225】

一側面によれば、本発明は、それぞれ伝送装置と描画領域を共有する複数の受信装置と通信するために伝送装置によって実行される方法であり、伝送装置は、伝送装置の入力センサによって検出された入力操作のトレースを表すインクデータ200を複数の受信装置に送信する。この方法は一般に、（a）インクデータ生成ステップ、（b）メッセージ形成ステップ、（c）送信ステップ、という3つのステップを含む。インクデータ生成ステップは、（i）ポインタの動きに従って生成されるペンイベントデータ（入力1）を継続的に入力し、ストロークデータの断片化データを生成するステップであって、ストロークオブジェクトは、ポインタの前記動きのトレースを表すために複数のポイントオブジェクトを含み、断片化データは規定の単位Tごとに生成されるステップと、（ii）前記トレースの開始ポイントにおけるペンイベントデータの生成に対応するペンダウンタイムにおけるコンテキスト情報（入力2）に基づいて描画スタイルオブジェクト230を生成するステップであって、描画スタイルオブジェクト230は、ストロークオブジェクトの前記トレースの描画フォームを定義するステップとを含む。メッセージ形成ステップは、（i）描画スタイルオブジェクト230を含む第1のメッセージを生成するステップと、（ii）第1のメッセージに続く1以上の第2のメッセージを生成するステップであって、1以上の第2のメッセージは断片化データを含むステップとを含む。最後に、送信ステップは、定義された通信プロトコルに従って、第1のメッセージと1以上の第2のメッセージとを順に送信するステップを含む。

【0226】

他の側面によれば、中継方法は、伝送装置の入力センサによって検出された入力操作のトレースを表すインクデータ200を受信し、受信したインクデータ200を、それぞれ伝送装置と描画領域を共有する複数の受信装置に対して中継するために提供される。この方法は一般に、（a）受信ステップ、（b）制御ステップ、（c）データメッセージ中継ステップ、（d）送信ステップ、という3つのステップを含む。受信ステップは、描画領域に関する情報を含む制御メッセージを受信すること、及び、描画領域内にレンダリングされるインクデータ200を含むデータメッセージを受信することを含む。制御ステップは、描画領域を共有する複数の受信装置の通信アドレスを列举する接続リストを更新することを含む。データメッセージ中継ステップは、（i）データメッセージに含まれるストロークオブジェクト210に基づいて描画領域を更新することを許容するか否かを決定すること、（ii）もし更新が許容されるなら、前記接続リストに列举された受信装置に対して直接又は他の中継装置を介して中継されるべき新たなデータメッセージを生成すること、（iii）もし更新が許容されないなら、描画領域の更新要求が拒絶されたことを示

10

20

30

40

50

す拒否メッセージを生成すること、を含む。最後に、送信ステップは、受信装置に対し、通信パケットで新たなデータメッセージを送信することを含む。

【0227】

さらなる側面によれば、本発明は、一般に4つのステップを含むインクデータ200を受信する方法を提供する。第1のステップは、第1の断片IDが関連付けられたメッセージ内のストロークオブジェクト210の第1の断片を受信することを含む。第2のステップは、第3の断片IDが関連付けられたメッセージ内のストロークオブジェクト210の第3の断片を受信することを含む。なお、第3の断片IDは第1の断片IDと連続していない。第3のステップは、受信したストロークオブジェクト210の第1及び第3の断片に基づき、失われたストロークオブジェクト210の第2の断片を補間し、補間された第2の断片をディスプレイすることを含む。第4のステップは、ストロークオブジェクト210の終端を受信した後、全体としてのストロークオブジェクト210の再送信を要求するために、ストロークオブジェクト210のストロークIDを含む要求を送信することを含む。

10

【0228】

さらに他の側面によれば、インクデータ200が格納されるサーバを用いて複数のストロークオブジェクトを含むインクデータをストリーミングする方法が提供される。この方法は一般に2つのステップを含む。第1のステップは、格納されたストロークオブジェクトを順番に読み出すことを含む。第2のステップは、ストロークオブジェクトを、ストロークオブジェクトの連続的な部分にそれぞれ対応する定義されたタイミングで、サーバから1以上の受信装置に送信することを含む。

20

【0229】

第2の実施の形態の記述

第1の実施の形態を参照して上述した図1は、本発明の第2の実施の形態に従ってインクデータ200が生成され、通信され（送信され、中継され、受信され、処理され、流れ、など）得るシステムの全体を説明する。例えば、図1の装置10-3は、リアルタイムコラボレーションタイプアプリケーションであり得るアプリケーションサービス#2を用いて、リアルタイムに、装置10-1-1及び装置10-1-2からそれぞれ出力されるインクデータ200を結合（合成）することができる。装置10-3は、装置10-1-1及び装置10-1-2から出力されるインクデータ200を、互いに異なりかつ重ねられる複数のレイヤとして、その画面上にリアルタイムにレンダリング（描画）してもよい。

30

【0230】

生成され、本発明の実施の形態による異なる装置の間で通信されるインクデータ200は、様々なタイプの装置、コンピュータ、オペレーションシステム、又はアプリケーションにより、通信リソース上で共有される。

【0231】

図25は、インクデータ200のための情報モデルの実体関連図である。図25は、図25のストロークオブジェクトが開始パラメータ301又は終了パラメータ303を含む必要がない点で図2と異なっているが、その他は図2と同じである。本発明の実施の形態によるインクデータ200は、ストロークオブジェクト210（又はストロークオブジェクトセット202）（図3参照）と、レンダリング（描画、視覚化、ラスター化）時にストロークオブジェクト210の形及び色を制御するために必要となる情報を含む描画スタイルオブジェクト（セット）230とを含む。第1の実施の形態を参照して上述した図3B及び図3Cは、それぞれ、図3Aに示した同じストロークオブジェクト210の2つの異なる描画スタイルオブジェクト230によるレンダリング（描画）結果を図示している。インクデータ200はさらに、著者、ペンID、ローカルに取得される日時情報、GPSによって取得される位置情報のような、ストロークオブジェクト210を説明する非描画関連情報を含むメタデータオブジェクト（セット）250を含む。インクデータ200はさらに、既存のストロークオブジェクト210を操作する（例えば、選択する、回転す

40

50

る、スライスする) ために必要となる情報を含む操作オブジェクト (セット) を含む。そのような情報は、それぞれがストロークオブジェクト 210 の所望の操作をもたらすために、既存のストロークオブジェクト 210 に対して実行可能である複数の操作オブジェクトの形式で組織化される。他の操作オブジェクトは、例えば、操作オブジェクトの切り取り (消去)、削除、コピー、拡大、及び縮小を含むことができる。

【0232】

図 3A は、複数のポイントオブジェクト (ポイント 1 ~ ポイント n) を含むストロークオブジェクト 210 を説明する。このストロークオブジェクト 210 は、ポイントの動きのトレースを表す情報を含む。ポイントは、ペン型の道具又は指である。具体的には、ストロークオブジェクト 210 は、トレースを形成する複数のポイントオブジェクト (ポイント 1 ~ ポイント n) の座標を含む。各ポイントオブジェクトは、半径、色、透明 (不透明) 度のような属性に関連付けられ得る。

10

【0233】

複数のポイントオブジェクト (ポイント 1 ~ ポイント n) の座標は、ペイイベントデータ (又はポイント操作) を適切にサンプリングすることによって取得又は導出されるもので、ポイントの動きに従って生成される。必要に応じて、サンプリングされたポイント間の補間を行う。第 1 の実施の形態を参照して上述したように、キャットマルローム補間アルゴリズムのような適切な曲線アルゴリズムが、トレースを滑らかな曲線で表すために採用され得る。

【0234】

20

複数のポイントオブジェクトの (x, y) 座標を生成するために、一般に 2 つの方法がある。1 つ目の方法では、ペイイベントデータが入力されている間、ペイイベントデータから単位時間ごとに導出されるポイントが、「生値型」のポイントとして出力される。2 つ目の方法では、1 つのストロークを構成するすべてのポイントが入力された後、ストロークのための近似曲線 (キャットマルローム曲線、ベジエ曲線など) に代表されるより高次の機能が生成され、近似曲線を表現するために必要な最小数のポイントオブジェクトが「最適化ポイント型」のポイントとして得られる。本発明の一実施の形態によれば、(x, y) 座標を生成する 2 つの方法は、各アプリケーションによって要求されるインクデータ 200 の通信の時間的制約により、選択的に切り替えられる。

【0235】

30

図 3B 及び図 3C の異なるレンダリング結果に説明されるように、各描画スタイルオブジェクト 230 は、トレースがスクリーン又はディスプレイ上でレンダリング (描画又は表現) される際のユニークな形、すなわち、トレースがどのようにスクリーン又はディスプレイ上に現れるか、に関する情報を含む。したがって、同じストロークオブジェクト 210 をレンダリングするために異なる描画スタイルオブジェクトを使用すると、それらがどのように現れるかという観点で、ストロークオブジェクト 210 の描画が異なるという結果になる。トレースの形は、1 以上のポイントオブジェクトの形 (例えば、図 3B の円及び図 3C のフレーク又は花びら)、トレースの幅 (例えば、図 3B の各円の半径及び図 3C の各フレークのサイズ)、各ポイントオブジェクトに関連付けられた角度又はオフセット (図 3C 参照)、トレースの色 (又はポイントオブジェクトの色)、トレースの (又はポイントオブジェクトの) 透明度 / 不透明度、トレースのテクスチャ (又はポイントオブジェクトのテクスチャ) などによって定義され得る。ストロークオブジェクト 210 を描画するために必要とされる情報は、そのストロークオブジェクト 210 に関連付けられた描画スタイルオブジェクト 230 に含まれる。

40

【0236】

図 26 は、本発明の実施の形態による通信システム図の全体である。この通信システムは、伝送装置 10-1、中継サーバ 10-2、及び複数の受信装置 10-3, 10-4、
・ ・ ・、10-n を含む。これらは皆、図 1 に示した一群のライブラリリソース 10 を共有し使用する。中継サーバ 10-2 はまた、より増進されたパフォーマンスのために、グローバルに分散した中継サーバの (コンテンツデリバリーネットワークと同様の) ピアツ

50

ーピアネットワークの一部であり得る。図 26 のシステムは、本発明の実施の形態によるリアルタイムコラボレーションを実現するために、インクデータ 200 の送信、中継、及び受信の各方法を実行するのに適している。

【0237】

図 26 の伝送装置 10-1 は、図 1 の装置 10-1-1 又は装置 10-1-2 に対応する。

【0238】

伝送装置 10-1 は、入力センサ 110 と、図示しない入力処理部と、インクデータ処理部 100T と、アプリケーション部 300-1 と、通信部 112 (「Tx Rx」) とを含む。インクデータ処理部 100T は、第 1 の実施の形態の図 5 に示した 100T に対応する。アプリケーション部 300-1 は、インクデータ処理部 100T にリンクされ、一群のライブラリリソース 10 (図 1) によってサポートされる 1 以上のユーザアプリケーションを含む。図示した実施の形態では、アプリケーション部 300-1 は、リアルタイム通信機能を利用するリアルタイムアプリケーション #1 を実行する。

【0239】

中継サーバ 10-2 は、典型的には中継サーバであり、インクデータ交換サービスを具体化して提供する。図示した実施の形態では、中継サーバ 10-2 は、リアルタイムでインクデータ 200 を交換し、中継することにより、中継サーバ 10-2 から見て遠隔地に位置する伝送装置 10-1 及び受信装置 10-3, 10-4、・・・、10-n に便宜を与える。

【0240】

中継サーバ 10-2 は、通信部 (510 Rx、511 Tx)、インクデータ中継部 540、アプリケーションコントロール部 520、及び、キャンバスデータ (又は描画データ) を記憶するように構成されたリポジトリを含む。以下、これらについてより完全に記述する。本発明の一側面によれば、リポジトリは、継続的に更新される描画領域 (キャンバス) の最新の状態に関する情報を、図 28 に説明されるストロークファイルフォーマット (SFF) で記憶する。中継サーバ 10-2 は、すべての伝送/受信装置が描画領域の最新状態 (すなわち、「キャンバス」が現在どのように見えるか) を共有できるようにするために、伝送/受信装置から要求があると、要求した装置に対して共通描画領域の最新状態を返す。一側面によれば、中継サーバ 10-2 は、複数の装置により使用される複数の通信プロトコル間の違いを、それらの装置の間でインクデータ 200 の通信及び交換を可能にするために吸収する。

【0241】

受信装置 10-3, 10-4、・・・、10-n はそれぞれ図 1 の装置 3 に対応し、通信部 (310 「Rx Tx」)、インクデータ処理部 100R、グラフィック処理部 300、及びアプリケーション部 300-2 を含む。アプリケーション部 300-2 は、インクデータ処理部 100R により処理されるインクデータ 200 の定義及び通信プロトコルを利用するアプリケーション #2 を実行する。受信装置のアプリケーション #2 は、伝送装置 10-1 によって使用されるアプリケーション #1 と同じものであってもよいし、アプリケーション #1 及び #2 の両方がインクデータ 200 の同じ定義と通信プロトコルを共有できる限りにおいて、異なるものであってもよい。

【0242】

図 26 に示す伝送装置 10-1、中継サーバ 10-2、及び複数の受信装置 10-3, 10-4、・・・、10-n の間の区分は図示の容易さのためのものに過ぎず、本発明の実施の形態の適用及び実装によっては、これらの装置の各種機能は、部分的又は全体的に統合されてもよいし、又は、さらに分割及び分配されてもよい。例えば、受信装置 10-3 は、伝送装置 10-1 のインクデータ処理機能 (図 27 に記述される「インクデータ生成」120T 及び「インクデータ整形」140T) を実行するように装備されてもよく、また、伝送装置 10-1 は、受信装置 10-3 のインクデータ処理機能 (図 31 に記述される「インクデータ整形」140R 及び「インクデータ生成」120R) を実行するよう

に装備されてもよい。いくつかの実装では複数の伝送装置が存在し、一方で、いくつかの実装では伝送装置が存在しない。後者のケースでは、例えば、中継サーバ10-2がそのリポジトリ内にインクデータ200を維持し、1以上のクライアントである受信装置10-3、10-4、・・・、10-nに向けてインクデータ200をストリーミングする。

【0243】

図26では、破線の矢印が、複数の装置を介して入力されるインクデータ200が複数のレイヤに重ねて表示される共通描画領域（又はキャンバス）を設定する制御メッセージのようなインクデータ制御メッセージの流れを説明する。インクデータ制御メッセージ名称は、プレフィックス「CTRL_」で始まる。

【0244】

実線の矢印は、共通描画領域に互いに重ねて表示されるために複数の装置を介して入力される実際のインクデータ200を含むインクデータメッセージの流れを説明する。インクデータの名称は、プレフィックス「DATA_INK_」で始まる。

【0245】

図26に示す「SMF」（ストロークメッセージフォーマット）は、伝送フォーマットにおいてパケット内に生成され配置されるメッセージであり、後述する図34に説明される。簡単に説明すると、伝送フォーマット内のインクデータ200は、知覚される遅延のないリアルタイム通信に適している。例えば、上述した「生値型」のポイントオブジェクトが、伝送フォーマットのストロークオブジェクト210を形成するために使用され得る。

【0246】

図26に示す「SFF」（ストロークファイルフォーマット）は、記録フォーマット内に生成され配置されるメッセージであり、後述する図28に説明される。簡単に説明すると、記録フォーマット内のインクデータ200は圧縮されており、記憶装置に適している。例えば、上述した「最適化ポイント型」のポイントオブジェクトが、記録フォーマットのストロークオブジェクト210を形成するために使用され得る。さらなる例では、SMF伝送フォーマットでは色情報がRGBで処理され得るのに対し、SFF記録フォーマットでは色情報は圧縮効率のためにYCCで処理され得る。

【0247】

本発明の様々な実施の形態において、伝送フォーマット（例えばSMF）は、複数の装置の間でリアルタイムにインクデータ200を通信し交換するために使用される。一方、記録フォーマット（例えばSFF）は、共通描画領域（キャンバス）を（初期のアクセスに応じて各装置に送信され得る）リポジトリ内に記憶するために使用される。

【0248】

破線矢印「A. CTRL_JOIN_REQ」は、コラボレーションサービスにより容易された共通描画領域の編集セッションに伝送装置10-1が初めて参加するときに発行されるメッセージである。伝送装置10-1は、共通描画領域に参加する最初の装置であってもよく、既に他の伝送装置によって開始された編集セッションに参加する遅参者であってもよい。破線矢印「A. CTRL_JOIN_REQ」は、受信装置10-3、10-4、・・・、10-nによって共有される描画領域（又はキャンバス）を設定するために伝送装置10-1が発信するメッセージの伝送方向を示す。CTRL_JOIN_REQメッセージは、中継サーバ10-2と交渉され又は共有される伝送装置の環境（例えば、その送信／受信環境を記述する一組のパラメータ）に関する情報を含むメッセージを含み又はそのような情報と関連付けられることとしてもよい。

【0249】

例えば、伝送装置の送信／受信環境を記述するパラメータは、ストローク送信／受信の単位（サイズ）、メッセージ再送信制御設定など（図30Aの524-1）、ペンツールセットを定義する描画パラメータ、座標システムなど（図30Bの524-2）、及び、リソース競合時の優先権やブロックするユーザのリストなど（図30Cの524-3）のユーザポリシーパラメータ、などのような通信パラメータを含む。

10

20

30

40

50

【0250】

破線矢印「A. CTRL_JOIN_REQ」は、CTRL_JOIN_REQメッセージに応じて、中継サーバ10-2が発信する応答メッセージの伝送方向を示す。CTRL_JOIN_REQは、伝送装置10-1が自身のインクデータ200を送信するために必要とする環境データを含む。特に、この環境データは、受信装置10-3、10-4、・・・、10-nの描画領域（又はキャンバス）の最新状態に関する情報を含むことができる。

【0251】

実線矢印「C. DATA_INK_BGNOS」「DATA_INK_MVDOS」・・・・「DATA_INK_ENDOS」（図34参照）は、描画領域（又はキャンバス）を更新するための更新メッセージとして、後に詳述する定義された通信プロトコルに従うタイミング及び伝送フォーマットで中継サーバ10-2に対し伝送装置10-1が送信するインクデータ200の伝送方向を示す。

10

【0252】

実線矢印「D. DATA_INK_BGNOS」及び「DATA_INK_MVDOS」は、中継サーバ10-2によって処理され、受信装置10-3、10-4、・・・、10-nに対してそれぞれの通信線上で送信されるインクデータ200の伝送方向を示す。

【0253】

図27は、図26の伝送装置10-1を機能ブロック図の形で説明する。伝送装置10-1は、（タッチ／ペン）センサ100、入力処理部111、インクデータ処理部100T、アプリケーション部300-1、ネットワーク通信部（Tx, Rx）112T、グラフィック処理部300、ディスプレイ113、及びオペレーティングシステム400-1を含む。

20

【0254】

センサ110は、ユーザの（ペンや指などのポインタを通じた）手書き動作又は操作を検出し、検出した手書き動作を表現する入力データ信号を生成する。例えば、静電センサ、感圧センサ、電磁共鳴（EMR）ベースのセンサが使用され得る。

【0255】

入力処理部111は、典型的には特定のセンサ装置とオペレーティングシステム400-1上で動作するそのドライバとに依存する入力データ信号を受信し、感知された座標位置とポインタ圧力情報などの他の情報とを含む「ペニイベントデータ」に変換し、「入力1」としてそのペニイベントデータを出力する。ペニイベントデータは、なおも手書き動作入力を検出するために使用された特定のセンサ装置に依存している。入力処理部111は、典型的には、センサ110のためのドライバ、又は、センサドライバに対応する処理を行うライブラリにより提供される。例えば、静電センサが使用されるとき、入力処理は掌拒否処理などのような、継続的にエンターされる入力に基づくジェスチャを判定する処理を含み得る。ペニイベントデータがセンサ／装置に依存することから、ペニイベントデータは、センサ110が圧力／傾きの検出機能を有するか否かに応じて、ポインタ圧力情報又はポインタ傾き情報を含むかもしれないし、含まないかもしれない。センサ110及び入力処理部111の構成は図示されたものに限定されず、センサ110及び入力処理部111の全部又は一部は、ペン形の装置のようなデジタル文具装置の形態で提供されてもよい。

30

40

【0256】

インクデータ処理部100Tは、インクデータ生成部120T及びインクデータ整形部140Tを含む。インクデータ処理部100Tは、第1の実施の形態の図5における100Tに対応する。インクデータ処理部100Tは、センサ／装置に依存するペニイベントデータ（「入力1」）を、様々な装置の様々なアプリケーションによって使用され共用され得るインクデータ200に変換することを担当する。

【0257】

インクデータ生成部120Tは、センサ／装置に依存するペニイベントデータを引き出

50

し又は受信し、それを非装置依存型のインクデータ200に変換する。インクデータ200は、本物のペンを使って紙の上に適用される（に描かれる、をよごす、に堆積される）インクのデジタル表現である。インクデータ生成部120Tは、図7のストロークオブジェクト操作部122に対応する。インクデータ生成部120Tは、規定の時間単位（例えば5秒）や規定のデータサイズ単位のような規定の単位Tごとにインクデータ（図7のメモリ770に記憶されるポイントオブジェクト $p_0 \sim p_{n-1}$ ）を引き出し、手書き動作により入力されたトレースを表すストロークオブジェクト210、又は、ストロークオブジェクト210の断片（部分）を生成する。

【0258】

インクデータ生成部120Tは、センサ110からペニイベントデータ（「入力1」）を受信し、また、アプリケーション部300-1（例えば、リアルタイムコラボレーションアプリケーション）又はオペレーティングシステム（400-1）からコンテキスト情報（「入力2」）を受信する。

【0259】

コンテキスト情報（「入力2」）は、ストロークの最初の部分が描かれた時点（すなわち「ペンダウン」の時点）における、ペニイベントデータのコンテキスト又は環境にかかる情報である。コンテキスト情報は、典型的にはセンサ110によりペニイベントデータの生成に先立ち、アプリケーション部300-1によって設定される。例えば、コンテキスト情報は、使用されるペン先のタイプ（例えば、ブラシ、クレヨン、鉛筆）、ストローク／トレースの色（赤、緑、青）、ペニストロークの透明度（又は不透明度）の値（アルファ）、ストローク／トレースの幅などを含むことができる。インクデータ生成部120Tは、ペニイベントデータの開始のタイミングで（図32のS605で。後述）コンテキスト情報（「入力2」）に基づき、ディスプレイ上にストロークオブジェクト210を描画（レンダリング）するために使用される描画スタイルオブジェクトを生成する。

【0260】

コンテキスト情報（「入力2」）はまた、著者情報、ペンID、日時情報、位置情報などのようなペニイベントデータについての非描画関連情報を含む。インクデータ生成部120Tは、そのような非描画関連コンテキスト情報に基づき、メタデータオブジェクト250を生成する。

【0261】

インクデータ生成部120Tは追加で、アプリケーション部300-1から操作情報（「入力3」）を受信する。入力3は、センサ110からの次の入力が通常のストロークオブジェクト210を定義するものではなく、既存のストロークオブジェクト210に対する操作オペレーション（例えば、スライス、削除、コピー、拡大など）を具体化し実行する操作オブジェクト270を定義するものであることを特定する。入力3が受信されると、1以上の既存のストロークオブジェクト $\#0 \sim \#i$ に対して、次のストロークオブジェクト $\#j$ が操作オブジェクト270として形成され、その操作オペレーションが適用される。

【0262】

インクデータ整形部140Tは、インクデータ通信部144T及び記録フォーマットデータ処理部142Tを含む。一般的に、インクデータ整形部140Tは、インクデータ生成部120Tによって規定の単位Tごとに生成されるストロークの断片化データを整形する（例えば、伝送パケット内に配置する）。インクデータ整形部140Tはまた、インクデータ生成部120Tによって生成された描画スタイルオブジェクト230、メタデータオブジェクト250、及び操作オブジェクト270の整形も行う。インクデータ整形部140Tは、アプリケーション部300-1から受信されるフォーマット選択情報（Format-Select）に従って、様々なオブジェクトと、各ストロークオブジェクト210の断片化データとをメッセージ及び伝送パケット内にフォーマットする。

【0263】

インクデータ通信部144Tは、アプリケーション部300-1から受信されるフォー

マット選択情報 (F m t _ S e l) が出力フォーマットとしての伝送フォーマット (例えば S M F) の使用を特定しているとき、以下の機能を実行する。

【0264】

1) インクデータ生成部 120 T によって生成されたストロークオブジェクト 210 を規定の単位 T ごとの断片化 (又は完全な) データとして受け付け、暗示的に定義された、又は、明示的に交渉されたパラメータに従って、様々なタイプのデータメッセージ (図 34 に示すように、プレフィックス「D A T A _ I N K _」で始まるメッセージ) を生成する。

【0265】

2) インクデータ生成部 120 T によって生成された描画スタイルオブジェクト 230 を受け付け、「最初」のデータメッセージ (図 34 の「D A T A _ I N K _ B G N O S」) にその描画スタイルオブジェクト 230 (図 34 の「F 104」) を加える。

【0266】

3) ストロークオブジェクト 210 のユニークなストローク I D を決定し、ストロークオブジェクト 210 の断片化データを含むすべてのデータメッセージにそのストローク I D (図 34 の「F 103」) を加える。

【0267】

4) 応答メッセージ (図 36 A の「D A T A _ I N K _ N A C K」又は図 37 の「D A T A _ I N K _ R E J O S」) の受信に応じて、交渉されたパラメータ (例えば、図 30 A の 524 _ 1 f、524 _ 1 a とともにあってもよいし、又は、524 _ 1 a なしで 524 _ 1 f のみがあってもよい) に従い、再送処理 (図 35 A、図 35 B、図 36 A 参照) 又は無視する処理 (図 37) を適応的に行う。

【0268】

記録フォーマットデータ処理部 142 T は、アプリケーション部 300-1 により生成されたインクデータ 200 を、伝送フォーマットとは異なる記録フォーマット (例えば S F F) の形式に処理する。例えば、記録フォーマット内のインクデータ 200 は、受信装置 10-3、10-4、・・・、10-n の複数のユーザによって共有 (アクセス) される共通描画領域 (カンバス) の現在状態を示すために、リアルタイムコラボレーションアプリケーションへの初期アクセスに応じて、伝送装置 10-1 から中継サーバ 10-2 にアップロードされ得る。これに対し、複数の装置間でリアルタイムにインクデータ 200 を通信するため、記録フォーマットに代えて伝送フォーマットが使用されてもよい。

【0269】

図 28 は、本発明の実施の形態で使用される記録フォーマットのサンプルを説明している。この記録フォーマットは、含まれるデータ/省略されるデータのタイプ、データの順序及び冗長性の点で、(図 34 の) 伝送フォーマットとは異なっている。

【0270】

含まれるデータ/省略されるデータのタイプ

例えば、既存のストロークオブジェクト 210 を修正するために操作オブジェクト 270 が生成されるとき、1 以上のコンピュータに存在するインクデータ 200 を変形するための操作オブジェクト 270 (例えば、第 1 の実施の形態を参照して上述したスライスオブジェクト 274) が伝送フォーマットで生成され、送信される。これに対し、記録フォーマットでは、操作 (更新) が完了した後の状態のみを記録するのに十分であり、したがって、操作オブジェクト 270 自身を維持する必要はない。したがって、図 28 の記録フォーマット例は、操作オブジェクト 270 を含まない。

【0271】

データの順序及び冗長性

メタデータオブジェクト 250 は、著者情報、ペン I D などのようなストロークオブジェクト 210 についての非描画関連情報を含む。非描画関連情報はしばしば、同時に入力されたいくつかのストロークオブジェクト 1~5 にとって同一である。例えば、同じペンを使用する同じ著者は、複数のストロークオブジェクト 1~5 をこの順で生成する。

10

20

30

40

50

【0272】

記録フォーマットでは、記録フォーマットで書かれたデータを整形する時点で全体のデータ内容が既知となっていることから、冗長性が低減され得る。図示した例では、同じ値の著者をストロークオブジェクト1～5のすべてに適用することができ、したがって、同じ値が5回繰り返される必要はない。描画スタイルオブジェクト230に含まれる描画関連情報もまた、例えば、同じタイプのペン先形状（例えばブラシ）が複数のストロークオブジェクトを描画するために使用されるというように、いくつかのケースで冗長性を含み得る。したがって、他の例としては、同じ値のペン先形状をストロークオブジェクト1, 3, 5に適用しており、この値が記録フォーマット内で3回繰り返される必要はなくなっている。同様に、同じ値のペン先形状がストロークオブジェクト2, 4に適用される場合

10

、その値は記録フォーマット内で繰り返される必要はない。この点について、記録フォーマット内においてストロークオブジェクトの時間的な順番（すなわち、それらが入力された順）を維持することは重要ではない。また、記録フォーマット内に注入される時点で各ストロークオブジェクトは完全に完成しているので、1つのストロークオブジェクトを形成するポイントオブジェクトの合計数を、ストロークオブジェクト自身の中にも含めることができる。

【0273】

これに対し、伝送フォーマットでは、もしリアルタイム通信が重要であるならば、ストロークオブジェクトの時間的な順番を再編することは困難である。リアルタイム通信では一般に、ストロークオブジェクト1～5が入力され生成された時間的な順番で、ストロークオブジェクト1～5に関する情報を送信することが必要である。また、単一のストロークオブジェクトの断片化データを送信するとき、そのストロークオブジェクト内にいくつかのポイントオブジェクトが含まれることになるのかを予め知ることは不可能である。したがって、ストロークオブジェクトを完結させる最後の断片化データについては、そのことを示す必要がある（最後の断片化データは、ストロークオブジェクト内に含まれるポイントオブジェクトの合計数に関する情報を含むこととしてもよい）。

20

【0274】

本発明の実施の形態によれば、同じ量の情報は、伝送フォーマットより記録フォーマットの方が少ないバイト数で含まれ得る。伝送フォーマットは、知覚可能な時間遅延のないリアルタイム通信を実現する目的のために、冗長性と厳格な時間的順番を有する必要があるからである。したがって、省記憶スペース化の目的のために、中継サーバ10-2のリポジトリに記憶される最新の描画領域情報は好ましくは記録フォーマットで形成され、それはその後、中継サーバ10-2に接続される様々な装置により、より少ない送信リソース量を用いてアクセスされ、引き出され得る。

30

【0275】

図27を再度参照すると、伝送装置10-1のネットワーク通信部112T（Tx, Rx）は、インクデータ処理部100Tによって生成されたメッセージをペイロードとして含む通信パケット（図34参照）を生成し、そのパケットを、メディアに接続されたネットワークインターフェイス（イーサネット（登録商標）など）を介して出力する。

40

【0276】

様々な通信プロトコルが、必要に応じ、通信環境の点から見て暗黙の又は（交渉を介した）明示の定義に基づいて使用され得る。例えば、ビルトインの再送メカニズムを含むTCP、HTTPオーバーTCP、SSLのようなプロトコルを使用してもよいし（図35A）、それ自身は再送メカニズムを提供しないUDP（又はRTP/UDP）のようなプロトコルを使用してもよい（図35B）。さらに、UDPが使用される場合には、メッセージ再送メカニズムを採用してもよいし（図36A）、リアルタイム型アプリケーションの時間的要求に従う目的のためにメッセージ再送メカニズムを省略してもよい（図36B）。また、制御メッセージとデータメッセージとは、それぞれ異なる通信プロトコルを用いることも可能である。例えば、破線矢印で示される制御メッセージの送信には再送メ

50

カニズムを有するプロトコルを採用し、一方で、実線矢印で示されるデータメッセージの送信には再送メカニズムを有しないプロトコルを採用することも可能である。

【0277】

アプリケーション部300-1は、本発明の実施の形態によるインクデータ送信方法を使用するアプリケーションを提供する。アプリケーション部300-1は、共通描画領域（又はキャンバス）の最新状態を管理する中継サーバ10-2（具体的には図29の520）に対し、ネットワーク通信部112T（Tx）を介して、CTRL_JOIN_REQなどを発行する。アプリケーション部300-1は、入力装置110を使用してその上で描画を実行する描画領域（キャンバス）を決定する。

【0278】

アプリケーション処理部300-1は、以下の情報を処理し、伝送装置10-1のインクデータ処理部100Tに対して提供する。

- ・色情報、ペン先形状、著者情報、生成日時などのコンテキスト情報（入力2）、
- ・操作オブジェクトを構成する入来入力を特定する操作情報（入力3）、及び／又は
- ・伝送フォーマットと記録フォーマットのいずれが選択されるべきかを指定するフォーマット選択情報（Fmt_Sel）。

【0279】

アプリケーション処理部300-1は、インクデータ生成部120Tによって生成されたインクデータ200を受信し、リモートに位置する受信装置（10-3、10-4、
・
・
・、10-n）と、伝送装置自身のローカルディスプレイ（もしあれば）の両方に出力する能力を有する。

【0280】

グラフィック処理部300は、インクデータ200に基づいてピクセルデータを生成する。グラフィック処理部300は、ローカルディスプレイ上にストロークオブジェクトを直ちに描画（レンダリング）する能力を有し、これは、複数の装置を介して入力されたインクデータ200が異なるレイヤとして重ねられた共通描画領域の状態を表し得る。

【0281】

本発明の実施の形態によるインクデータ伝送装置10-1は、使用のために、描画領域（キャンバス）の最新の（更新された）状態を決定し、規定の単位T（例えば5ミリ秒）ごとに、ストロークオブジェクト210の断片化（又は完全な）データを生成するために、ペンイベントデータを継続的に処理する。インクデータ伝送装置10-1はさらに、コンテキスト情報（入力2）の各部分を選択することにより、関連する描画スタイルオブジェクト230及びメタデータオブジェクト250を生成し、入力3に基づいて操作オブジェクトを生成し、生成されたオブジェクトを、中継サーバ10-2及び受信装置10-3、10-4、
・
・
・、10-nへの送信のために伝送フォーマットに整形する。規定の単位Tは、コラボレーションアプリケーションからの要求に基づいて適応的に決定され得る。

【0282】

図29は、中継（又はストリーミング）サーバ10-2のブロック図である。中継サーバ10-2は、1以上の伝送クライアント装置から1以上の受信クライアント装置10-3、10-4、
・
・
・、10-nに受信される断片化データを中継することにより、リアルタイムコラボレーションサービスを提供する。中継サーバ10-2の機能は、単一の装置内に含まれることとしてもよいし、ネットワークによってリンクされた複数のサーバに分散されることとしてもよい。

【0283】

中継サーバ10-2は、メッセージ受信部510、描画領域管理部520、インクデータ中継部540、及びメッセージ送信部511を含む。

【0284】

メッセージ受信部510は、制御メッセージとデータメッセージとを、伝送装置10-1から別々に受信する。これらのメッセージは、あるプロトコル（RTP/HTTP/HTTPS（SSL）/TCP/UDP/Web socketなど）に従う通信パケットの

10

20

30

40

50

形で、かつ、予め定義されるか、或いは、装置が初めて共通描画領域に参加したときに装置間で行われるオンデマンド交渉に基づいて定義されてよいメッセージタイプ（断片化、再送信、最大遅延など）で、伝送装置 10-1 と中継サーバ 10-2 の間を伝送される。図示するように、メッセージ受信部 510 は、制御メッセージとデータメッセージを区別するために複数の受信ポートを採用することができる。ただし、制御メッセージとデータメッセージを区別するために他の方法も採用され得る。例えば、制御メッセージを処理する分離装置（サーバ）が、データメッセージを処理する中継サーバ 10-2 とは別に用意され得る。また、共通の受信ソケットバッファから取得可能なメッセージヘッダが 2 つのタイプのメッセージを区別するために使用されてもよい。

【0285】

描画領域管理部 520 は、複数の装置から入力され、中継サーバ 10-2 を通じて交換されたインクデータ 200 が異なるレイヤとして互いに重ねられる描画領域の状態を管理する。描画領域管理部 520 は、サービス管理部 522 及び描画領域情報管理部 524 を含む。

【0286】

サービス管理部 522 は、本発明の実施の形態によるインクデータ中継方法を用いるサービスを管理する。サービス管理部 522 は、例えば会計機能、新たなユーザの認証機能、共通描画領域への契約ベースの視聴アクセスを提供し、共通描画領域に対してインクデータ 200 を入力するために各ユーザを許可し或いは許可しない機能など、すなわち所謂 AAA（会計(Accounting)、認証(Authenticating)、許可(Authorizing)) 機能を実行するために、外部サーバ（図示せず）と協力する。

【0287】

描画領域情報管理部 524 は、インクデータ中継方法の動作を制御するために使用される描画領域情報を管理する。描画領域情報は一般に、3 つのタイプの情報（図 30A の 524__1、図 30B の 524__2、図 30C の 524__3）を含む。描画領域情報管理部 524 は、これら 3 つのタイプの情報を管理し、これら 3 つのタイプの情報の更新に基づいて、各描画領域（キャンバス）に接続されている 1 以上のユーザを列挙するユーザ接続リスト 541 の維持、パージを行う。

【0288】

図 30A は、インクデータ 200 の送受信に関連する一組の通信パラメータ 524__1 を示している。これらの通信パラメータは、例えばアプリケーションが起動する時点で通信している装置の間で交換される。通信パラメータ 524__1 は、以下を含み得る。

【0289】

(524__1 a) パケット再送信パラメータは、インクデータ 200 を運ぶために使用される通信パケット（又はプロトコルスタック）に再送メカニズムがビルトインされているか否かを定義する。例えば、送信に（再送信サポートのある）TCP を用いる一方で受信に（再送信サポートのない）UDP を用いる場合、送信には TRUE が設定され、受信には FALSE が設定される。これは例えば、強固かつ信頼性の高い伝送装置 10-1 から中継装置 10-2 へのインクデータの送信を実現する一方、同時に、リアルタイムで知覚可能な遅延のない、中継サーバ 10-2 から複数の受信装置 10-3, 10-4、・・・、10-n へのインクデータ 200 のストリーミングを提供する。

【0290】

(524__1 b) MTU (Maximum Transmission Unit)、MSS (Maximum Segment Size) パラメータは、伝送装置が接続されるメディア（例えばイーサネット（登録商標））のタイプに依存する MTU 及び MSS を定義する（例えば 1460 バイト）。

【0291】

(524__1 c) ストロークデータ断片化可否パラメータは、ストロークオブジェクト 210 を、それぞれが全体としてのストロークオブジェクト 210 より小さい複数の断片に断片化するか否かを設定する。

【0292】

10

20

30

40

50

(524__1d) 最大Rx遅延パラメータは、受信遅延の許される最大値を、例えばミリ秒 (m s e c) で設定する。

【0293】

(524__1e) メッセージ暗号化スキームパラメータは、暗号化が使用されるか否か、されたとしたら何の暗号化／復号アルゴリズムが使用されるのか、を定義する。

【0294】

(524__1f) メッセージ再送パラメータは、通信レイヤに再送メカニズムを含まないUDPのようなプロトコルのために、メッセージレイヤで再送信を実行するか否かを定義する。例えば、このパラメータは、メッセージレイヤでの再送信を使用すること (図36A) と、メッセージレイヤでの再送信を使用しないこと (図36B) とを切り替えるために使用される。

10

【0295】

(524__1g) 音声同期可否パラメータは、音声とストロークデータが同期して再生されるか否かを定義するもので、断片の形態で又は完全な形態 (ストローク単位で) でインクデータ200を送信するか否かを決定するために使用され得る。例えば、音声とストロークデータが同期して再生されるとき、ストロークデータは、各断片が対応する音声データと同期して再生され得るよう、それぞれにタイプスタンプが付された各断片に断片化され得る。

【0296】

(524__1z) 他パラメータ設定識別子は、通信パラメータの所定の組を定義する。

20

【0297】

図30Bは、インクデータ200の描画 (レンダリング) に関連する一組の描画パラメータ524__2を示しており、以下を含み得る。

【0298】

(524__2a) 描画領域IDパラメータは、ユーザ装置が参加し得る共通描画領域を設定する。図示した例では、図30Bは、共通描画領域 (又はキャンバス) IDである「#123」を示している。いくつかの実施の形態では、中継サーバ10-2は、それぞれの中で共同編集が行われており、ユーザ装置が参加のために1つを選択できる複数のキャンバスの複数のIDをユーザに提示することとしてもよい。他の実施の形態では、ユーザ装置が新しい描画領域を開始するとき、ユニークな描画領域IDが新しく開始された描画領域に割り当てられることとしてもよい。

30

【0299】

(524__2b) ユーザローカルなキャンバスのオフセット、回転、スケールのパラメータは、共通描画領域のグローバルな座標システムと、共通描画領域の編集セッションに参加するために使用されるユーザ装置のローカルな座標システムとの間の関係を定義する。例えば、これは2人の異なるユーザが、2つの異なる角度から、それぞれ描画中のインクデータの上部及び下部を編集することを可能にする。

【0300】

ペンツールセットIDは、連帯して描画領域上にペinstロークがどのように現れるかを定義するペンパラメータの集合 (ペン先形状、色、ストローク幅、インクの透明度／不透明度など) に割り当てられるIDである。複数のIDが複数の集合のために予め定義されており、ユーザは、共通描画領域上の共同編集セッションに参加するとき、その中から1つを選択することができる。

40

【0301】

図30Cは、ストロークデータの優先度、フィルタリング、サービス品質 (QoS) を含む共通描画領域に対するユーザアクセスを管理するポリシーに関連する一組のユーザポリシーパラメータ524__3を示しており、以下を含み得る。

【0302】

(524__3a) ユーザアカウント優先度パラメータは、インクデータ伝送サービス内でストロークデータに与える優先度を設定する。図示した例では、このパラメータは「H

50

i g h」に設定されており、これは例えば、中継サーバ10-2のリソース調停（又はQoS）制御544が、「H i g h」優先度のストロークデータを、優先度が「L o w」に設定された他のストロークデータに対して優先的に処理及び送信することを意味する。

【0303】

（524__3b）ブロックユーザリストは、あるユーザがブロックしたいと思う他のユーザ、すなわちそのユーザからのインクデータ200を受信したくないと願うユーザを含む。このリストは、ユーザがインクデータ200を送信したいと願わないユーザも含み得る。そのユーザからの受信又はそのユーザへの送信を否定するためにユーザをリスト化する代わりに、そのユーザからの受信又はそのユーザへの送信を受け入れるためにユーザをリスト化することも可能である。

【0304】

インクデータ中継部540は、データメッセージが追加（描画）される描画領域に現在接続されているすべての装置をリスト化してなる接続リスト541を参照して、伝送装置10-1から受信したデータメッセージを、1以上の他の受信装置に中継する。接続リスト541はまた、接続リスト541内に列挙された各装置と通信するためにどの通信、描画、ユーザポリシーのプロトコルが使用されるべきかをリスト化している。様々な実施の形態において、インクデータ中継部540は、パケット単位ではなくメッセージ単位で、しかし「1つの流れ」として、ストロークオブジェクトメッセージを処理する（転送する、落とす）。

【0305】

インクデータ中継部540は、新ストロークフロー検出部542、リソース調停（又はQoS）制御544、フィードバック送信部546（DATA__T o S e l f）、及びオールキャスト送信部548（DATA__T o O t h e r s）を含む。

【0306】

新ストロークフロー検出部542は、受信メッセージのヘッダをチェックし、そのメッセージが新しいストローク（「DATA__I N K__B G N O S」）を含むか否か、又、そのメッセージが、中継処理が既に始まっているものと同じ（現在の）ストロークオブジェクト210の断片化データを含むか否かを判定する。

【0307】

リソース調停（又はQoS）制御544は、各ストロークオブジェクト210の始端を参照することにより、そしてもし必要なら各ストロークオブジェクト210の終端も参照することにより、送信リソースを制御する。新しいストローク（B G N O S）の始端を処理するとき、リソース調停（又はQoS）制御544は、様々な基準に基づき、描画領域（キャンバス）内に新しいストロークを受け入れるか否かを決定する。例えば、もし伝送装置10-1から新たに受信されたストロークオブジェクト210のストロークIDが既に描画領域内で使用されているストロークIDと同一であるならば、リソース調停（又はQoS）制御544は、同じストロークIDを有する新たに受信されたストロークオブジェクト210を拒絶することができる。他の例として、もし特定の時点でストロークオブジェクト210を転送し又は処理するためにネットワークリソースが不十分であると判明した場合、リソース調停（又はQoS）制御544は、ネットワークリソースの状況が改善するまで、新たに受信されるストロークオブジェクトの入力を拒絶する。これは、十分なネットワーク又はコンピュータのリソースがないことによりストロークオブジェクト210を完成させる前に入力処理を中止するためだけに、ユーザが新しいストロークオブジェクト210の入力を開始するという状況が発生することを防止する。さらなる例として、もし特定の装置（ユーザ）が一時的に更新権を否定されるのであれば、リソース調停（又はQoS）制御544は、その特定の装置によって生成されたすべてのストロークオブジェクトを拒絶する。例えば、リソースは、「H i g h」優先度（図30Cの524__3）を有するユーザから伝送されたストロークデータを処理することに、「L o w」優先度を有するユーザから伝送されたストロークデータに対して先取権のある状態で割り当てられる。リソース調停（又はQoS）制御544は、いかなる装置から受信される新たなスト

10

20

30

40

50

ロックをも拒絶すると決定したとき、フィードバック送信部 546 に「REJECT」を送信する。リソース調停（又は QoS）制御 544 は、ある装置から受信される新たなストロークを受け入れると決定したとき、他のすべての装置（その新たなストロークを送信した装置を除く）に対し、オールキャスト送信部 548 を介して、その新たなストロークを送信（転送）する。

【0308】

本発明の様々な実施の形態によれば、各ストロークオブジェクト 210 が断片化データに断片化され、順番に送信及び中継されるので、リモートに位置する装置は、知覚可能な時間遅延なしでリアルタイムに入力されるのと同様に、ストロークオブジェクト 210 を共有することができる。同時に、リソース調停（又は QoS）制御 544 は、複数のユーザの間のいかなる衝突をも解決し、新たなストロークオブジェクト 210 を入力する権利を認められたすべてのユーザに対して十分なリソースが提供されること（例えば、最初のユーザが入力している間、他のユーザは入力することを一時的に妨げられるとしても）を確実にするトラフィック制御機能を実行する。

【0309】

フィードバック送信部 546（DATA_To_Self）は、中継サーバ 10-2 に対してメッセージを送信した伝送装置 10-1 に対してのみ応答を送り返す。例えば、リソース調停（又は QoS）制御 544 が伝送装置 10-1 による新たなストロークオブジェクト 210 の入力を許可しないと決定したとき、フィードバック送信部 546 は、その伝送装置 10-1 に対してのみ拒否メッセージ（図 37 参照）を送信する。

【0310】

オールキャスト送信部 548（DATA_To_Others）は、描画領域に現在接続されているすべての装置（接続リスト 541 に含まれるもの）に対してメッセージを送信する。ただし、中継サーバ 10-2 に対してそのメッセージを送信した伝送装置 10-1 を除く。図示した例では、伝送装置 10-1 からメッセージが受信され、中継可能であると判定されたとき、オールキャスト送信部 548 は、すべての受信装置 10-3, 10-4、・・・、10-n に向けてメッセージを送信する。

【0311】

中継サーバ 10-2 は、このようにして、一の伝送装置からインクデータ 200 を受信し、全体としてのネットワークリソースの効率的な利用を最適に制御しつつ、それを 1 以上の受信装置に対して選択的に中継する。

【0312】

図 31 は、図 26 のインクデータ通信システムのインクデータ受信装置 10-3 のブロック図である。受信装置 10-3 は、ネットワーク通信部 310（Rx, Tx）、インクデータ処理部 100R、アプリケーション部 300-2、グラフィック処理部 300、オペレーティングシステム 400-2 を含む。

【0313】

オペレーティングシステム 400-2 上で動作するアプリケーション部 300-2 は、本発明の実施の形態によるインクデータ受信方法を利用するアプリケーションを含む。そのアプリケーション又はオペレーティングシステムの種類は、伝送装置 10-1 のアプリケーション部 300-1 のアプリケーション（又はオペレーティングシステム）と同じであってもよいし、両方のアプリケーションが共通描画領域を通じて互いにインクデータ 200 の入力、送信、及び受信を許容する限りにおいて異なってもよい。これは、共通に定義される（標準化される）インクデータ 200 が、異なるアプリケーション及びオペレーティングシステムの間で、特定のプラットフォームから独立して、送信及び受信され得るからである。受信装置 10-3 が共通描画領域の編集セッションに参加する通信初期化のタイミングで、アプリケーション部 300-2 は、他のユーザがおそらく異なるレイヤとしてそれぞれのインクデータ 200 を重ねている描画領域（キャンバス）の最新状態をダウンロードする。

【0314】

アプリケーション部 300-2 は、それが生成したインクデータ 200 を、中継サーバ 10-2 の描画領域に対してアップロードすることができる。受信装置 10-3 によってローカルに生成されたインクデータ 200 は、共同編集セッションにおそらく先立って、記録フォーマットとされ得る。一方、共同編集セッションの間にリアルタイムで生成されたインクデータ 200 は、伝送フォーマットとされ得る。記録フォーマット及び伝送フォーマットどちらのインクデータ 200 であっても、中継サーバ 10-2 に対してアップロードされ得る。この接続の中で、アプリケーション部 300-2 は、インクデータ生成部 120R によって生成されたインクデータ 200 を出力し、描画領域内に重ねるようグラフィック処理部 300 に命令する。

【0315】

ネットワーク通信部 310 (Tx, Rx) は、伝送装置 10-1 のネットワーク通信部 112T (Tx, Rx) に対応する。ネットワーク通信部 310 は、ネットワークインターフェイスを介して通信パケット (Packet) を受信し、そのパケットのペイロード部分からメッセージを取り出す。受信装置 10-2 によって使用される受信プロトコルは、伝送装置 10-1 によって使用される通信プロトコルとは異なっているもよい。例えば、伝送装置 10-1 は、中継サーバ 10-2 にメッセージを送信するために TCP、HTTP オーバー TCP、HTTPS オーバー SSL のような再送メカニズムを含むプロトコルを採用し得る一方、中継サーバ 10-2 は、受信装置 10-2 にメッセージを送信するために、UDP、RTP オーバー UDP のような、それほど信頼できないけれどもストリーミングに適したプロトコルを採用することとしてもよい。どの通信プロトコルを使用するかは、通信する装置の間で通信開始時に交渉を通じて決定されればよい。

【0316】

インクデータ処理部 100R は、インクデータ整形部 140R 及びインクデータ生成部 120R を含み、伝送フォーマット又は記録フォーマットのいずれかで整形されたインクデータからインクデータ 200 を取り出す。インクデータ処理部 100R は、伝送装置 10-1 のインクデータ処理部 100T に対応する。

【0317】

インクデータ整形部 140R は、インクデータ通信部 144R 及び記録フォーマット操作部 142R を含む。インクデータ通信部 144R は、伝送フォーマットの形式で更新されたインクデータ 200 を処理する。具体的には、インクデータ通信部 144R は各データメッセージ内の情報 (例えばストローク ID) を取り出し、送信側によって規定の単位 T で分割されたストロークデータ 210 の断片である断片化データを出力する。インクデータ通信部 144R はまた、ストロークオブジェクト 210 のために、1 番目のメッセージ、すなわち「DATA__INK__BGNO S」メッセージに含まれる描画スタイルオブジェクト 230 を取り出す。描画スタイルオブジェクト 230 は、関連するストロークオブジェクトをレンダリング (描画) するために必要な情報を含む。

【0318】

記録フォーマット操作部 142R は、最新の描画領域を再構築するために、中継サーバ 10-2 のリポジトリに記憶 (アーカイブ) されるものとして、記録フォーマットで描画領域情報を受信し処理する。

【0319】

アプリケーション 300-2 は、出力/入力されるインクデータのフォーマットに従い、インクデータ通信部 144R から又は記録フォーマット操作部 142R から選択的にインクデータ 200 を受信するよう、インクデータ生成部 120R を制御する。

【0320】

例えば、アプリケーションが初めて共通描画領域における共同編集セッションに参加するとき、記録フォーマット (SFF) で書かれたファイル内の最新の描画領域情報を引き出すために、「CTRL__JOIN__REQ」要求が発行される。引き出されたの最新の描画領域のインクデータ 200 は記録フォーマットで書かれており、そのようなものとして、記録フォーマット操作部 142R で処理される。続けて引き出されるインクデータ 2

10

20

30

40

50

00は、図34に示した「DATA__INK__BGNO S」及び他のメッセージ(DAT A __INK*)のように、伝送フォーマット(SMF)であり得、そのようなものとして、インクデータ通信部144Rにおいて一度に1メッセージ(又は1ストロークを表す複数のメッセージ)が処理される。

【0321】

インクデータ200のリアルタイム処理を実現するため、インクデータ処理部144Rは、ストロークオブジェクト210の断片化データを、ストロークオブジェクト210を形成するすべてのポイントオブジェクト(又は断片化データ)を引き出すのを待つのではなく、それらが受信されたときにグラフィック処理部300に対して出力する。

【0322】

受信装置10-3のインクデータ生成部120Rは、伝送装置10-1のインクデータ生成部120Tの送信処理に対応する受信処理を実行する。

【0323】

それぞれ複数のポイントオブジェクト、描画スタイルオブジェクト230、メタデータオブジェクト250、及び、ストロークオブジェクトに関連する操作オブジェクトからなる複数のストロークオブジェクトからなるインクデータ200は、すべて、送信側で既に生成されている。したがって、受信側で新たに生成される必要はない。むしろ受信側では、インクデータ生成部120Rが、そのアプリケーション#2によって利用可能な形態でインクデータ200を記憶し、更新する。例えば、インクデータ生成部120Rは受信した第1のデータ型(例えば整数型)のインクデータ200を、アプリケーション#2によ

10

20

【0324】

受信装置10-3のグラフィック処理部300は、伝送部10-1のグラフィック処理部300に類似する。グラフィック処理部300は、受信したインクデータ200に基づいてピクセルデータを生成する処理を行い、受信装置10-3のディスプレイスクリーン113上にストロークオブジェクトをディスプレイする。

【0325】

図32は、本発明の実施の形態による、伝送装置10-1、中継サーバ10-2、及び受信装置10-3の間で行われるインクデータ通信を説明するシーケンス図である。

30

【0326】

図32において、伝送装置10-1、中継サーバ10-2、及び受信装置10-3の機能及びセクションは、上記図27、図29、図31で使用されたものと同じ参照番号により識別される。本シーケンスは、既に開始されているリアルタイムコラボレーションセッションに伝送装置10-1が参加し、複数のユーザにより入力されたインクデータ200が互いに重ねられ得るケースを説明する。

【0327】

<アプリケーション部300-1を開始>

ステップS601では、伝送装置10-1が、本発明の実施の形態によるインクデータ通信方法を利用するアプリケーション部300-1を活性化する。アプリケーション部300-1の活性化は、インクデータ処理部100内のインクデータ生成部120T及びインクデータ整形部140Tの活性化のきっかけとなる。

40

【0328】

初めに、伝送部10-1のアプリケーション部300-1が、中継サーバ10-2の画像領域管理部520内に保持されている複数の描画領域のうちの1つをターゲット描画領域(524__2a)として特定する制御メッセージ(CTRL__JOIN__REQ。パラメータ524__1, 524__2, 524__3を有していても有していなくてもよい)を発行する。中継サーバ10-2の描画領域管理部520は、特定された描画領域の最新状態を、高度に圧縮されていてもよい記録フォーマット(SFF)で伝送装置10-1に返す

50

(CTRL_JOIN_REP)。その後、アプリケーション部300-1は、記録フォーマットで書かれたインクデータ200に基づいて描画領域の最新状態を再構築するよう、記録フォーマット操作部142Tを呼び出す。伝送装置10-1はそれにより、受信装置10-3及び10-4のような他の装置が互いに重ねられるインクデータ200の描画(入力)を既に開始しているかもしれない描画領域(又はインクデータの現在状態)の最新状態をローカルに再構築する(又は初期構築する)。この時点で、伝送装置10-1は入力待ち状態に入る。

【0329】

＜インクデータ伝送処理＞

伝送装置10-1は、トリガーとして「ペンイベントデータ入力開始」の検出を用いることにより、インクデータ伝送処理を実行する。ステップS605では、伝送装置10-1の入力処理部111がストロークの開始点の入力を検出する。図32では、ステップS605からステップS609までのハッチングされた期間「d」の間、ペンイベントデータが継続的に、インクデータ処理部100Tに入力される。その間、1つのストロークがストローク開始点で始まり、ストローク終了点で終わるように描かれる(以下、「ストローク単位」と称する)。この時間(期間d)の間、入力処理部111は単位Tごとに継続的に、例えば5ミリ秒ごとに、インクデータ生成部120Tに対してペンイベントデータを出力する。

【0330】

インクデータ処理部100Tは、ペンイベントデータ入力の開始をトリガとして用い、アプリケーション部300-1から受信されるコンテキスト情報(入力2)に基づいて描画スタイルオブジェクト230を生成する処理を開始する。具体的には、インクデータ処理部100は、アプリケーション部300-1から受信されるペンイベントデータに関するコンテキスト情報(入力2)に基づいて、共通描画領域(キャンバス)上にストロークオブジェクト210を描画(レンダリング)するために使用される情報を含む描画スタイルオブジェクト230を生成する。インクデータ処理部100は、図画をレンダリングするため、受信側にとって必要であるコンテキストデータの一部分を選択的に取り出すことにより、描画スタイルオブジェクト230を生成する。描画スタイルオブジェクト230は、ストロークの色(赤、緑、青)、ペン先タイプ(ブラシ、クレヨン、鉛筆)、透明度又は不透明度の値(アルファ)、ストロークオブジェクト210内で透明度又は不透明度を変化させることが許容されるか、ストローク(トレース)の幅(例えば0.2~6.0ポイント)、ストロークオブジェクト210内でストローク(トレース)の幅を変化させることが許容されるか、ラスター化の方法、ストロークのテクスチャのタイプなどのような情報を含む。

【0331】

描画スタイルオブジェクト230の生成は、ペンイベントデータの入力に基づくストロークオブジェクト210の開始点の入力の検出によって、すなわち図32のS605で、トリガされる。したがって、描画動作を開始する前にユーザが繰り返しペンの色を変更したとしても、古い(時代遅れの)色情報は描画スタイルオブジェクト230内には含まれず、したがって、中継サーバ10-2に送信されて例えば送信リソースを取り散らかすこともない。

【0332】

インクデータ処理部100Tのインクデータ整形部140T(より具体的にはインクデータ通信部144T)は、ストロークオブジェクト210が生成されることのために、生成された描画スタイルオブジェクト230を最初のデータメッセージ(DATA_INK_BGNOS1001)に加える(図34参照)。

【0333】

DATA_INK_BGNOS1001メッセージは、そのメッセージが、そのストロークオブジェクト210(例えばメッセージタイプBGNOS、F101)、描画領域ID(F102)、描画領域内でストロークオブジェクト210を一意に識別するために使

10

20

30

40

50

用されるストローク I D (F 1 0 3)、描画スタイルオブジェクト 2 3 0 (F 1 0 4)、及び、オプションで(もし空間が許せば)ストロークオブジェクト 2 1 0 の部分を構成するいくつかの断片化データ (F 1 0 5) のための最初のメッセージであることを示す情報と、それに続く C R C 誤り訂正值とを含む。

【 0 3 3 4 】

描画スタイルオブジェクト 2 3 0 を生成し最初のデータメッセージ (D A T A _ I N K _ B G N O S 1 0 0 1) に含めた後、中継サーバ 1 0 - 2 から「 A C C E P T 」メッセージの受信を待つ代わりに、伝送装置 1 0 - 1 は、ストローク単位より小さい規定の単位 T ごとに、継続的に断片化データを生成する。複数の断片化データは、連帯して 1 つのストロークオブジェクト 2 1 0 を形成する (S 6 0 8)。具体的には、ペンイベントデータ入力操作の終了が検出されるまでの間、インクデータ生成部 1 2 0 T は、インクデータ 2 0 0 を製造するために、規定の単位 T ごとに 1 つのストローク (又はトレース) を形成するペンイベントデータを処理し、インクデータ整形部 1 4 0 T は、規定の単位 T に対応する通信メッセージ (図 3 4 の「 D A T A _ I N K _ M V D O S _ 1 , _ 2 , . . . , _ n 」) の形に断片化データを整形し、各メッセージ内の断片化データのそれぞれにストローク I D を割り当てる。

【 0 3 3 5 】

ここで使用される規定の単位 T は、ペンイベントデータ入力のサンプリング時間単位と同じかそれより大きい時間の単位、例えば 5 ミリ秒であってよい。したがって、単位 T は、5 ミリ秒、5 0 ミリ秒、2 0 0 ミリ秒などであり得る。代わりに又は追加で、規定の単位 T は、2 5 6 バイト及び 5 1 2 バイトの固定長のような、データサイズの単位であってよい。データ (バイト) サイズベースの単位 T は、図 3 4 に示した所与の通信プロトコルに従うメッセージ転送単位 (M T U) のデータ (バイト) サイズより小さく設定され得る。さらなる代替として、単位 T は例えば、時間単位 T とデータサイズ単位 T との間で動的に変更又は切り替えられることとしてもよい。

【 0 3 3 6 】

単位 T は、各アプリケーション又はタイミングの要求に応じて、適応的又は選択的にアプリケーション部 3 0 0 - 1 によって設定され得る。例えば、ストロークオブジェクト 2 1 0 が、あたかもリアルタイムに「描かれて」いるかのように継続的かつ成長的に受信装置 1 0 - 3 上に現れることを可能にする目的のため、アプリケーション部 3 0 0 - 1 は、生成されたポイントオブジェクト間で必要な補間処理が生じ得る最小の単位に対応するように単位 T を設定することとしてもよい。他の例として、同じメッセージヘッダ情報 (例えば、1 つのストロークオブジェクト 2 1 0 のためのすべてのメッセージに関連付けられるストローク I D) を送らなければならないことに関連する通信プロトコルのオーバーヘッド (通信量) を低減する目的のため、アプリケーション部 3 0 0 - 1 は、すなわちストロークオブジェクト自身の単位 (ストローク単位) にまで至る可能な限り大きな値に単位 T を設定することとしてもよい。他の例として、ペンイベントデータの入力レートと表示レートとの間に大きな違いがある場合 (例えば、ペンイベント入力 は 3 ~ 4 ミリ秒で到来するが、表示は 1 5 ミリ秒ごとにリフレッシュされる場合) にメッセージ全体を低減する目的で、アプリケーション部 3 0 0 - 1 は、表示レートにマッチするよう単位 T を設定してもよい。このケースでは、各ストローク断片化メッセージは、1 以上のポイントオブジェクトのためのデータを含むであろう。

【 0 3 3 7 】

図 3 3 は、伝送装置 1 0 - 1 及び/又は中継サーバ 1 0 - 2 で実行され得る単位 T を規定する方法を説明するフローチャートである。ステップ S 7 0 1 では、伝送装置と受信装置の間、又は、伝送/受信装置と中継サーバ 1 0 - 2 との間でパラメータ T が明示的に交渉されるか否かが判定される。もし「 Y E S 」であれば、明示的に交渉された値が単位 T として設定される (ステップ S 7 0 3)。例えば、単位 T は、ストロークデータが断片化されると設定 (S 5 2 4 _ 1 c、「 T R U E 」) し、かつ、受信遅延の最大値が 1 0 0 ミリ秒であると設定 (S 5 2 4 _ 1 d、「 1 0 0 」ミリ秒) するパラメータに従って交渉さ

れ定義され得る。一方、もしステップ S 7 0 1 の判定結果が「NO」であれば、アプリケーション部 3 0 0 の時間に関係する他の要求が単位 T を設定するために使用される。例えば、ステップ S 7 0 5 では、インクデータ 2 0 0 と他のデータ（音声、ビデオなど）が同期して再生されるかが判定される。もし「YES」であれば、ステップ S 7 0 9 で、「アグリゲーションが要求されたか」が判定される。これについては、図 4 3 及び図 4 4 を参照して後に完全に説明されるであろう。もし「NO」であれば、ステップ S 7 1 1 で、1 ストロークに対応する単位より短い単位 T が単位 T として設定される。また、規定の単位 T ごとの（断片化された）データのそれぞれに、他のタイプのデータとの同期再生の目的のためのタイムスタンプが付与される。一方、もしステップ S 7 0 5 の判定結果が「NO」であれば、タイミングについて特段の制約はないと仮定し、単位 T は、例えばストローク全体の単位のようなデフォルト値に設定される。

10

【0338】

図 3 2 に戻り、伝送装置 1 0 - 1 の入力処理部 1 1 1 がペンイベントデータ入力の終了（S 6 0 9）を検出すると、すなわち、1 つのストロークオブジェクト 2 5 0 の入力操作の終了を検出すると、インクデータ処理部 1 0 0 T は、ステップ S 6 1 1 においてメタデータオブジェクト 2 5 0 を生成する。メタデータオブジェクト 2 5 0 は、ディスプレイ上にストロークオブジェクト 2 1 0 を描画（レンダリング）するために使用される（描画スタイルオブジェクト 2 3 0 に含まれる）情報以外の、生成されたストロークオブジェクト 2 1 0 に関する情報を含む。例えば、メタデータオブジェクト 2 5 0 は、著者情報、ペン ID などを含む。

20

【0339】

<インクデータ中継処理>

中継サーバ 1 0 - 2 は、1 つのストロークデータ 2 1 0 のために断片化データを、伝送装置 1 0 - 1 からのデータメッセージの中において受信されたとき、修正することなく受信装置 1 0 - 3 に転送する。

【0340】

中継サーバ 1 0 - 2 は、まず、伝送装置 1 0 - 1 からストロークオブジェクト 2 1 0 の最初のメッセージ（DATA__INK__BGNO S）を受信する。それから、図 3 2 のステップ S 6 1 3（受け入れ）において、リソース調停（又は Q o S）制御が受信された最初のメッセージ DATA__INK__BGNO S とともに始まるストロークオブジェクト 2 1 0 の新たなエントリ（更新）を許可するか否かを決定する。図 3 2 のシーケンスは、リソース調停（又は Q o S）制御が、伝送装置 1 0 - 1 によって生成されたストロークオブジェクト 2 1 0 のエントリを許可すると決定した場合を図示している。中継サーバ 1 0 - 2 はその後、単にメッセージヘッダを確認することにより同じストローク ID に関連付けられたすべての後続のデータメッセージを識別し転送できるようストローク ID を格納し、受信装置 1 0 - 3 に対して最初のメッセージ DATA__INK__BGNO S を転送する。

30

【0341】

中継サーバ 1 0 - 2 は、同じストロークオブジェクト 2 1 0 のために後続のデータメッセージ（DATA__INK__MVDOS）を、それらのヘッダに含まれるストローク ID を確認しつつ、リソース調停（又は Q o S）制御に後続のデータメッセージのそれぞれについて独立した決定をさせることなく転送し続ける。このように、リソース調停の決定は、ストロークごとに 1 回だけ実行される必要がある。

40

【0342】

ストロークオブジェクト 2 1 0 のための複数のデータメッセージのうちの最後の 1 つが受信されるステップ S 6 1 5 では、中継サーバ 1 0 - 2 はその最後のメッセージを受信装置 1 0 - 3 に転送し、リソース調停（又は Q o S）制御は、この時点で完全に完成されたそのストロークオブジェクト 2 1 0 を転送するための中継サーバ 1 0 - 2 の処理リソースを解放する。

【0343】

50

ステップ S 6 1 7 では、1 つのストロークオブジェクト 2 1 0 の転送の完了に続く後処理動作として、ストロークオブジェクト 2 1 0 が描画領域に追加され、同時に中継サーバ 1 0 - 2 の描画領域リポジトリに格納される。

【0344】

中継（転送）処理がストロークオブジェクト 2 1 0 ごとに実行されることから、伝送フォーマットでリアルタイムに転送されたインクデータ 2 0 0 は、中継処理が完了した時点で記録フォーマットに容易に変換されることができ、圧縮された記録フォーマットでリポジトリ内に格納される。このことは、複数の装置の 1 つによって生成されたストロークオブジェクト 2 1 0 の中継が完了する度に、中継サーバ 1 0 - 2 が、その複数の装置によって共有される共通描画領域（又はキャンバス）の状態を効率的に更新することを可能にする。

10

【0345】

＜インクデータ受信処理＞

受信装置 1 0 - 3 はまず、新しく生成されたストロークオブジェクト 2 1 0 の最初のデータメッセージ（DATA__INK__BGNO S）を中継サーバ 1 0 - 2 から受信する。受信装置 1 0 - 3 は、例えば色、ペン先タイプ、透明度／不透明度、レンダリング方法などのような描画スタイルオブジェクト 2 3 0 に含まれる情報を用いて、そのディスプレイ上に受信したストロークオブジェクト 2 1 0 の断片化データを描画するための描画（レンダリング）オペレーションを開始する。

【0346】

インクデータ処理部 1 0 0 R は、描画スタイルオブジェクト 2 3 0 に含まれる同じ情報を用いて、続けて受信されるデータメッセージに含まれる同じストロークオブジェクト 2 1 0 の断片化データを処理し続けることによってインクデータ 2 0 0 を生成し、そのインクデータ 2 0 0 をグラフィック処理部 3 0 0 に出力する。

20

【0347】

グラフィック処理部 3 0 0 は、断片化されたインクデータ 2 0 0 の描画（レンダリング）処理を、それらが受信されるに従って開始及び継続する。図 3 2 のステップ S 6 1 7 - 1 ~ S 6 1 7 - n はそれぞれ、異なる時刻のディスプレイ上におけるストロークオブジェクト 2 1 0 のレンダリングされた形態を表している。図 3 2 は、ストロークオブジェクト 2 1 0 が部分的にのみ、しかし成長しながら描画される状態 G 1, G 2, G 3 から、ストロークオブジェクト 2 1 0 が完全に描画される状態 G 4 まで、ユーザが他のユーザによって紙上に描かれた実際のペンストロークを見る過程と同様にしてストロークオブジェクト 2 1 0 の断片化データがますます受信され処理されるにつれ、そのストロークオブジェクト 2 1 0 が連続的に描画されることを示している。

30

【0348】

図 3 4 は、本発明の一実施の形態による、複数の装置間でのリアルタイム通信に適した通信パケットの形にアレンジされた、データメッセージの伝送フォーマットのサンプルを図示している。図示した例は、伝送装置 1 0 - 1 が接続されているメディアインターフェイス（例えばイーサネット（登録商標））により規定される通信プロトコルのメッセージ転送単位（MTU）より小さい規定の単位 T を用いるものである。

40

【0349】

図 3 4 の一番上にある通信パケットは、伝送装置 1 0 - 1 のネットワーク通信部 1 1 2 から出力される通信パケットを示している。

【0350】

図 3 4 の「DATA__INK__BGNO S」は、新しいストロークオブジェクト 2 1 0 のための複数のデータメッセージのうちの最初のデータメッセージを示している。

【0351】

F 1 0 1 フィールドは、データメッセージの型を特定する。最初のデータメッセージ「DATA__INK__BGNO S」の F 1 0 1 フィールドは、それが新しいストロークオブジェクト 2 1 0 のための複数のデータメッセージのうちの最初のものであることを意味す

50

る「BGNOS」を示す。

【0352】

F102フィールドは、新しいストロークオブジェクト210が追加され重ねられる描画領域の描画領域IDを含む。

【0353】

F103フィールドは、ストロークオブジェクト210のストロークIDを含む。ストロークIDは、各ストロークオブジェクト210に共通描画領域で使用されるときにユニークに割り当てられる。様々な方法が、描画領域ごとに各ストロークオブジェクト210にユニークなストロークIDを割り当ててを可能にする。例えば、描画領域中で現在使用されている（最も高い）ストロークIDの値は、新しいストロークオブジェクト210に割り当てられる新しいユニークなIDを得るためにインクリメントされ得る。又は、新しいストロークオブジェクト210を入力するために使用されるペンのペンIDの値が、新しいストロークオブジェクト210のためのユニークなストロークIDを得るためにインクリメントされてもよい。伝送装置10-1はこのようにして、それが生成する新しいストロークオブジェクト210にユニークなストロークIDを独立して割り当てることができる。また、上で議論したように、中継サーバ10-2のリソース調停（又はQoS）制御544は、伝送装置10-1がユニークなストロークIDを割り当ててことに失敗した場合、オーバーラップする複数のストロークIDの使用を妨げる。伝送装置10-1は、新しいストロークオブジェクト210にユニークなストロークIDを独立して割り当てることができるので、そうでなければ異なる装置により生成されたストロークオブジェクト210にユニークなストロークIDを一元的に割り当ててのために要求される通信シーケンスは必要とされない。したがって、伝送装置10-1は、ペンダウンタイム、すなわちユーザがストロークの描画を開始した直後にインクデータの送信を開始することが可能になる。

【0354】

F104フィールドは、描画スタイルオブジェクト230、又は、描画スタイルオブジェクト230に含まれ、いずれもストロークオブジェクト210をどのように描画（レンダリング）するかに関係する一組の描画パラメータを含む。この一組のパラメータは、例えば、形状パラメータ（図3Bに示す円、又は、図3Cに示すフレーク／小片）、形状の半径又はサイズ、形状の半径又はサイズの最小値及び最大値、（その筆圧やペンの傾きで描かれるストロークがディスプレイ上にどのように現れるかに影響する）筆圧又はペンの傾きの情報などを含み得る。

【0355】

いくつかの実施の形態では、一組の描画パラメータの全体がF104フィールドに含まれる。他の実施の形態では、一組以上の描画パラメータ（又は1以上の描画スタイルオブジェクト）が前もって伝送装置10-1と中継サーバ10-2（と受信装置10-3）の間で共有され、どの組の描画パラメータがストロークオブジェクト210をレンダリングするのに使用されるかを特定する1つのパラメータセットのID（又は描画スタイルオブジェクトのID）のみがF104フィールドに含まれることとしてもよい。例えば、よく使われるペン先タイプ（鉛筆、ブラシ、ペン）及びよく使われる色の限られた数の組み合わせのような、頻繁に使用されるパラメータ設定が予め、各設定にユニークなIDを与えつつ定義されてよく、これらの設定IDが、ストロークオブジェクト210がディスプレイ上にどのようにレンダリング（描画）されるかを通信するためにF104フィールドに含まれてもよい。パラメータ設定は、装置間で初期交渉の時点で明示的に交渉されてもよいし、システムのプロトコルの一部として暗黙的に定義されてもよい。

【0356】

F105フィールドは、連帯してストロークオブジェクト210を形成する複数の断片化データのうちの最初の断片化データを含む。最初のデータメッセージのF105フィールドは、例えば、ストロークオブジェクト210の開始座標位置を含む。図34が最初のデータメッセージ（DATA__INK__BGNOS）の中に含まれるものとして最初の断

片化データを（F 1 0 5 内に）示しているが、もし（描画スタイルオブジェクトを含まなければならない）最初のデータメッセージ内に十分な空きがなければ、最初の断片化データ（F 1 0 5）は、2 番目のデータメッセージ内に含まれてもよい。

【0357】

図 3 4 のデータメッセージ「DATA__INK__MVDOS」__1, __2, ... は、すべて同じストロークオブジェクト 2 1 0 に関係しており、それぞれ 2 番目の断片化データ（F 1 0 6）、3 番目の断片化データ（F 1 0 7）、... を含む。最初のデータメッセージとは違い、これら後続のデータメッセージは描画スタイルオブジェクト（F 1 0 4）を含まず、F 1 0 1 において、「後続の」（最初ではない）タイプのデータメッセージとして識別される。後続のデータメッセージはすべて、最初のデータメッセージと同じ描画領域 ID 及び同じストローク ID を含む。

10

【0358】

データメッセージ「DATA__INK__ENDOS」は、ストロークオブジェクト 2 1 0 のための最後のデータメッセージであり、ストロークオブジェクト 2 1 0 の最後の断片化データを含み得る（F 1 0 8）。また、最後のデータメッセージは、F 1 0 9 に、著者情報などの非描画関連情報を含むメタデータオブジェクト 2 5 0 を含む。

【0359】

図 3 5 A は、TCP のようなビルトインのデータ再送メカニズムを含む通信プロトコルにおいて使用され得る伝送フォーマット、具体的には、データメッセージ「DATA__INK__MVDOS」を含む通信パケットを説明している。一方、図 3 5 B は、UDP のようなビルトインのデータ再送メカニズムを含まない通信プロトコルにおいて使用され得る通信パケットを説明している。図 3 5 A のデータメッセージフォーマットとは違って、再送メカニズムのないプロトコルでの使用のための図 3 5 B のデータメッセージフォーマットは、各データメッセージにユニークに割り当てられる通し ID を含む追加のフィールド F 1 1 0 を含んでよい。この通し ID は、メッセージの消失を検出し、再送信を起動／要求するために使用され得る。

20

【0360】

具体的には、図 3 6 A は、データ再送メカニズムを含まない UDP のような通信プロトコルでの実行に適した、通し ID を用いるデータ再送処理のサンプルを示すシーケンス図である。例えば、図 3 6 A は、送信側のパケット再送信パラメータ（5 2 4 __1 a）が「FALSE」に設定され、メッセージ再送信パラメータ（5 2 4 __1 f）が「ENABLE」に設定されている場合に適合する。ハッチングされた期間「d」は、図 3 2 の期間 d に対応する。

30

【0361】

図 3 6 A は、それぞれ通し ID # 4 5 6, # 4 5 7, 及び # 4 5 9 に関連付けられた第 1、第 2、第 4 のデータメッセージが伝送装置 1 0 - 1 から送信され、中継サーバ 1 0 - 2 に成功裏に受信されたことを示している。図 3 6 A はまた、通し ID # 4 5 8 に関連付けられた第 3 のデータメッセージが、伝送装置 1 0 - 1 と中継サーバ 1 0 - 2 の間のネットワーク内のどこかで失われ、中継サーバ 1 0 - 2 によって受信されないことを示している。

40

【0362】

中継サーバ 1 0 - 2 は、通し ID # 4 5 8 に関連付けられた第 3 のデータメッセージが期待されるが受信されず、代わりに通し ID # 4 5 9 に関連付けられた第 4 のデータメッセージが受信された時刻 T 1 において、通し ID # 4 5 8 に関連付けられた第 3 のデータメッセージが失われたことを検出する。その後、中継サーバ 1 0 - 2 は、通し ID # 4 5 8 に関連付けられた第 3 のデータメッセージが受信されていないことを示す通し ID # 4 5 8 を含む否定応答（NACK。DATA__INK__NACK）メッセージを発行する。

【0363】

伝送装置 1 0 - 1 は、NACK メッセージが受信された時刻 T 2 において、通し ID # 4 5 8 に関連付けた第 3 のデータメッセージが失われたことを検出し、通し ID # 4 5 8

50

に関連付けた第3のデータメッセージ及び通しID#458に続く後続のデータメッセージ（すなわち、#459、#460、・・・）を再送信する再送信処理を開始する。まさに失われたデータメッセージだけでなくすべての後データメッセージを再送信する理由の1つは、知覚可能な時間遅延を避けるため、ストロークオブジェクト210の断片化データの送信を時系列で継続するためである。他の理由は、1つのデータメッセージが失われるときには、その失われたデータメッセージに続くデータメッセージもまた失われる可能性が高いからである。

【0364】

上述した再送信処理方法によれば、再送メカニズムを含まないプロトコルが使用されるときであっても、1つのストロークオブジェクト210を形成するために必要なすべてのデータメッセージが成功裏に受信されることを確実にすることが可能になる。この再送信処理方法は、伝送装置10-1と中継サーバ10-2の間で発生するものとして上述されたが、この方法は中継サーバ10-2と受信装置10-3との間でも同様に実行され得る。

【0365】

図36Bは、メッセージ再送信の他の例を示している。図36Aと同様に、ハッチングされた期間「d」は、図32の期間dに対応する。図36Bは、送信側のパケット再送信パラメータ（524__1a）が「FALSE」に設定され、メッセージ再送信パラメータ（524__1f）が「DISABLED」に設定されている場合に適合する。メッセージ再送信パラメータは、メッセージ再送信が可能であるか否かに関わりなく、メッセージ伝送遅延の許容量がほとんどない（すなわち、いかなる知覚可能な伝送遅延も有しないという厳格な要求がある場合）ときに「DISABLED」に設定される。

【0366】

図36Aと同じように、図36Bでもまた、それぞれ通しID#456、#457、及び#459に関連付けられた第1、第2、第4のデータメッセージが伝送装置10-1（又は送信側となる中継装置10-2）から送信され、中継サーバ10-2（又は受信装置10-3）に成功裏に受信されている。受信側である中継サーバ10-2は、通しID#458に関連付けられた第3のデータメッセージが期待されるが受信されず、代わりに通しID#459に関連付けられた第4のデータメッセージが受信された時刻T1において、通しID#458に関連付けられた第3のデータメッセージが失われたことを検出する。その後、時刻T2Bにおいて受信側は、再送要求を発行することなく、成功裏に受信された通しID#456、#457、及び#459に関連付けられたメッセージに基づき、失われた#458に関連付けられたデータメッセージに対応するデータの失われた部分を補充或いは補間するべく、補間処理（又はエラー隠蔽処理）（S1205）を実行し、それによって継続的で途切れのないディスプレイ処理を実現する。時刻T3Bで、通しID#461に関連付けられた最後のデータメッセージの処理が完了する。その後の時刻T4Bで受信側は、受信されたストロークIDを用いて、DATA__INK__ALL__OS__REQメッセージ1503（図39B参照）を発行する。その後送信側10-1（又は10-2）は、同定されたストロークの全体を含むメッセージであるDATA__INK__ALL__OSメッセージ1501（図39A参照）を送信する。受信側10-2（又は10-3）は時刻T5Bで、必要に応じて、受信されたDATA__INK__ALL__OSメッセージ1501を使用し、補充或いは保管されたデータの失われた部分のデータ訂正を実行する。

【0367】

図36Cは、図36BのステップS1205における補間処理（エラー隠蔽処理）で扱われる問題を説明している。メッセージ#457、#458、及び#459が1つの完全なストロークデータを形成すると仮定する。図36Cでは、メッセージ#456は制御位置P4561、P4562、及びP4563を含む。黒四角の位置P4561は、ストロークの開始位置であり、位置の絶対座標を含む。図示した例では、この絶対座標は（100，0）である。図では、黒四角の位置が（参照されるべきものとして）絶対座標に関連

付けられている。一方、黒丸の位置 P 4 5 6 2 は、参照されるべきものとして絶対座標又は直前の先行位置の座標に対する差分（オフセット）に関連付けられている。図示した例では、位置 P 4 5 6 2 は R e l（+20, 0）にあり、これは、P 4 5 6 1 の絶対座標（100, 0）に対して水平方向に+20、垂直方向に+0に位置していることを意味している。黒丸位置 P 4 5 6 3 は、直前の先行する黒丸位置 P 4 5 6 2 に対して R e l（+20, 0）にある。メッセージ# 4 5 7 は、それぞれその直前の先行位置に対して R e l（+20, 0）にある3つの位置 P 4 5 7 1, P 4 5 7 2, 及び P 4 5 7 3 を含む。図 3 6 B に示すように、メッセージ# 4 5 8 は受信側に受信されないと仮定する。すると、最後のメッセージ# 4 5 9 のクロスハッチングされた開始位置 P 4 5 9 1 は、決定されることができなくなる。位置 P 4 5 9 1 は、参照位置がない限り有用でない相対的なオフセット情報のみを含むからである。 10

【0368】

クロスハッチングされた開始位置 P 4 5 9 1 は、本発明の実施の形態による様々な方法により決定され得る。本発明の実施の形態による第1の方法は、なおも図 3 6 C を参照して記述されるように、受信側で実行される線形予測に関わる。例えば、メッセージが50ミリ秒ごとに送信される場合、失われたメッセージ内の推定される軌道部分は、既に得られている位置座標の速度及び加速度に基づいて予測され得る。図 3 6 C では、ストロークは P 4 5 6 3（メッセージ# 4 5 6 の最後の制御位置）から P 4 5 7 3（メッセージ# 4 5 7 の最後の制御位置）まで合計で R e l（+60, 0）だけ前進しており、したがって、行方不明の（失われた）メッセージ# 4 5 8 の最後の制御位置もまた、P 4 5 7 3 に対して R e l（+60, 0）だけ前進しているものと予測され得る。 20

【0369】

図 3 6 D は、本発明の実施の形態による、クロスハッチングされた開始位置 P 4 5 9 1 を得るための第2の方法を説明している。この方法では、黒四角位置 P 4 5 6 1, P 4 5 7 1, P 4 5 9 1 により示すように、送信側が、複数の断片化されたメッセージのそれぞれの開始制御位置において絶対座標を含むように変えることができる。一実施の形態では、伝送装置 10-1 は、もし（a）記録型とは異なるフォーマットでの出力が要求され、かつ、b）（i）ストロークの断片化が使用され（524__1において「ストロークデータ断片化可否」が「TRUE」）、（ii）メッセージ再送信がメッセージレイヤでは実行されず（524__1 a の「パケット再送信」が「TRUE」であり、かつ、524__1 f の「メッセージ再送信」が「DISABLED」）、（iii）各断片化メッセージに絶対座標を含むことが明示的に指示されている、という条件の1以上が満たされるならば、この方法を実行することを選択し得る。図 3 6 D はすべての断片化メッセージに絶対座標が含まれることを示しているけれども、絶対座標を含む頻度は変更され得る。例えば、IETF の RFC 3550 や RTP 報告などのような通信プロトコル報告は、受信側で検出された受信品質に基づき、送信側において絶対座標の含有頻度を変化させるために使用され得る。 30

【0370】

クロスハッチングされた位置 P 4 5 9 1 を得るための第3の方法は、図 3 6 C の線形（前進）予測とともに用いられ得る後方確認である。図 3 6 E では、図 3 6 C のケースとは異なり、最後のメッセージ# 4 5 9 の最後の制御位置 P 4 5 9 3 が四角く作られており、図示した例のように絶対座標（333, 0）を含む。メッセージ（# 4 5 8）の消失が検出されると、受信側は、絶対座標を含む最後のメッセージ# 4 5 9 の最後の制御位置 P 4 5 9 3 に基づき、クロスハッチングされた丸位置 P 4 5 9 1 の後方確認を実施する。図示した例では、クロスハッチングされた位置 P 4 5 9 1 は、制御位置 P 4 5 9 3（333, 0）から R e l（-20, 0）によって P 4 5 9 2 を計算し、もう1つの R e l（-20, 0）によって P 4 5 9 1 に至るまで後方に向かって計算され、（293, 0）となる。 40

【0371】

図 3 6 F は、上述した方法のいずれかに従って得られるクロスハッチングされた制御位置 P 4 5 9 1 を用いる、図 3 6 B のステップ S 1 2 0 5 で実行される補充又は補間処理（ 50

エラー隠蔽処理)を説明している。実線1201は、メッセージ#457に基づいて生成される部分的なストロークデータを示し、白丸P4573は、メッセージ#457に含まれる断片化されたストロークオブジェクト210の最後(終わり)の制御位置を示している。実線1203は、メッセージ#459に基づいて生成される他の部分的なストロークデータを示し、クロスハッチングされたP4591は、メッセージ#459に含まれる断片化されたストロークオブジェクト210の最初(始まり)の制御位置を示している。部分的なストローク1201と部分的なストローク1203の間の(消失したメッセージ#458に対応する)失われたデータは、補間を用いる、破線矢印1207に示す部分的なストローク1201から前進する方向と、他の破線矢印1209に示す部分的なストローク1203から後進する方向との両方向にかかる予測に基づいて導出される。

10

【0372】

図36Gは、双方向予測処理の一例を説明している。この例では、P4573とP4591を結ぶ直線(又は近似曲線)1221が形成され、この直線1221を、近似曲線関数としてそれぞれ表現され得る2つの等しい直線(又は曲線)セグメント1223及び1225に分割する中点P__pred__midstが発見される。

【0373】

上述したように、通信プロトコルが通信レイヤにおける再送メカニズムを含まないとき(例えばUDP)、システムは、図36Aに示すようなメッセージレイヤでのメッセージ送信を実行することと、図36Bに示すようなメッセージレイヤでさえメッセージ再送信を実行しないこととを適応的に選択し得る。したがって、例えばインクデータ200が音声データと同期して再生されるときのようなアプリケーションが厳格なタイミング要求を有しているとき、知覚可能な送信時間遅延を回避し、継続的でリアルタイムなインクデータ200の受信とディスプレイを実現するため、通信レイヤ及びメッセージレイヤのいずれにおいても再送信が無効にされる。

20

【0374】

図37は、伝送装置10-1によって発行された共通描画領域の更新要求が中継サーバによって拒否される場合のインクデータ通信を説明するシーケンス図である。例えば、伝送装置10-1が共通描画領域に新たなストロークオブジェクト210を追加することを要求し、その要求が中継サーバ10-2によって拒否される。伝送装置10-1が描画スタイルオブジェクト230を含む最初のデータメッセージ(DATA__INK__BGNO S)を送信する点までは、図32のシーケンス図に含まれるものと同じである。

30

【0375】

中継サーバ10-2が最初のデータメッセージを受信するとき、インクデータ中継部540は、処理リソースの任意の制限、ネットワークリソースの任意の制限、ユーザアクセスポリシーなどの規定の基準に基づき、ステップS613(拒否)において、最初のデータメッセージの受信を拒否する。

【0376】

中継サーバ10-2はその後、伝送装置10-1に対し、フィードバック送信部546から拒否メッセージ(DATA__INK__REJOS)を送信する。

【0377】

伝送装置10-1は、ストロークオブジェクト210のための最初のデータメッセージに応じて拒否メッセージが受信されたとき、ステップS610において、送信処理を中止し、ストロークオブジェクト210のための後続のデータメッセージの送信をやめる。別の言葉で言えば、拒否メッセージが返送されない限り、伝送装置10-1は、連続する複数のデータメッセージにおいて、1つのストロークオブジェクト210のためにすべての断片化データを送信する送信処理を、明示のACCEPTメッセージの返送を待つことなく継続する。

40

【0378】

本発明の様々な実施の形態のインクデータ送信方法によれば、(例えば完了するまでに数秒を要する)長いストローク(トレース)が1つの装置によって描かれたときでさえも

50

、異なる装置は、そのストロークの完成を待つことなく、共通描画領域に描かれるのと同時にそのストロークのディスプレイを開始することができる。所与のプロトコルシステム内にビルトインされ得る、又は、再送メカニズムのないプロトコルにおいて追加的に使用され得るいくつかの送信方法は、ストロークオブジェクト 210 が複数の断片化データに分けて送信されたときでさえも、受信側で完全なストロークオブジェクト 210 が成功裏に受信されることを確実にするために使用され得る。また、伝送装置 10-1 によって送信されたストロークオブジェクト 210 のためのデータメッセージのいずれかが中継サーバ 10-2 によって拒絶／拒否されるとき、伝送装置 10-1 は、送信処理を無効にすることによって、大切な処理リソース及びネットワーク（送信）リソースの浪費を回避する。

10

【0379】

図 38 は、中継サーバ 10-2 が伝送装置 10-1 からストロークオブジェクト 210 の断片化データを受け取り、受信装置 10-3 に対して中継するために、そのストロークオブジェクト 210 のための断片化データのすべてを結合するようにした、図 32 に示したシーケンス図の第 1 の変形例である。伝送装置 10-1 から中継サーバ 10-2 までの規定の単位 T ごとの断片化データの送信は、図 32 に示したものと同じである。

【0380】

この実施の形態では、受信装置 10-3 は、受信装置 10-3 が中継サーバ 10-2 によって中継されたインクデータ 200 をどのようにして受信するかを定義する受信パラメータを設定するために、中継サーバ 10-2 と予め交渉してもよい。例えば、受信装置 10-3 が中継サーバ 10-2 に対し、受信装置 10-3 が規定の単位 T ごとではなくストローク単位でインクデータ 200 を受信するつもりであること（524__1c の「ストロークデータ断片化可否」の値が「FALSE」）を示す情報のような通信設定情報を含む制御メッセージ（CTRL__JOIN__REQ524__1）を送信する。中継サーバ 10-2 は、中継サーバ 10-2 が（他の受信装置に対しては規定の単位 T でインクデータ 200 を中継することができる一方で）受信装置 10-3 に対しストローク単位でインクデータ 200 を中継することとなるよう、受信装置 10-3 から受信される通信設定情報を描画領域情報管理部 524 に格納する。

20

【0381】

その後、図 32 に示すように、伝送装置 10-1 はストロークオブジェクト 210 の断片化データの送信を開始する。中継サーバ 10-2 は、ステップ S614（バッファリング）において、最後のデータメッセージ（DATA__INK__ENDOS）が受信されるまで、そのストロークオブジェクト 210 のための断片化データの記憶を続ける。最後のデータメッセージが受信されたとき、中継サーバ 10-2 は、ストロークオブジェクト 210 の全体、関連する描画スタイルオブジェクト 230、メタデータオブジェクト 250 などとともに含むデータメッセージを準備して送信し、S617でのレンダリングのために、それを「DATA__INK__ALLOS」として受信装置 10-3 に対し送信する。

30

【0382】

この例では、中継サーバ 10-2 が全ストロークオブジェクト 210 に関係するインクデータ 200 を一度に送信するので、ストロークオブジェクト 210 内に含まれるポイントオブジェクトの数は既知となる。したがって、インクデータ 200 は、伝送フォーマットに代えて記録フォーマットで送信されることができる。記録フォーマットは冗長性が小さいので、伝送フォーマットよりも圧縮され得る。

40

【0383】

図 39A は、記録フォーマットでの「DATA__INK__ALLOS」のデータメッセージフォーマットのサンプルである。このサンプルは、関連する描画スタイルオブジェクト 230、メタデータオブジェクト 250、そのストロークオブジェクト 210 に含まれるポイントオブジェクトの数などとともに、1つのストロークオブジェクト 210 のためのインクデータ 200 のすべてを含んでいる。示されるように、このメッセージはメッセージ CRC を含み、もしメッセージサイズが所与の通信パケット MTU を超えた場合には

50

、そのメッセージは、インクデータメッセージレイヤの下層に位置する通信レイヤで2以上の通信パケット（パケット151、パケット152、パケット153など）に分割される。

【0384】

インクデータ200をストローク単位ごとに送信するとき、断片化データを規定の単位Tごとにリアルタイムでストリーミングするために使用されるものとは異なる通信プロトコルを使用することができる。例えば、UDPのような再送メカニズムを有しないプロトコルが規定の単位Tごとに断片化データをストリーミングするために使用されるのに対し、ストローク単位でインクデータ200を送信するためには、再送メカニズムを含むTCP又は（基本的にTCPに依拠する）HTTPが使用され得る。

10

【0385】

受信装置10-3は、ステップS617でストローク単位ごとのインクデータ200を受信し、1つの描画内にストロークオブジェクト210を描画（レンダリング）するためのグラフィック処理を適用する。

【0386】

図38の修正されたシーケンス例は、トラフィック速度が遅く、各ストロークが連続的に受信されなければならないリアルタイムストリーミングは実現困難であるようなネットワーク環境にある受信装置10-3による使用に適している。そのようなケースでは、受信装置10-3は、1つのストロークオブジェクトを一度にデータ損失なしで受信するために、ストロークオブジェクト受信するためのより強固な受信方法を採用することを選択できる。

20

【0387】

図39Bは、ストロークIDが知られているときにストローク全体にかかるストロークオブジェクト210のデータを要求するメッセージである「DATA__INK__ALL__REQ」1503のデータメッセージフォーマットのサンプルを説明する。このメッセージは、ストロークIDを使用するストロークオブジェクトデータの全体の再送信を要求するために使用され得る。

【0388】

図40は、ストロークオブジェクト210が「そのまま」（すなわち、断片化されずに）伝送装置10-1から中継サーバ10-2を介して受信装置10-3に伝送されるようにした、図32に示したシーケンス図の第2の変形例である。

30

【0389】

この実施の形態では、伝送装置10-1は、伝送装置10-1が中継サーバ10-2に対してインクデータ200をどのように送信するかを定義する送信パラメータを設定するために、中継サーバ10-2と予め交渉してもよい。例えば、伝送装置10-1が中継サーバ10-2に対し、伝送装置10-1が規定の単位Tごとではなくストローク単位でインクデータ200を送信するつもりであること（524__1cの「ストロークデータ断片化可否」の値が「FALSE」）を示す情報のような通信設定情報を含む制御メッセージ（CTRL__JOIN__REQ）を送信する。中継サーバ10-2は、中継サーバ10-2が（他の伝送装置からは規定の単位Tでインクデータ200を受信することができる一方で）伝送装置10-1からストローク単位ごとにインクデータ200を受信するよう制御されることとなるよう、伝送装置10-1から受信される通信設定情報を描画領域情報管理部524に格納する。

40

【0390】

図40では、図32に示すように、伝送装置10-1は「ペンイベントデータ入力開始」（例えば「ペンダウン」イベント）の検出をトリガとして用いることにより、インクデータ伝送処理を実行する。具体的には、ステップS605において、伝送装置10-1の入力処理部111がストローク開始点の入力を検出する。図40では、ペンイベントデータは、1つのストロークがストローク開始点で描画を開始しストローク終了点で描画を終了するまでの間、入力処理部111において、ステップS605からステップS609ま

50

でインクデータ処理部 100 T に継続的に入力される（そして同様に、インクデータ生成部 120 T において S 607 から S 611 まで。集合的には S 608 B）。

【0391】

伝送装置 10-1 のインクデータ処理部 100 T は、「ペンイベントデータ入力開始」（例えば、「ペンダウン」イベント）（S 605）をトリガとして用い、コンテキスト情報（入力 2）に基づいて描画スタイルオブジェクト 230 を生成する。

【0392】

伝送装置 10-1 のインクデータ整形部 140 T はその後、中継サーバ 10-2 への送信対象である最初のデータメッセージ（DATA__INK__BGNO S）を整形することができ、中継サーバ 10-2 は、それを受け入れる（S 613（受け入れ））か拒否するかを決定することができる。この場合における最初のデータメッセージは、中継サーバ 10-2 が伝送装置 10-1 からのインクデータ 200 を受け入れられるか否かを確認するために送信されるもので、そのようなものであるために、例えばストロークオブジェクト 210 の開始点の座標を含む必要はない。

【0393】

中継サーバ 10-2 から拒否メッセージ（NACK）が受信されない限り、伝送装置 10-1 は、ストロークオブジェクト 210 の全体のためのインクデータ 200 を、ストロークオブジェクト 210 の全体と、関連する描画スタイルオブジェクト 230 及びメタデータオブジェクト 250 などを含むメッセージ「DATA__INK__ALLOS」の形に整形し、中継サーバ 10-2 に対して送信する。

【0394】

インクデータ 200 をストローク単位で伝送するとき、インクデータを規定の単位 T で伝送するために使用されるものとは異なるデータフォーマットを使用することができる。例えば、規定の単位 T で断片化されるインクデータ 200 が伝送フォーマットで伝送され得るのに対し、ストローク単位のインクデータ 200 は記録フォーマットで伝送され得る。また、ストローク単位のインクデータ 200 を伝送するとき、より強固な（高い信頼性を有する）インクデータ 200 の伝送を実現するために、再送メカニズムを提供する通信プロトコルが使用され得る。

【0395】

中継サーバ 10-2 は、ストロークオブジェクト 210 の全体とともに、関連する描画スタイルオブジェクト 230、メタデータオブジェクト 250 などを含むデータメッセージ（DATA__INK__ALLOS）を受信し、それを「そのまま」（DATA__INK__ALLOS）、受信装置 10-3 に対して中継する。これを実現するため、受信装置 10-3 は、受信装置 10-3 が中継サーバ 10-2 から中継されるインクデータ 200 を規定の単位 T ごとではなくストローク単位で受信することを定義する受信パラメータを設定すべく、前もって中継サーバ 10-2 と交渉することとしてもよい。

【0396】

受信装置 10-3 は、ストローク単位でインクデータ 200 を受信し、ステップ S 617 で、1つの描画内にストロークオブジェクト 210 を描画（レンダリング）するためのグラフィック処理を適用する。

【0397】

図 40 の修正されたシーケンス例が、トラフィック速度が遅く、それゆえ各ストロークが継続的に送受信され得るリアルタイムの実現が難しいネットワーク環境にある伝送装置 10-1 及び受信装置 10-3 による使用に適したものとなり得る。いくつかのケースでは、伝送装置 10-1 及び受信装置 10-3 は、ストロークオブジェクト 210 を、一度に 1つのストロークオブジェクト 210 ずつデータ損失なく送受信するための、より強固な送受信方法を採用することを選択し得る。

【0398】

図 41 は、中継サーバ 10-2 が、伝送装置 10-1 からストロークオブジェクト 210 を受信し、受信装置 10-3 に対して中継又は送信するために、受信したストロークオ

10

20

30

40

50

プロジェクト 210 を複数の断片化データに分割する、図 32 に示したシーケンス図の第 3 の変形例である。

【0399】

先に示した図 40 のシーケンスと同様、伝送装置 10-1 は、ストロークオブジェクト 210 の全体にかかるインクデータ 200 を含むデータメッセージを、関連する描画スタイルオブジェクト 230、メタデータオブジェクト 250 などとともに生成し (DATA__INK__ALL OS)、中継サーバ 10-2 に対して伝送する (インクデータ生成部 120 T におけるステップ S607 から S611、集合的には S608 C を参照)。

【0400】

期間「d」の間、伝送装置 10-1 のインクデータ生成部 120 T は、ストロークオブジェクト 210 を形成するためにペンイベントデータから導出された複数のポイントオブジェクトのそれぞれに対応する時間をインクデータ 200 内に埋め込む。例えば、アプリケーション部 300-1 が音声と関係して使用されるとき (例えば、524__1g 「音声同期可否」パラメータが「ENABLE」)、伝送装置 10-1 は、音声又はその他のデータの再生タイミングとの同期の目的で、インクデータ 200 に対し、スライスごとにタイムスタンプ情報を挿入する。時刻は、中継サーバ 10-2 がインクデータ 200 を中継する際に用いることができ、かつ、受信装置 10-3 がインクデータ 200 を再生 (レンダリング) する際に用いることのできる「再生時刻」として埋め込まれる。例えば、時刻 T1 に生成されたインクデータ 200 の第 1 の部分 (第 1 のポイントオブジェクト) は、対応する時刻 D__T1 に中継サーバ 10-2 によって中継され、時刻 T2 (T1 と T2 の間) に生成されたインクデータ 200 の第 2 の部分 (第 2 のポイントオブジェクト) は、対応する時刻 D__T2 に中継サーバ 10-2 によって中継される、などである。同様に、時刻 T1 乃至 T4 に生成されたインクデータ 200 の第 1 乃至第 4 の部分 (ポイントオブジェクト) は、それぞれ、S617 の「グラフィック処理開始」ステップで始まる時刻 T1 から T4 に基づく対応するタイミングで、受信装置 10-3 によって再生 (描画) される。

【0401】

中継サーバ 10-2 は、伝送装置から、ストロークオブジェクト 210 の全体と、関連する描画スタイルオブジェクト 230、メタデータオブジェクト 250 などとのためのインクデータ 200 を含むデータメッセージ DATA__INK__ALL OS を受信し、伝送装置 10-1 が図 32 のシーケンス例で断片化データを生成するやり方と同様に、このインクデータ 200 を規定の単位 T ごとに複数の断片化データに分割する。規定の単位 T は、この例ではストローク単位より小さい。中継サーバ 10-2 は、生成した断片化データを、それぞれ D__T1, D__T2, D__T3, D__T4 のタイミングで受信装置 10-3 に対して出力する。

【0402】

受信装置 10-3 は、図 32 のシーケンス例における受信装置 10-3 と同様、連帯して 1 つのストロークオブジェクト 210 を形成する、断片化データを含む複数のデータメッセージを受信して処理する。具体的には、受信装置 10-3 は、それが 1 つのストロークオブジェクト 210 を構成する一連のデータメッセージのうちの最初のデータメッセージであることを示す最初のデータメッセージ DATA__INK__BGN OS を受信する。受信装置 10-3 は、この最初のデータメッセージに含まれる描画スタイルオブジェクト 230 を取り出し、ストロークオブジェクト 210 の断片化データのレンダリング (描画) を開始するために、描画スタイルオブジェクト 230 内の情報を使用する。インクデータ処理部 100 R は、後続のデータメッセージ内に受信される後続の断片化データの処理を続け、描画部 300 は、ストロークオブジェクト 210 の残りのレンダリング (描画) を成長する (増加する) ようなやり方で続ける。図 41 は、ストロークオブジェクト 210 が、それが部分的にのみ存在し、しかし成長しつつある状態 G1, G2, G3 からストロークオブジェクト 210 が完全に描画されている状態 G4 まで、その断片化データが徐々に受信され処理されるに従って連続的に描画されることを示している。上記の例では、

中継サーバ10-2がストロークデータの中継伝送のタイミングを制御しているが、中継サーバ10-2にストロークの単位でストロークデータを伝送させ、その後受信装置10-3が、受信したストロークデータの各断片の再生タイミングを制御するように構成することも可能である。さらに、中継サーバ10-2にストロークデータを維持させ、スケジュールされた時間に、1以上の受信装置に対してストロークデータのストリーミングを開始するように構成することも可能である。例えば、スケジュールされた時間は2020年1月1日の0:00に設定されることができ、受信装置は、そのスケジュールされた時間に、それぞれのディスプレイに描画されるべきメッセージ「Happy New Year」を示す一組のストロークの受信を開始することができる。

【0403】

上述した様々な実施の形態及び変形例は、さらなる実施の形態を提供するために結合され得る。本実施の形態の側面は、さらなる実施の形態を提供するために修正され得る。

【0404】

例えば、いくつかの実施の形態においては、受信装置10-3は、中継サーバ10-2が伝送装置10-1から伝送されたデータメッセージを受け入れるか拒否するかを決定するように構成されるのと同様に、中継サーバ10-2から中継されたデータメッセージを受け入れるか拒否するかを決定するように構成され得る。中継サーバ10-2は、拒否メッセージ(DATA__INK__REJOS)を受信すると、(同じデータメッセージが他の受信装置には転送され得る一方で)伝送装置10-1からのデータメッセージがその受信装置10-3には転送されないことを示すように接続リスト541を更新する。

【0405】

いくつかの実施の形態では、共通描画領域の原点は、ユーザ特有の描画領域の原点(すなわち、伝送/受信装置におけるディスプレイ113の原点)と同一である。他の実施の形態では、ユーザ特有描画領域の原点及び寸法が共通描画領域の原点及び寸法と同じである必要はない。図42は、共通描画領域の原点に関連して、ユーザ特有描画領域内におけるユーザ特有のストロークの開始ポイントの概念を説明する図である。

【0406】

図42では、ポイント1601の(0,0)が共通描画領域の原点を示しており、一方で、ポイント1603のユーザ10-1(0,0)は、ユーザ10-1にとってのユーザ特有描画領域の原点を示し、ポイント1605のユーザ10-3(0,0)は、他のユーザ10-3にとってのユーザ特有描画領域の原点を示している。(0,0)からユーザ10-1(0,0)へのベクトルは(200,30)となるように図示されており、(0,0)からユーザ10-3(0,0)へのベクトルは(130,260)となるように図示されている。通信を開始するときに、伝送装置10-1と受信装置10-3は、それぞれの共通描画領域からの原点ベクトル(図示した例における(200,30)及び(130,260))を中継サーバ10-2と共有する。したがって、伝送装置10-1がオフセット1-1(1607)で開始するストロークAを中継サーバ10-2に送信するとき、伝送装置10-1は、ユーザ10-1(0,0)から開始ポイントであるオフセット1-1までのベクトルであるベクトル「オフセット1-1」のみを送る必要がある。その後中継サーバ10-2は、共通描画領域の原点との関係で開始ポイント「オフセット1-1」の位置を決定するために、受信したベクトル「オフセット1-1」をベクトル(200,30)と結合することができる。

【0407】

中継サーバ10-2はその後、共通描画領域の原点に対して開始ポイント「オフセット1-1」の位置を定義する、ポイント1601(0,0)からポイント1603ユーザ10-1(0,0)を介してオフセット1-1(1607)に至る結合ベクトルを、受信装置10-3に送信することができる。受信装置10-3はその後、受信した結合ベクトルから自身の原点ベクトル(130,260)を減ずることにより、ユーザ10-3(0,0)(1605)という原点を有する自身のユーザ特有描画領域に対して、開始ポイント「オフセット1-1」の位置を定義することができる。或いは、中継サーバ10-2が、

受信装置 10-3 のユーザ特有描画領域のユーザ 10-3 (0, 0) (1605) という原点に対する開始ポイント「オフセット 1-1」の位置を算出するための減算動作を行ってもよい。中継サーバ 10-2 がユーザ 10-3 (0, 0) (1605) という原点に対する開始ポイント「オフセット 1-1」の算出された位置を受信装置 10-3 に対して送信した場合、受信装置 10-3 は直ちに、自身のユーザ特有描画領域内で受信した位置を処理することができる。

【0408】

これらの実施の形態では、中継サーバ 10-2 が共通描画領域の原点に関する座標変換を行うので、伝送及び受信装置 10-1 及び 10-3 は、自身で変換を行う必要はなく、自身の原点に対する座標データとして座標データ（ベクトルデータ）を送受信する。すなわち、一旦伝送及び受信装置 10-1 及び 10-3 が共通描画領域の原点に対する自身の原点を中継サーバ 10-2 と共有した場合、座標変換処理は、完全に伝送及び受信装置 10-1 及び 10-3 に対して透明となり得る。

【0409】

インクデータ 200 を送信するための規定の伝送単位 T は、ストローク単位と同じかそれより小さいとして記述されたが、ストローク単位より大きい単位でインクデータ 200 を伝送することも可能である。すなわち、インクデータ 200 を、ストローク単位よりも大きい意味の単位ごとに「集める」ことも可能である。例えば、図 33 に示した規定の単位 T の処理では、ステップ S709 において、アグリゲーションが要求されたかが判定される。もし「YES」であれば、ステップ S713 において、単位 T がストローク単位より大きい値、すなわち意味の単位に設定される。

【0410】

図 43 は、意味単位を説明する図であり、図 44 は、意味単位でインクデータ 200 を伝送するフローのサンプルを説明する図である。

【0411】

図 43 は、7つの手書き文字からなる単語「Network」を図示している。7011 は1番目のストロークを示し、7012 は2番目のストローク（連帯して最初の文字「N」を構成する）を示し、7013 は3番目のストロークを示し、7014 及び 7015 はそれぞれ、連帯して3番目の文字「t」を構成する4番目と5番目のストロークを示している。ここで、1番目及び2番目のストローク 7011 及び 7012 は、それらが連帯して1つの文字を示すことから、1つの意味単位を構成するものと看做される。同様に、4番目及び5番目のストローク 7014 及び 7015 は、それらが連帯して1つの文字を示すことから、1つの意味単位を構成するものと看做される。

【0412】

図 44 は、1つの意味単位を構成する複数のストロークを「集め」、意味の単位でストロークデータを伝送するフローのサンプルを図示している。この例では、アプリケーション 300-1 が文字認識エンジンにリンクされる。図 44 では、ハッチングされた期間「d_for_7011」が1番目のストローク 7011 が検出される期間である。インクデータ生成部 120T は、アプリケーション 300-1 によって使用される文字認識エンジンの分析結果に基づき、1番目のストローク 7011 が完全な意味単位を構成していないことを検出する。意味単位が完成していないので、ステップ S1911 において、プロセスはインクデータ伝送ステップをスキップする。その後、ハッチングされた期間「d_for_7012」の間に2番目のストローク 7012 が検出される。ステップ S1913 において、インクデータ生成部 120T は、1番目及び2番目のストローク 7011 及び 7012 が1つの意味単位を構成していることを検出する。ステップ S1915 において、インクデータ生成部 120T は、1番目及び2番目のストローク 7011 及び 7012 を集め、そしてまた、1番目及び2番目のストローク 7011 及び 7012 が連帯して1つの意味単位を構成し、連帯して「N」を意味していることを示すメタデータ情報を生成する。伝送装置 10-1 は、受信側（中継サーバ 10-2 又は受信装置 10-3）に対し、メタデータ情報とともにストローク（インク）データ（1501 DATA__INK

— A L L O S) を伝送する。ステップ S 1 9 1 7 では、受信側がメタデータを取り出し、1 番目及び 2 番目のストローク 7 0 1 1 及び 7 0 1 2 が連帯して 1 つの意味単位を構成し、連帯して「N」を意味していることを決定する。その後、ステップ S 1 9 1 9 では、受信側がそのディスプレイ上に、1 番目及び 2 番目のストローク 7 0 1 1 及び 7 0 1 2 をレンダリング（描画）する。

【0413】

第 3 の実施の形態

第 3 の実施の形態は、手書きデータを表すように構成されたインクデータを生成し再生する方法と、再生されたインクデータを使って図画を出力（レンダリング）する方法に向けられたものである。特に、インクデータを生成するための特定の入力装置センサに依存するペンイベントデータ 9 2 0 2 を抽象化（一般化）する図 9 2 の処理が記述される。例えば、非装置依存型の共通の属性値を生成するために、筆圧データを有するペンイベントデータ（タイプ 1）と、筆圧データを有しないペンイベントデータ（タイプ 2）とを抽象化（一般化）する処理が記述される。

【0414】

第 3 の実施の形態の背景

上記非特許文献 5 は、インクデータの標準を記述しており、標準の目的を次のように規定している。「J o t [標準] は、インクが非常に広い範囲のアプリケーション及び装置にわたって使用されることを可能にする。標準の交換フォーマットを使えば、多数のシナリオが可能となる。ここには、インク共有のいくつかの例がある。もちろん、J o t が多様なプラットフォームで実行されるのと同じくらい、より多くのアプリケーションが作られ得る。」

【0415】

この標準は、インクデータの属性の 1 つである筆圧データが存在することを示す「i n k F o r c e D a t a P r e s e n t」と呼ばれるフラグを使用する。

【0416】

ストロークを構成する複数のポイント（複数のドット）のそれぞれが、以下の表 1 のデータ構造で定義される。

【表 1】

表 1 非特許文献 5 のデータ構造

```
typedef struct tag_INK_POINT {
    XY32          position;    // required x/y point data
    S16           force;       // optional force data
    S16           height;      // optional z height data
    S16           rho;         // optional rotational data
    ANGLE16       angle;       // optional theta and phi data
    INK_BUTTONS   buttons;     // optional proximity, contact, button data
} INK_POINT, FAR *P_INK_POINT;
```

【0417】

上記において、「f o r c e」が筆圧に対応する値である。したがって、筆圧情報を取得可能な電子装置に対して、i n k F o r c e D a t a P r e s e n t フラグの値は筆圧データの存在を示すブール値に設定され、筆圧（「f o r c e」）の値は入力情報として入力される。

【0418】

非特許文献 1 は、他のインクデータ標準を記述しており、標準の目的を次のように規定

している。「ハードウェア及びソフトウェアベンダーは、典型的には、専有の或いは制限的なフォーマットを使用するデジタルインクを供給し、代表してきた。公開のかつ包括的なデジタルインクフォーマットの欠如は、複数のベンダーによって開発される異種の装置を跨がるデジタルインクのキャプチャ、伝送、処理、及び提示を厳しく制限してきた。この必要性に応じ、インクマークアップ言語（InkML）は、ソフトウェアアプリケーション間でのデジタルインクの交換を促進するために、シンプルでかつプラットフォーム中立なデータフォーマットを提供する。」

【0419】

この標準では、ストロークは「トレース」と呼ばれ、そのデータ構造は、予め定義された複数の属性セットの中から自由に選択される属性によって定義され得る。

10

【0420】

デフォルトの設定では、トレースは、以下の表2-1の構造による十進数の組み合わせ（X，Y）のリストとして表される。

【0421】

表2-1 非特許文献1のデータ構造（デフォルト）

```
<traceFormat xml:id="DefaultTraceFormat">
  <channel name="X" type="decimal"/>
  <channel name="Y" type="decimal"/>
</traceFormat>
```

【0422】

デフォルトのデータ構造に対し、さらなる属性がF（force）のような定義される追加のチャンネル名により追加され得る。例えば、ペン圧力を示す属性（チャンネル名）は、次のようなものである。

20

【表2】

チャンネル名	次元	デフォルトの単位	説明
F	Force	%	ペン先の力

【0423】

<traceFormat>の<channel name>タグを例えば「F（Force）」に設定することにより、力属性を含む「トレースフォーマット」をカスタム定義することができる。

30

【0424】

トレースはその後、十進数の組み合わせ（X，Y）として、任意のカスタム定義された「トレースフォーマット」に従って表される。したがって、カスタム定義されたトレースフォーマットで表されるインクデータ（例えば座標、筆圧など）は、継続する値のリストとして提供される。

【0425】

非特許文献2は、産業においてインクデータを標準化しようとする他の試みを記述するもので、その目的を次のように規定している。「インクシリアルライズトフォーマット、すなわちISFは、書かれたインクデータを格納するためのマイクロソフト（登録商標）のフォーマットである。このフォーマットは主に、携帯情報端末、タブレットPC、ウルトラモバイルPCがスタイラスにより入力されたデータを格納するために使用される。」非特許文献3はさらに、「インクオブジェクトは一連のストロークに過ぎず、各ストロークは一連のポイントであり、ポイントはX及びY座標である。新たなモバイル装置の多くはまた、圧力や角度のようなカスタム情報を提供する。加えて、[それらは] インクデータとともにカスタム情報を格納するために使用され得る。」

40

【0426】

筆圧情報が含まれるとき、次の属性がデータフォーマット内に含まれ得る。これは、圧

50

力が x , y の次に重要であることを示している。

T A G _ N O R M A L _ P R E S S U R E

【 0 4 2 7 】

T A G _ N O R M A L _ P R E S S U R E 情報を含む情報はその後、順番に並べられて出力される。

【 0 4 2 8 】

第 3 の実施の形態の概要

非特許文献 1, 2, 及び 5 に提案される上述したインクデータの構造は、情報を抽象化又は一般化することなく、ペン型入力装置によって得られた装置特有の情報を出力しようとするものである。例えば、筆圧データがペン型入力装置による属性として取得されるとき、その筆圧データは、圧力データを受信して処理する能力を有する電子装置に対して出力され得るが、ほとんどの静電容量型のタブレットのような、そのような能力を有することが期待されない電子装置に対しては出力されない。同様に、ペン型入力装置が筆圧データを取得する能力を有しないとき、電子装置が筆圧データを受信して処理する能力を有しているとしても、その電子装置に対して筆圧データが出力されることはない。その上、静電容量型のセンサのような、指による手書き入力を受信する能力を有するいくつかの「指」型入力装置は、指がセンサ表面に押し付けられたときに指の圧力データを生成する能力を有しない。このように、すべてのペン型入力装置又は指型入力装置が筆圧データを取得する能力を有しているわけではなく、すべての電子装置（例えば、タブレット及び PC）が筆圧データを取得し、受信し、処理する能力を有しているわけではないので、現在利用可能なインクデータ内の筆圧データの利用は、どちらかと言えば限定されている。このことは、現在利用可能なインクデータを、手書きデータをよりリアルにシミュレートし、表現するという目的に適さないものとする。なぜなら、現実においては、ユーザによって適用される筆圧が、ペンストロークの幅や濃度に影響し、又は、サイズを変更するしみやよごれを生成することにより、ペンストローク（又はペントレース）がどのように紙の上に現れるかということに多大な影響を及ぼすからである。

【 0 4 2 9 】

非特許文献 1, 2, 及び 5 に提案される技術は、ペンイベントの間、どれだけの筆圧が適用され、ペンの角度はどれくらいか、というような情報を含むペンの動き（すなわち、「ペンイベント」）を記録し、再生することを目的としている。しかしながら、記録され再生されることが究極的に必要である情報は、複数のストローク又はトレースがスクリーン上にどのように現れるかという、結果としての一連のペンイベントの「外観」である。一連のペンイベントの結果としての外観は、典型的にはグラフィックデータとして使用される。非特許文献 3 は、ベクトルデータの標準を提案する。しかし、非特許文献 3 は入力装置としてのペンの使用に無関心であり、結果としてそのデータは、ペンの使用の結果として得られるストロークを表し、表現するのに適さない。例えば、非特許文献 3 の技術は、バージョン 1.1 では、ストローク幅を変えることやストロークの透明度を変えることを許容しない。

【 0 4 3 0 】

筆圧データをサポートしない装置によっても使用可能であるよう筆圧データを要求しないインクデータモデル（意味又は言語）に基づいてインクデータを生成し、再生する方法、及び、そのインクデータモデルに基づくインクデータフォーマットに対するニーズが存在する。いくつかの実施の形態では、インクデータを生成する方法は、すべてではないにしてもほとんどの多様な装置で容易に利用可能な情報に基づいて圧力の代用パラメータを生成する能力を有する。圧力情報を表現するために本発明のインクデータモデル内における代用パラメータを使用することは、実際のペンストロークをシミュレートする多様なニュアンスの表現やリアルな外観とともにレンダリング（描画）されたペンストロークを提供することを実現する。

【 0 4 3 1 】

第 3 の実施の形態の記述

以下に示す本発明の様々な代表的な実施の形態の記述は、一般的に、以下に示す 6 セクションにまとめられる。

- [1] システムアーキテクチャの全貌。図 1 及び図 4 5～図 4 7 B を参照。
- [2] ストローク言語 (S L) 及びストロークファイルフォーマット (S F F)
- [2-1] インクデータモデル (「ストローク言語」)。図 4 8 A～図 4 8 I を参照。
- [2-2] ストロークファイルフォーマット (S F F)。図 4 8 J～図 4 8 L を参照。
- [3] インクデータ処理部 1 0 0 (生成部 1 0 0 T)。図 4 9～図 6 3 を参照。
- [4] インクデータ処理部 1 0 0 R (再生部 1 0 0 R)。図 6 4～図 6 6 を参照。
- [5] インクデータレンダリング処理。図 6 7 を参照。
- [6] 効果。インクデータ描画例。図 6 8～図 7 2 を参照。

10

【0 4 3 2】

- [1] システムアーキテクチャの全貌 (図 1 及び図 4 5～図 4 7 B)

第 1 の実施の形態を参照して上述した図 1 は、本発明のインクデータ 2 0 0 が生成され、再生され、スクリーン上にレンダリング (描画) される、本発明の第 3 の実施の形態によるシステムの全貌を説明している。特に、図 1 に実線で描かれたインクデータ交換インフラ 1 0 は、共通の言語モデルに基づいてインクデータ 2 0 0 を使用するライブラリによって実現されるインフラを表しており、そこでは、圧力データをサポートする様々なアプリケーションサービス及び装置、並びに、圧力データをサポートしない他のアプリケーションサービス及び装置によって共通に使用可能となるようにインクデータ 2 0 0 が生成される。

20

【0 4 3 3】

図 1 に示す装置 1 0-1-1 は、入力センサとして、筆圧データを感知できるペン型の入力装置を含む。装置 1 0-1-1 は、アプリケーションサービス # 1 によって提供されるアプリケーションを用いてインクデータ 2 0 0 を生成する。生成されたインクデータ 2 0 0 はその後、目的地のメディア (例えばネットワーク) に対応する適切な出力フォームで出力され得る。

【0 4 3 4】

装置 1 0-1-2 は、ユーザの指によりなされる手書き入力を受信する能力を有するタブレット型の入力装置である。装置 1 0-1-2 のセンサは、筆圧データを出力する能力を有しないが、圧力データを要求しないアプリケーションサービス # 2 のために提供されるアプリケーションを用いてインクデータ 2 0 0 を生成することができる。インクデータ 2 0 0 はその後、目的地のメディアに対応する適切なフォームで出力され得る。

30

【0 4 3 5】

装置 1 0-3 は、アプリケーションサービス # 2 に加入しているデスクトップ型の P C である。装置 1 0-3 は、アプリケーションサービス # 2 によって提供されるアプリケーションを用いて、装置 1 0-1-1 又は装置 1 0-1-2 から出力されるインクデータ 2 0 0 を処理する (例えば、ディスプレイスクリーン上にレンダリングしたり、再配布する) ことができる。

【0 4 3 6】

装置 1 0-1-1, 1 0-1-2 及び装置 1 0-3 上で動作するアプリケーションサービス # 1 及びアプリケーションサービス # 2 はいずれも、インクデータ 2 0 0 を表す共通の情報モデル (「ストローク言語」) を利用する。

40

【0 4 3 7】

本発明の実施の形態によるインクデータ 2 0 0 は、多様なサービスアプリケーション上の様々なタイプの装置によって使用可能とされ、それ自体に筆圧データを取得する又は処理する能力を有する装置及びサービスアプリケーションに限定されない。

【0 4 3 8】

図 4 5 は、インクデータ 2 0 0 の処理部 1 0 0 (生成部 1 0 0 T) におけるデータ入出力を説明するとともに、左手側においてインクデータ生成方法を説明し、また、インクデータ処理部 (再生部 1 0 0 R) におけるデータ入出力を説明するとともに、右手側におい

50

てインクデータ生成方法を説明している。インクデータ処理部100Tは、第1の実施の形態の図5における100Tに対応する。

【0439】

本発明の実施の形態によるインクデータ処理部100Tは、装置10-1-1のセンサ及び装置10-1-2のセンサからペンイベントデータを受信する。ペンイベントデータは、ペン（又は指）のストロークを生成したペン（又は指）の動きを表しており、ストロークの座標データ（例えば（X，Y）位置）を含む。ペンイベントデータはまた、筆圧データやペンの傾きデータなどの装置依存型のデータを含み得る。図45では、装置10-1-1からのペンイベントタイプ1データは筆圧データを含む。装置10-1-1のセンサは、圧力感知可能スタイラスとともに使用されるEMRタイプのペンタブレットのためのペンタブレット装置ドライバ又はAPIのような、圧力データを出力する能力を有する装置である。これに対し、装置10-1-2からのペンイベントタイプ2データは、如何なる圧力データも含まない。装置10-1-2は、iPad（登録商標）タブレットや他の容量タイプのタッチセンサタブレットのような圧力データを出力する能力を有しない装置である。インクデータ処理部100は、装置依存型のデータ（タイプ1及びタイプ2）を受信し、非装置依存型の本発明の実施の形態によるインクデータ200を生成して出力する。このインクデータ200は、圧力データを処理する能力に関わらず、様々な装置によって共有され得る。

【0440】

ペンイベントデータに基づいてインクデータ200を生成するため、インクデータ処理部100はまた、図45に示すように、ペンイベントデータを入力するために使用されるアプリケーション又はオペレーティングシステムから、ペンイベントデータについてのコンテキスト情報（入力2）を受信する。生成されたインクデータ200は、様々なオブジェクト（ストロークオブジェクト210、メタデータオブジェクト250、描画スタイルオブジェクト230、操作オブジェクト270など。図48B参照）のコレクションであり、これらは図5に示すツリー構造で整理される。コンテキスト情報（入力2）は、例えば、描画アプリケーションから受信されるペン（先）タイプ（例えば、鉛筆、クレヨン、ブラシなど）及び／又はペン色の情報、ペンタブレットの解像度情報及びサンプリングレート、ペンイベントの入力時刻及び位置情報、ペンID、OSによって提供されるユーザID、並びに、ストロークを生成（描画）するために使用された（ソフトウェア）アプリケーションによって提供されるストロークに関する他の任意の情報を含む。

【0441】

インクデータモデルは、インクデータ200を処理するアプリケーション領域（ドメイン）において使用されるデータの意味を定義する。様々なアプリケーション及びサービスの間で共通のインクデータモデルが共有されるとき、情報は、インクデータ200を処理するドメイン内のシステム構造の中において、信頼性を持って自由に共有され得る。

【0442】

図45では、破線矢印が、インク情報モデルが定義するインクデータ200の各クラスオブジェクトを示しており、インクデータオブジェクトのコレクションがインクデータ200を構成する。

【0443】

インクデータ処理部100は、生成したインクデータ200を、データオブジェクトのコレクションとして、インクデータ200を必要とするアプリケーションによって要求されたメディアフォーマット（例えば、ファイル、パケットなど）で出力する。一般的に、メディアフォーマットには2つのタイプがある。

（1）永続的（半永久的）な形態でインクデータ200を記憶するストロークファイルフォーマット（SFF）

（2）メッセージ又はリアルタイム通信においてインクデータ200を伝送するのに適したストロークメッセージングフォーマット（SMF）

【0444】

インクデータ処理部100Rは、インクデータ200をSFF又はSMFファイルで受信し、特定のメディアに記録されたインクデータ200を解釈して、バイト又はビットのシーケンスとして取り出し、取り出したインクデータ200を、インクデータ200を利用する様々なタイプのアプリケーション300に対して提供する。インクデータ処理部100Rは、アプリケーション300-1, ..., 301-nのうちの1つからの要求に応じて、インクデータオブジェクトを取り出し、インクデータモデルによって定義される情報を再生し、要求しているアプリケーション300-1, 300-2に対して、取り出した(再生した)インクデータ200を提供する。インクデータ処理部100Rは、典型的には、要求しているアプリケーション300に動的に又は静的にリンクされたライブラリとして具体化され、パーソナルコンピュータ内のプロセッサ上で実行される。他の実施の形態では、インクデータ処理部100Rは、インクデータモデルを解釈することによってインクデータ200を再生するように構成されたクラウドサーバであってもよい。

【0445】

アプリケーション300-1, ..., 300-nのそれぞれは、インクデータ処理部100Rによって取り出され、再生されたインクデータ200の必要な量/部分を引き出し、様々な動作を実行する際に、引き出したインクデータ200を利用する。例えば、アプリケーション300-1は、図示した実施の形態では描画アプリケーションである。描画アプリケーションは、再生されたインクデータ200の完全なセットを引き出し、引き出したインクデータ200に対して、例えば、拡大縮小、ラスタ化、レンダリング動作を適用し、画像ファイルとして出力することもできる。

【0446】

様々な実施の形態において、インクデータ処理部100及びインクデータ処理部100Rは、生成部/再生部の機能を実現するライブラリを利用するアプリケーション300とともに、それぞれメモリとCPU(中央処理ユニット)又はGPU(グラフィック処理ユニット)を含む1以上のコンピュータ装置内に具体化され得る。例えば、インクデータ処理部100、インクデータ処理部100R、及び描画アプリケーション300-1は、1つのコンピュータ装置内に具体化され得る。この場合、インクデータ処理部100及びインクデータ処理部100Rは、共通のインクデータオブジェクトを共有するために、装置のメモリを使用し得る。図46A~図46Cはそれぞれ、本発明の実施の形態によるインクデータ生成装置又は方法の3つの構成例を説明している。

【0447】

図46Aは、装置10-1-2に結合され得るストローク入力端末とは異なるサーバ内にインクデータ処理部100Tが存在するような、装置又は方法を説明する。装置10-1-2は、筆圧データを含み又は含まない、装置依存型のペンイベントデータを取得する。図示した実施の形態では、装置10-1-2のセンサが筆圧データを含まない、タイプ2の装置依存型のイベントデータを取得している。装置10-1-2はその後、取得した装置依存型のペンイベントデータを、インターネットのようなネットワークを介して、インクデータ処理部100Tを具体化しているサーバに伝送する。サーバ内で実行されるインクデータ処理部100Tは、装置依存型のペンイベントデータを装置10-1-2から受信し、インクデータモデルによって定義されるインクデータ200を生成する。ペンイベントに関するコンテキスト情報(入力2)は、装置10-1-2によってインクデータ処理部100Tに提供され得るか、又は、例えばサーバによって提供されるアプリケーションが共通のペンタイプを利用する場合のようにコンテキスト情報(入力2)が必要ない場合には省略され得る。

【0448】

図46Bは、インクデータ処理部100Tが装置10-1内に存在するような、他の装置又は方法を説明する。この場合、装置10-1-1はまた、装置10-1-1を制御するためのデバイスドライバを含む。装置10-1は、筆圧データを含み又は含まない装置依存型の生データを取得する。図示した実施の形態では、装置10-1-1のセンサは、筆圧データを含む生データを取得する。装置10-1のセンサは、USBのような適切

なインターフェイスを介して、端末の I / O に対し筆圧データを含む生データを送る。端末のデバイスドライバは、受信した生データを処理し、筆圧データを含むタイプ 1 の装置依存型のペンイベントデータを製造する。装置 10-1-1 のインクデータ処理部 100 は、デバイスドライバからタイプ 1 の装置依存型のペンイベントデータを受信し、インクデータモデルによって定義される非装置依存型のインクデータ 200 を生成する。このとき、図示したように、インクデータ処理部 100 はまた、端末上で動作するアプリケーション及び／又は OS からペンイベントデータに関するコンテキスト情報（入力 2）を受信してもよい。生成されたインクデータ 200 は、バイトシーケンス、又は、ビット若しくはバイトのシーケンスの形で順序付けられ、適当なメディア（ネットワーク、ストレージ装置など）上に記録される。

10

【0449】

図 46C は、インクデータ処理部 100T が入力センサを含む端末（例えばタッチセンサ）内に存在するような、さらに他の装置又は方法を説明する。例えば、図 46C の装置 10-1-2 は、端末として機能し得る静電容量タイプのタッチセンサであり、タッチセンサに加えてインクデータ処理部 100 を含んでいる。このタッチセンサは、装置依存型の位置データ（例えば、指によってタッチされた位置）を取得する能力を有する制御 IC を含むが、ペン（又は指）の圧力データ又は他のペン特有のデータ（角度、傾きなど）を取得することができるかもしれないし、できないかもしれない。図示した実施の形態では、タッチセンサは、筆圧データを含まないタイプ 2 のペンイベントデータを取得して送信している。インクデータ処理部 100T は、センサの制御 IC からタイプ 2 の装置依存型のペンイベントデータを受信し、インクデータモデルによって定義される非装置依存型のインクデータ 200 を生成する。このとき、インクデータ処理部 100 はまた、もしあれば、装置 10-1-2 上で動作するアプリケーション又は OS から、ペンイベントデータ（入力 1）に関するコンテキスト情報（入力 2）を受信することができる。インクデータ 200 はその後、永続的な形態（SFF）又はメッセージングパケットの形態（SMF）などで適当なメディア（ネットワーク、ストレージ装置など）上に記録されるバイトシーケンス又はビットシーケンスの形で出力され得る。

20

【0450】

図 47A 及び図 47B は、本発明の実施の形態によるインクデータ再生方法の動作を説明する。

30

【0451】

図 47A は、サーバが、インクデータ処理部 100R（受信及び再生側）と、アプリケーション 300-1 と、サーバとともに動作するインクデータ処理部 100T によって生成されたあるフォーマット（例えば SFF）のインクデータ 200 が記憶され得るメモリ（RAM）とを含むような、インクデータ再生方法を説明する。インクデータ処理部 100R は、第 1 の実施の形態の図 22 に示した 100R に対応する。アプリケーション 300-1 は、アプリケーション 300-1 のために必要なデータを得るためにインクデータ処理部 100R の静的／動的なライブラリを発動する。図示した実施の形態では、アプリケーション 300-1 は、描画ディスプレイプログラムである。インクデータ処理部 100R は、RAM 内に記憶されるインクデータの中から必要なデータを選択的に取り出して再生し、再生された必要なデータをアプリケーション 300-1 に提供することができる。例えば、再生された必要なデータは位置データ及び RGB 色データを含むことができるが、可変のストローク幅データを含んでもよいし含まなくてもよい。アプリケーション 300-1 は、インクデータ処理部 100R から再生された必要なデータを受信し、補間、幾何学的形状の生成及びラスタ化などの必要な描画動作を実行し、それによってディスプレイスクリーンに（レンダリングされた）画像を出力する。

40

【0452】

図 47B は、装置 10-3 がインクデータ処理部 100R 及びアプリケーション 300-2 を含むような、他の再生方法を説明する。このケースでは、インクデータ処理部 100（図示せず）によって生成されたインクデータ 200 は、ストリーミングプロトコルに

50

従って（例えば、SMFで）メッセージ内に準備され、或いは、パケット化され、例えばネットワークに出力される。アプリケーション300-2は、インクデータ処理部100Rに対し、アプリケーション300-2のために必要なデータを所得するよう要求する。図示した実施の形態では、アプリケーション300-2は、任意のテキスト符号化スキームに従って手書きのテキストを機械読み取り可能なテキストに変換する文字画像テキスト化アプリケーションである。そのようなものとして、アプリケーション300-2は位置データ（X-Yデータ）を要求するが、RGB色データを必ずしも要求しない。また、アプリケーション300-2は、インクデータ200の著者についての情報を（例えば、特定の著者特有の手書きを区別するために）要求することができる。

【0453】

インクデータ処理部100Rは、ネットワークから受信された（SMFで書かれた）メッセージ又はパケット内のインクデータ200の中からアプリケーション300-2にとって必要なデータを取り出して再生することができる。アプリケーション300-2は、再生された必要なデータをインクデータ処理部100Rから受信し、必要なテキスト化動作を実行して、おそらくは著者IDとともにテキスト化の結果を出力する。

【0454】

上述したように、インクデータ処理部100及びインクデータ処理部100Rの機能及び動作は、各アプリケーション及び各実施の形態で使用されるメディアのタイプによっては、インターネット上の様々な装置（入力装置、端末、サーバなど）に分配され、統合され得る。

【0455】

[2] ストローク言語（SL）及びストロークファイルフォーマット（SFF）

[2-1] インクデータモデル（「ストローク言語」）（図48A～図48I）

図48A及び図48Bは、本発明の実施の形態によるインクデータ200（ストローク言語）のための情報モデルを説明する実体関連図である。

【0456】

図において、1つの箱は、この情報モデルで使用される1つのオブジェクト又はクラスを示している。例えば、オブジェクト「ストローク」、「ポイント」、及び「メタデータ」が使用される。オブジェクトは、属性データのコレクションとして表現される構造のようなデータ構造とともに、オブジェクト由来の言語によって定義されるクラスオブジェクトを含む。

【0457】

図48Bの楕円は、箱内に含まれるオブジェクトの属性を示す。例えば、属性「位置」、「色」、「半径」がデータオブジェクト「ポイント」のために使用される。属性のうち、実線の楕円で示されるものは、対応するデータオブジェクトにとって必須な属性である。例えば、属性「位置」はデータオブジェクト「ポイント」にとって必須である。破線で示される楕円は、追加され得る拡張可能な属性である。例えば、属性「色」は、データオブジェクト「ポイント」にとって単に追加し得るものであり、必須ではない。図中の菱形は、菱形によって接続されるデータオブジェクト間の関係を示している。例えば、「含まれる」と分類された菱形は、菱形によって接続されたデータオブジェクトの一方が他方のデータオブジェクトに含まれることを意味している。

【0458】

一般的に、「情報モデル」は、オブジェクト領域（ドメイン）でデータの意味を定義し、ドメイン内で使用される概念、関係、制約、ルール、及び処理を表す。情報モデルは、共有されるドメイン内のコンテキストにおいて、様々なアプリケーション及びサービスの間で信頼性の高い情報の要求を可能にする系統的な構造を提供する。インクデータ処理部100Tによって出力され、及び／又は、インクデータ処理部100Rによって再生されるインクデータモデルは、ユーザによって動かされるペンによって入力されるストロークのトレースと、ペンタイプ（鉛筆、ブラシなど）やペン色のような、トレースを入力するために使用されたペンの属性とを表現することができるよう構成される。

【0459】

図48Aは、インクデータ200のための情報モデルのサブドメインを説明する。インクデータ200のための情報モデルは、様々なデータオブジェクトの機能及び目的に基づき、概念的に、以下の4つのサブドメインにカテゴライズされる。

【0460】

(SM) ストロークモデルサブドメイン。ともにインクデータ200の位置(幾何学的形状)及びレイアウトを定義するストロークオブジェクト210及びポイントオブジェクトを含む。

【0461】

(MD) メタデータサブドメイン。主にストロークオブジェクト210のメタデータを定義する。

【0462】

(R) 画像データに対する描画スタイルオブジェクト230(主にストロークデータ(ストロークオブジェクト)を変換するために必要となる情報定義するラスター化サブドメイン)。

【0463】

(M) 操作サブドメイン。ストロークオブジェクトを削除する、分割する、動かす、コピーする、のように、動的にストロークオブジェクト又はストロークオブジェクトのインスタンスを操作するための一群のデータオブジェクトを含む。

【0464】

これらのサブドメインのそれぞれが、図48Bを追加的に参照して以下に説明されるであろう。

【0465】

<(SM) ストロークモデルサブドメイン>

ストロークモデルサブドメインは、ストロークオブジェクト210、ポイントオブジェクト、及びキャンバスオブジェクトを含む。

【0466】

(SM1) ストロークオブジェクト210は、インクデータモデルの本質的な構成要素であり、インクデータ200の核を構成する。

【0467】

図48Cは、ストロークオブジェクト210の構造を説明している。ストロークオブジェクト210は、集合的にストロークの幾何学的形状を定義する「1番目のポイント」から「n番目のポイント」までの複数のポイントオブジェクトを含む。ストロークオブジェクト210は、InkML(非特許文献1)で「トレース」、SVG1.1(非特許文献3)で「パス」、HTML5(非特許文献4)明細書で「キャンバスパス」とタグ付けされたデータに対応する。

【0468】

ストロークオブジェクト210は、オブジェクト属性として、第1の実施の形態を参照して上述したように、開始パラメータ301及び終了パラメータ303を含むことができる。属性として開始パラメータ301及び終了パラメータ303を含むストロークオブジェクト210においては、開始パラメータ301及び終了パラメータ303はポイントオブジェクトとは別に定義される。ストロークオブジェクト210は、ストロークごとに拡張可能な属性として、「色」及び「半径」(簡便のため「幅」に置き換えられ得る)の属性を含んでもよい。これらの属性については、後に詳細に記述されるであろう。

【0469】

(SM2) ポイントオブジェクトは、N個のポイントオブジェクトが1つのストロークオブジェクト210の中に含まれるようなオブジェクトである。ポイントオブジェクトは、必要な属性として、所与(例えば2D)の座標系における位置を示す「位置」を含む。ポイントオブジェクトは、拡張可能な属性として、「色」、「半径」などを含んでもよい。「色」は、ポイントごとに設定されてもよいし、ストロークごとに設定されてもよい。

10

20

30

40

50

【0470】

概念的なツリー構造の中で同じ属性が複数のデータオブジェクトに対して定義されるとき、より低いレベルのオブジェクト（例えば、ポイントオブジェクトにより近いオブジェクト）に与えられる属性に優先権が与えられる。例えば、もしストロークの色（R, G, B, 及び透明度アルファ（ α ））が4Dベクトル（R1, G1, B1, α 1）で表現され、かつ、もしストロークの3番目のポイントの色が（R2, G2, B2, α 2）と定義されるならば、3番目のポイントの色は（R2, G2, B2, α 2）と決定される。他の例としては、もし色属性が、ストロークオブジェクト210に対して適用されるシェイプフィリングラスタ化（描画スタイル）オブジェクトと、ストロークオブジェクト210自身との間で異なっていたら、ストロークオブジェクト210自身の色属性が優先権を有する。

10

【0471】

ストロークオブジェクト210又はポイントオブジェクトは、それ自身が「追加」コマンドを所有していると考えられ得る。それは、新たなストロークオブジェクト210又は新たなポイントオブジェクトが生成される都度、その新たなストロークオブジェクト210又はその新たなポイントオブジェクトをデータ構造に追加すべきことを命令する。

【0472】

（SM3）キャンバスオブジェクトは、1以上のストロークが入力されるときに使用される描画領域（「キャンバス」）のサイズを示すもので、属性として例えば幅「W」及び高さ「H」を含む。それはまた、和紙、カーボン紙、普通紙、コピー用紙、写真用紙などの紙のタイプごとにインクの吸収量を示す。紙のタイプの情報は、インクデータ200の実際の画像表現を決定するために、ラスタ化（描画スタイル）オブジェクトと結合され得る。

20

【0473】

図48Dは、キャンバスオブジェクトを説明している。左の絵は、ストロークが入力されるとき描画領域のサイズを示している。それは、幅「W1」及び高さ「H1」により定義される描画領域にストロークが入力されることを図示している。右の絵は、インクデータ200から取得されるキャンバスサイズ（W1, H1）がストロークを再生するためにどのように使われ得るかを説明している。図示した例では、ストロークが、モバイル端末のような、ストロークが描かれたときに使用された領域（W1, H1）より小さい幅「W2」及び高さ「H2」の描画領域を含む装置上で再生されることを前提としている。そのような場合、元の描画領域（W1, H1）及び元の描画領域に対するストロークの関連するサイズ／位置は、レンダリングする装置（W2, H2）の所与のキャンバス領域にフィットするようストロークを拡大し、低減（縮小）し、切り取り、又はオフセットする際に使用され得る。

30

【0474】

<（MD）メタデータサブドメイン>

メタデータサブドメインは、それぞれがストロークオブジェクト210のためのメタデータを定義する複数のメタデータオブジェクト250を含んでおり、メタデータは、そのストロークオブジェクト210を生成するために使用されたペンイベントデータに関するコンテキスト情報（入力2）を含む。

40

【0475】

（MD1）メタデータオブジェクト250は、ストロークオブジェクト210を「描写」するために、ストロークオブジェクト210と一対一の関係を有する。

【0476】

1）タイムスタンプ属性はストロークが記録された時刻の情報を含み、例えば定義されたフォーマット（32ビットの符号なし整数）で書かれたUNIX（登録商標）時刻の値を表す。図48Eは、ストロークごとにタイムスタンプ属性を使用することの効果の説明している。2つのストロークa及びbが2人の異なるユーザにより実質的に同時に記録されたとする。左の絵は、ストロークaがストロークbの後に描かれたこと、すなわち、ス

50

ストローク a のタイムスタンプ値がストローク b のタイムスタンプ値より大きいことを示している。右の絵は、ストローク a がストローク b の前に描かれたこと、すなわち、ストローク a のタイムスタンプ値がストローク b のタイムスタンプ値より小さいことを示している。例えば、異なるストロークにそれぞれ関連付けられた異なるタイムスタンプ値に基づいてどのストロークが各交点で他のストロークの上に位置するべきかを決定することで、複数のユーザによって入力された複数のストロークを正確にレンダリングすることが可能になる。

【0477】

2) 著者属性は、ストロークを記録した著者を特定する情報を含む。

【0478】

3) ペン ID 属性は、ストロークを記録するために使用されたペンを特定する情報である。理想的には、ID は各ペンに対してグローバルにユニークである。ペン ID が利用可能でない、又は、ペン ID が強固に確立される必要があるときには、ペン ID は、ペンによってなされた入力を検出するために使用されたセンサのセンサ側 ID に関連して使用され得る。

【0479】

上述した属性を用いることで、ストロークのメタデータは、限定されない例として、いつ、誰によって、どの特定のペンを用いてストロークが描かれたか、を記述することができる。

【0480】

< (R) 描画スタイルオブジェクト (ラスター化) サブドメイン >

描画スタイルオブジェクト (ラスター化) サブドメインは、ストロークが入力されたときにストロークオブジェクト 210 に何の修飾する又は修正する処理が関連付けられたかをそれぞれ保持する一群の描画スタイルオブジェクト 230 を含む。描画スタイルオブジェクト 230 は、どのストロークオブジェクト 210 が生成されたかに基づいて、ペンイベントデータに関するコンテキスト情報 (入力 2) から構築される。例えば、描画スタイルオブジェクト 230 は、異なる描画ツール属性 (ブラシ、鉛筆、クレヨンなど) 及び異なるペン先幅のようなストロークに関連付けられる (適用される) 様々な修飾又は修正処理を記録する。以下のオブジェクト (「描画スタイルオブジェクト」と総称される) は、ラスター化サブドメインの一部となり得る。

【0481】

(R1) ラスター化スタイルオブジェクトは、ストロークオブジェクト 210 に対して M : N の割合の関係を有する「レンダリング」オブジェクトである。例えば、M 個 (例えば 5 個) の (拡張可能なオブジェクトを含む) スタイルオブジェクトが、N 個 (例えば 20 個) のストロークオブジェクトに適用され得る。スタイルオブジェクトはいわゆるスーパークラスオブジェクトであり、その属性は、シェイプフィリングスタイルオブジェクトのような他のオブジェクトクラスによって、拡張された関係の中で受け継がれ得る。スタイルオブジェクトは、属性として「合成」を含む。「合成」は、ストロークを以前に生成されたストローク又は背景と混ぜ合わせるときに、どのタイプの機能 (例えば、通常、複数、現在及び以前のストロークの中の最小又は最大、消去など) が使われるべきかを示す。図 48 F は、ストロークオブジェクト 210 を修飾又は修正するためにスタイルオブジェクトを拡張する代表的なオブジェクトの 3 つの概念的な例、すなわちシェイプフィリングスタイルオブジェクト、粒子散乱スタイルオブジェクト、及び領域複製スタイルオブジェクトを含む。

【0482】

(R2) シェイプフィリングスタイルオブジェクトは、図 48 F の一番上に図示されるように、ストロークオブジェクト 210 がスクリーン上にレンダリング (描画) されるときに、ストロークのアウトライン及び色を定義するためにストロークオブジェクト 210 に適用される。シェイプフィリングスタイルオブジェクトは、ストロークオブジェクト 210 を構成する複数のポイントオブジェクトに関連してそれぞれ位置取りされる複数の円

10

20

30

40

50

を定義する。各円は、各ポイントオブジェクトが半径及び色属性に関連付けられていない場合に、半径及び色属性に関連付けられ得る。ポイントオブジェクト又は複数のポイントオブジェクトからなるストロークオブジェクト210を定義するための半径及び色属性の使用は本発明の特徴の1つであり、後に詳細に記述される。シェイプフィリングスタイルオブジェクトはまた、オーバーラップする一連の円の包絡線によって定義されるストロークのアウトラインのエッジを視覚的に除去するために何のアルゴリズムが使用されるべきかを定義するアンチエイリアス属性を含む。

【0483】

(R3) 粒子散乱スタイルオブジェクトは、図48Fの中ほどに図示されるように、上述したシェイプフィリングスタイルオブジェクトで使用される円に代えて「粒子」を用いる。粒子散乱スタイルオブジェクトは、上述したシェイプフィリングスタイルオブジェクトと同様、「半径」及び「色」の属性を含む。

10

【0484】

粒子散乱スタイルオブジェクトはまた、属性「乱数シード」(図48B参照)を含む。これは整数値であり、鉛筆やウォーターカラーブラシのようなツールのために「荒さ」や「飛散」をシミュレートとする目的で疑似乱数を生成するために使用される。「乱数シード」属性は、ユーザがファイルを開け、又はリモートユーザがインクデータを受信するときに、毎回同じ絵を正確にレンダリングすることを可能にするために、ファイルフォーマット内に格納される。

【0485】

「シェイプ回転」属性は、各粒子がランダムな回転角で回転されるべきこと、又は、ある軌道に沿って回転されるべきことを示す。

20

【0486】

「間隔」属性は、2つの連続する粒子の間の距離を示す。

【0487】

属性「範囲」は、図48Fに矢印で示すように、軌道方向と垂直な方向における各粒子のオフセット値を示す。矢印で定義される幅の範囲内で粒子の位置が、乱数シードに基づいて生成される乱数に基づいて、ランダムにオフセット(変更)され得る。

【0488】

属性「フィル」は、ハッチングのような、シェイプに適用されるべきテクスチャを定義する。

30

【0489】

属性「フィルオフセット」は、シェイプに適用されるハッチングのようなテクスチャの切り抜き位置を示し、同じテクスチャが連続して適用されるのを避けるべく切り抜き位置を定義し変更するために使用される。

【0490】

属性「ビルドアップ」は、ペンが連続的な期間の間固定ポイントに位置しているときに、時間の経過に応じて生成される粒子の数が増加されるか否かを示す。

【0491】

属性「ミキシング」は、2つの連続する粒子が互いに重なる位置の色を計算するためにどのタイプ(例えば、通常、複数、なし、など)の機能が使用されるべきかを定義する。例えば、色はそれらの粒子のうちの1つの色と同じとされるべきか、それともより濃い色にされるべきか(例えば、2つの色の混合)を定義することができる。

40

【0492】

(R4) 領域複製スタイルオブジェクトは、図48の一番下に図示されるように、スタイルオブジェクトを拡張するために使用される。領域複製スタイルオブジェクトは、ポイントの間を補間する閉じた曲線によって定義される領域を設定する。属性「変換マトリックス」は、閉じた曲線に囲まれた領域内で内容に適用されるべきアフィン(affine)変換マトリックスを保持する。その領域の内容は、変換マトリックスに基づいて、異なる座標にレンダリングされる。マトリックスに基づく変形は、ストロークオブジェクト210のレ

50

ンダリングスタイルのみに影響し、ストロークオブジェクト 210 自身を操作したり修正したりすることはない。

【0493】

＜(M) 操作サブドメイン＞

操作サブドメインは、インクデータモデルに従って生成された既存のストロークオブジェクト 210 の全部又は一部を操作する（分割する、消去する、コピーアンドペーストする、修正する、など）ための変換（又は操作オブジェクト）を定義する。

【0494】

(M1) スライスオブジェクト 274 は、ストロークオブジェクト 210 の 1 つ又は 2 つの部分を取り出すために適用される操作オブジェクト 270 である。図 48 G は、スライスオブジェクト 274 の動作を説明している。図 48 G の上の絵は、スライスオブジェクト 274 が適用される前のストロークオブジェクト 210 を図示している。スライスオブジェクト 274 は、「幅」を持ちストロークオブジェクト 210 を横切る（と交差する）他のストロークによって表される。「幅」を持つスライスオブジェクト 274 は、典型的には「消去」機能と関連付けられる。「幅」を持つスライスオブジェクト 274 が交差するストロークオブジェクト 210 上の位置はそれぞれ、 k 番目のポイントと $k+1$ 番目のポイントの間と、 $k+2$ 番目のポイントと $k+3$ 番目のポイントとに位置する。元のストローク曲線は各ポイントをキャットマルローム曲線で補間することによって生成されており、元のストロークの形を修正することなくそのストロークを 2 つのストロークに切断するためにストロークがスライスされるとき、新たに生成された終了点のために新たな終了ポイントオブジェクトが生成されることはない。その代わり、図 48 G の下の図に示すように、新たな終了パラメータ 303 n のための値が 1 番目のストロークのための属性として設定され、新たな開始パラメータ 301 n のための値が 2 番目のストロークのための属性として設定される。新たな終了パラメータ 303 n 及び開始パラメータ 301 n は、最初にディスプレイされたセグメント又は最後にディスプレイされたセグメントを定義する 2 つのポイント間の内部分割ポイントを示す 1 つ又はいくつかの「float」ポイントとして表現される。例えば、最初のストロークのための終了パラメータ 303 n は、元の k 番目のポイントと元の $k+1$ 番目のポイントの間の内部分割点として定義され得る。したがって、元のストロークの形状が、2 つの新たに生成されたストロークの形状を表すために使用され得る。いくつかの実施の形態では、図 48 G の「幅」に沿って延在するスライスされた（削除された）ストローク部分は、元のストロークから分割された「3 番目の」ストロークとして表現され得る。

【0495】

上述した方法によれば、元のストロークの 2 つのポイント間の内部分割ポイントは、新たに生成された（スライスされた）ストロークの属性（新たな終了パラメータ 303 n 及び新たな開始パラメータ 301 n）として保持される。そういうものとして、新しいスライス動作の結果として新たなポイントオブジェクトが生成されることはなく、「入力ポイント」の元のコレクションは変更されない。したがって、補間されたキャットマルローム曲線から曲線が導出されるとき、曲線のアウトラインはスライス操作の前と後とで変化しない。

【0496】

属性「変更スタイル」及び「変更メタデータ」は、スライス動作によって元のストロークから分けられた複数のストロークの属性が、分けられたストロークのために変更される（新たに定義される）か（例えば、「著者」属性）、又は、元のストロークに関連付けられた属性から変更されず同じものとなるか（例えば、ペン色属性）を示す。

【0497】

開始及び終了パラメータ 301 及び 303 はそれぞれ、元のストロークから分けられた 2 つのストロークの開始及び終了位置を示す属性である。

【0498】

属性「ポイント範囲」は、図 48 G の「幅」により定義される横断ストローク（スライ

10

20

30

40

50

シング操作オブジェクト)が操作対象のストロークオブジェクトと交差するポイントの範囲を定義する。図48Gでは、ポイント範囲は $k+1$ 番目のポイントと $k+2$ 番目のポイントとを含む。

【0499】

図48Hは、スライシング操作オブジェクトを適用することによって実現される「消去」機能を説明している。そのような「消去する」操作オブジェクトが既存のストロークオブジェクト210に適用されるとき、ストロークオブジェクト210は、中間に「消去」される正確な部分(例えば、図18のホールセグメント1801内の $P_intersect_L$ と $P_intersect_R$ との間)を有して、新たな終了パラメータ303nを有する第1のストローク(ストローク1)と、新たな終了パラメータ303nを有する第2のストローク(ストローク2)との2つのストロークに分割される。この場合、元の単一のストロークオブジェクト210を修正するためのこの操作がファイナライズ(コミット)されるとき、2つの新たなストローク(ストローク1及びストローク2)が生成され、中間部分は消去される。

【0500】

(M2)選択オブジェクトは、図48Iに示すように、複数(N個)のスライスオブジェクト274(スライシング操作オブジェクト)を「含む」操作オブジェクトである。選択オブジェクトは、複数(N個)のスライスオブジェクト274(スライス1、スライス2、スライス3)で囲まれた領域(スライス2)を「選択」し、それによって選択された範囲内の既存のストロークオブジェクト210の任意の部分がその後、その部分に変換マトリックスを適用することによって、操作され(移動され、コピーされ、拡大され、縮小され)得る。図48Iでは、点線の閉じた曲線で形作られる投げ縄で示される選択範囲が、部分的にスライスされたストロークオブジェクト210を含み、これはその後、操作されることができる。一例では、図48Iに図示したように、移動され(移され)ることができる。

【0501】

属性「変換マトリックス」は、アフィン(affine)変換マトリックスである。図示した例は、 tx 及び ty によって説明される領域内の値を示している。マトリックスオブジェクトが特定の操作を定義するために使用されるとき、まるで実質的に異なる位置に移動したかのように、領域(スライス2)内のポイントを表示することが可能となる。このケースでは、この操作オブジェクトが元の単一のストロークオブジェクト210を修正するためにファイナライズ(コミット)されるとき、3つの新たなストローク(ストローク1、ストローク2、及びストローク3)が生成される。

【0502】

属性「複製」は、オブジェクトを異なる位置へ実質的に移動するためにアフィン変換マトリックスを適用した後であっても、元の(変形前の)位置でオブジェクトを保持するか否かを示す。例えば、実質的に移動した新たな位置／領域に加えて元の位置／領域を保持することにより、元の位置／領域が新たな位置／領域にコピーされるというコピー動作を実現することが可能になる。

【0503】

[2-2]ストロークファイルフォーマットのサンプル(図48J～図48L)

図48Jは、本発明の実施の形態による、インクデータ構造がシリアル化されて維持されるストロークファイルフォーマット(SFF)と同様に、図48Bに示されるインクデータモデルの定義に従い、インクデータ処理部100(100T又は100R)によって生成され処理されるインクデータの構造(オブジェクトツリー)を説明する。

【0504】

図48Jの上部は、メモリスペースのようなその内部処理リソース内でインクデータ処理部100によって生成されるインクデータ構造を説明している。例えば、 i 個のストロークオブジェクトのそれぞれの中で、1以上のポイント(ポイント#1～# j)が例として挙げられ、1つのメタデータオブジェクト250と、1つの描画スタイルオブジェクト

230とが互いに関連付けて（インスタンスツリーの形で）定義される。データ構造は、図48Bの実体関連（ER）図に示されるように、インクデータモデルの定義によるものである。図48Jでは、ストローク#1とストローク#2は、それらのスタイルオブジェクトのそれぞれにおける相当多数のデータは同じであるけれども、異なる名前の付けられたスタイルオブジェクト（スタイル#1及びスタイル#2）を含む。このようなことは、例えば、ストローク#1とストローク#2とが同じ（同じペン先タイプ及びペン色を有する）描画ツールを用いて同じ描画アプリケーション内で描かれているときに発生する。

【0505】

図48Jの下部は、図48Jの上部に示したデータ構造が配置されるストロークファイルフォーマット（SFF）のファイルの例を説明する。すなわち、各ストロークオブジェクト#1～#iのために、ストロークオブジェクト210を構成するポイントオブジェクト#1～#jと、ストロークオブジェクト210を描写するメタデータオブジェクト250と、ストロークオブジェクト210がスクリーン上にどのようにレンダリング（描画）されるかを定義する描画スタイルオブジェクト230とがSFFファイル内に定義される。示されるように、「InkDataFile」と名付けられた1つのファイルは、複数のストローク#1～#iについての情報を含む。

【0506】

<ストローク#1についての情報>

（1）第一に、ストローク#1自身に関する情報が記述される。この情報は、ストローク#1の開始パラメータ301及び終了パラメータ303のようなストローク#1の属性がその中に符号化されるバイトシーケンス（又はバイナリシーケンス。ここでは交換できるように使用される）を含む。この情報はまた、ストローク#1を構成する一群のポイントオブジェクト#1～#jがその中で符号化されるバイトシーケンスを含む。

【0507】

（2）第二に、ストローク#1を「描写する」メタデータオブジェクト250がその中に符号化されるバイトシーケンスが含まれる。

【0508】

（3）第三に、ストローク#1を「レンダリングする」描画スタイルオブジェクト230がその中に符号化されるバイトシーケンスが含まれる。

【0509】

<ストローク#2についての情報>

上記ストローク#1についての場合と同様、同じ整形処理がストローク#2についても実行される。図48Jの例では、描画スタイルオブジェクト#1の値は、描画スタイルオブジェクト#2の値と同じである。この場合、持続的なファイルフォーマット（SFF）においては、ファイルスペースを節約し、冗長性を避けるために、同じ値を繰り返さないことが好適であり得る。したがって、描画スタイルオブジェクト#2として同じ値が繰り返されることはなく、示されるように、メタデータオブジェクト#2の後には描画スタイルオブジェクトが含まれない。再生側では、インクデータ処理部100は、新しい（異なる）描画スタイルオブジェクト230が見つかるまで、各ストロークオブジェクト210をレンダリングするために同じ描画スタイルオブジェクト230を使い続けるであろう。すなわち、ファイル生成側及び再生側では、ストロークオブジェクト210のためのスタイルオブジェクトが省略されている場合、前のストロークオブジェクト210のために使用したスタイルオブジェクトが使用されるべきことが予め同意されている。

【0510】

図48Kは、データオブジェクト（メタデータオブジェクト250、キャンバスオブジェクト、描画スタイルオブジェクト230）の各クラスがSFFファイル内に順序付けられて記憶されるとき各クラスのデータタイプを説明する。図48Kでは、「InkDataFile」がデータ構造のルートに位置するオブジェクトである。

【0511】

図48Kでは、各ブロックの最初の行が本発明のインクデータモデルによって定義され

るオブジェクトの名前を示している。各ブロックの2行目以降は、左側のカラムにおいて名付けられたオブジェクトの属性を示し、右側のカラムにおいてそれらのデータタイプ（整数、符号なし文字列、など）。属性は、ASN.1、BER、DERといった符号化方法、又は、「sint32型」、「float」のような第1の実施の形態の図10に示したスキーマファイル内に示した符号化方法などの適切な方法を用いて符号化され得る。

【0512】

図48Kに説明される属性の意味は、いくつかのストロークオブジェクトが与えられたSFFファイル内に含まれるかを示す「strokeCount」のようなインクデータ200を持続させる目的で必要となる情報を除き、図48Bを参照して上述した属性と同じである。

【0513】

構造のルートにあるデータオブジェクトInkDataFileは、本発明の実施の形態によるファイル内に維持されるバイナリシーケンスを含む。

【0514】

1行目の「Header」はInkDataFileを持続的に記憶する目的で必要となるデータであり、後続するバイナリシーケンスを処理するうえでの使用のために、インクデータモデルの定義のバージョンに関する情報などを含む。ヘッダはさらに、インクデータの生成側が意図したようにインクデータ200を再生するためにアプリケーション300又は再生装置100Rが必要とする、どのデータが圧縮されているか或いはいないか、使用されるバイナリ符号化方法、及び、その他の情報のような情報を含む。

【0515】

2行目の「DecimalPrecision」は位置座標値のような値の正確さのレベルを示し、記録フォーマット内で十進数により表現される。「DecimalPrecision」は、上述した第1の実施の形態の図10の4行目のパラメータに対応する。「DecimalPrecision」は、インクデータモデル内では定義されていないが、インクデータファイルを維持する目的のために必要となるひとまとまりのデータである。「DecimalPrecision」の使用は、本発明の特徴の1つであり、以下で詳細に記述される。decimalPrecision数（又は正確度）は、解像度の逆数に対応する。例えば、位置座標値が0.01単位（例えば、0.01ピクセル）の解像度で取得可能であるとき、decimalPrecisionとして表現される解像度は、100（0.01の逆数）として設定され得る。より最適な保管の目的で、decimalPrecision値は、指数の形で表現され得る。例えば、値100は、底10の2乗として表現され得る（いくつかのケースでは、底は省略されてもよい）。本発明の実施の形態によるインクデータ生成及び再生方法の中で正確度を用いることは、図58及び図66を参照して後に完全に記述されるであろう。

【0516】

3行目の「Metadata」は、図48B及び図48Eを参照して上で説明したメタデータオブジェクト250に対応する。

【0517】

4行目の「Canvas」は、図48B及び図48Dを参照して上で説明したキャンバスオブジェクトに対応する。

【0518】

5行目の「Style」は、図48B及び図48Fを参照して上で説明した（描画）スタイルオブジェクトに対応する。

【0519】

6行目の「strokeCount」は、インクデータファイルを維持する目的で必要となるひとまとまりの情報であり、特定のインクデータファイル内に含まれるストロークの数を示す符号又は整数値である。そのようなものとして、この属性はインクデータモデル自身には含まれない。典型的には、strokeCountはインクデータ符号シーケンスの中に、それらが動的に修正されるものではなく、その代わりに静的なストレージ型

のメディアに出力されるものであるときに、追加される。`strokeCount`は、例えばインクデータがリモートに位置するインクデータ再生装置のためにリアルタイム通信で継続的に出力されるものであるときに、設定されず、又は何らの値も含まない(`NULL`)。

【0520】

7行目の「`strokes`」は、インクデータ内に含まれる一群のオブジェクトインスタンス「`strokes`」(ストロークオブジェクト)に関連する。各「`stroke`」(各ストロークオブジェクト210)は、図48Lに記述されるように、1以上のポイントオブジェクトと様々な属性とを含む。

【0521】

図48Lは、1番～N(`strokeCount`)番のストロークオブジェクトである各ストロークの中に含まれる情報を説明する。1行目と2行目の「`Start_value`」及び「`End_value`」は、図48Cを参照して上述した開始パラメータ301及び終了パラメータ303に対応する。第1の実施の形態を参照して上述したように、これら2つのパラメータは、ストロークオブジェクト210の属性として、そのストロークオブジェクト210に含まれるポイントオブジェクトとは別に記憶される。

【0522】

4行目の「`StrokeType variableAlpha`」は、ストロークに関連付けられる透明度(アルファ)がストロークの長さに沿って可変かを示す。「`variableAlpha`」は、ストロークオブジェクト210の透明度がその長さに沿って可変であることが許容されているか否か(すなわち、固定されているか)を示すフラグであり、典型的にはブール値として表現される(`TRUE`又は`FALSE`)。いくつかの実施の形態では、属性「`StrokeType variableAlpha`」は図48Bに示したインクデータモデル自身の中には存在せず、ストロークオブジェクトがSFFファイル(インクデータファイル)内に維持されるときに使用される。本発明の様々な実施の形態で「`variableAlpha`」を使用することが、後に詳細に記述されるであろう。

【0523】

5番目から8番目の値「`alpha`」、「`red`」、「`green`」、及び「`blue`」は、連帯して、ストロークオブジェクト210の「色」属性を構成する。

【0524】

5行目の「`alpha`」は、例えば「`variableAlpha=false`」により示されるようにストロークの「`alpha`」が可変でない場合に使用される固定された透明度の値(又は不透明度/インクの濃度の値)を示す。「`alpha`」が固定されているとき、固定された「`alpha`」がストロークオブジェクト210の長さに沿って、すなわち、ストロークオブジェクト210を構成する複数のポイントオブジェクトのそれぞれに対して適用される。

【0525】

6行目から8行目の「`red`」、「`green`」、及び「`blue`」は、RGB色空間が使用されるときに、ストロークオブジェクト210の色データを決定するための情報である。

【0526】

9行目の「`pointsCount`」は、ストロークオブジェクト210内に含まれるポイントの数を示す。上述した「`strokesCount`」と同様、「`pointsCount`」は、SFFファイル内のインクデータファイルを維持する目的で使用されるもので、インクデータ200が(より一時的なリアルタイム通信型のメディアではなく)静的なストレージ型のメディアに出力されるタイミングで決定され得る。「`pointsCount`」は、例えばインクデータ200が代わりにリアルタイム型のメディアに出力されるときには、データ構造内に存在せず、又は何らの値も含まない(`NULL`)こととしてもよい。

10

20

30

40

50

【0527】

10行目の「points」は、ストロークオブジェクト210を構成する複数のポイントオブジェクトのそれぞれに関する情報を示す。

【0528】

ポイントオブジェクト（図48Lの「point」）は、処理されるストロークに含まれる1番～N（pointsCount）番のポイントのそれぞれに関する情報を含むデータオブジェクトである。図48Bに示すように、1つのストロークオブジェクト210内にN（pointsCount）個のポイントオブジェクトが含まれる。

【0529】

「point」を定義する箱の1行目の「if」フレーズは、ストロークが長さ方向の可変アルファを含むか否かを示すvariableAlphaの値（TRUE又はFALSE）に応じた各「point」データオブジェクトのための構文変更を意味する。

10

【0530】

(i) 2行目～6行目は、variableAlphaの値がTRUEである（すなわち、ストロークが長さ方向の可変アルファ値を含む）場合に、「point」データオブジェクトに含まれるデータを示す。

【0531】

2行目の「x」は、処理されるポイントの最初の座標値を示す。

【0532】

2行目の「y」は、処理されるポイントの2番目の座標値を示す。

20

【0533】

したがって、「x」と「y」は、連帯してポイントの2D座標を定義する。

【0534】

4行目の「radius」は、中心としてポイントを含む円の半径のサイズを示す。「radius」は、ポイントオブジェクトに関連付けられる属性であり、筆圧値又はペン先の力に関連付けられる必要はない。むしろ「radius」は、後により完全に記述されるように、圧力や力の概念のようなより低レベルな概念を包含する、意味を有する一般化されたより高度な概念である。

【0535】

5行目の「alpha」は、ポイントオブジェクトに関連付けられたアルファ値を示す。

30

【0536】

ポイントオブジェクトの2行目から5行目に含まれるデータのデータ型が、図58を参照しながら後により完全に記述するように、描画処理アプリケーションで典型的に使用される浮動小数点（float）データ型でなく、整数（int）又はushort（又はuchar）であることに注意すべきである。

【0537】

(ii) 7行目～11行目は、variableAlphaの値がFALSEである（すなわち、ストロークが可変アルファ値を含まない。すなわち、アルファ（透明度）の値がストロークの長さに対して固定されている）場合に、「point」データオブジェクトに含まれるデータを示す。

40

【0538】

7行目の「x」は、ポイントの最初の座標値を示す（上述した2行目と同じ）。

【0539】

8行目の「y」は、ポイントの2番目の座標値を示す（上述した3行目と同じ）。

【0540】

したがって、「x」と「y」は、連帯してポイントの2D座標を定義する。

【0541】

9行目の「radius」は、中心としてポイントを含む円の半径のサイズを示す（上述した4行目と同じ）。

50

【0542】

このケースではストロークオブジェクト210が、ストロークオブジェクト210を構成する複数のポイントオブジェクトのそれぞれに適用されるべき固定されたアルファ値を有していることから、アルファ値はポイントオブジェクトのためには定義されない。

【0543】

上述したように、また、図57を参照して後により完全に記述するように、データオブジェクトの構文は、ストロークオブジェクト210が長さに沿って可変なアルファ値を有するか又は固定されたアルファ値を有するかを示すvariableAlphaの値(TRUE又はFALSE)に応じて変化する。

【0544】

12行目及び後続する行は、各「point」データオブジェクトについて、reserveRawFlagの値に応じて、選択的に追加の属性を含むべく構文が変化することを示す。reserveRawFlagがTRUEである場合、例えば各ポイントについて「タイムスタンプ」情報を含むように、元の情報を失うことなく、データオブジェクトを拡張する（拡充する）ことが可能である。

【0545】

インクデータモデルは、本発明の実施の形態によるインクデータモデルの構文及び意味論で定義されたストロークファイルフォーマット(SFF)で書かれたインクデータファイル内のデータオブジェクトとともに、図48A～図48Lを参照して上述された。次に、本発明の実施の形態によるデータ構造を有するインクデータ200を生成して出力するように動作可能であるインクデータ処理部100(100T)が記述される。

【0546】

[3] インクデータ処理部(図49～図63)

図49は本発明の実施の形態によるインクデータ処理部100Tの機能ブロック図である。このインクデータ処理部100Tは、図5の100Tに対応する。インクデータ処理部100Tは、上記図48A～図48Lに記述されるもののような、入力装置によって提供されるペンイベント及びペンイベントのコンテキスト情報(入力2)に基づくインクデータモデルの定義に従ってインクデータを生成する。インクデータ処理部100Tは、生成したインクデータ200を、例えば、上記図48J～図48Lに記述したSFFファイル(「インクデータファイル」)、バイナリシーケンス、パケットなどの形で出力する。

【0547】

インクデータ処理部100Tは、インクデータ生成部120及びインクデータ整形部140を含む。インクデータ生成部120は、図7のストロークデータオブジェクト操作部122に対応する。インクデータ生成部120は、筆圧データを含むタイプ1のペンイベントデータ及び筆圧データを含まないタイプ2のペンイベントデータのような様々なタイプの装置依存型の入力データ(「入力1」)を受信する。図49では、タイプ1のペンイベントデータは、タイムスタンプ情報(例えば「double_timestmap」)、複数組のXY座標(例えば「float x, y」)、及び筆圧データ(例えば「float pressure」)を含み、タイプ2のペンイベントデータは、タイムスタンプ情報(例えば「double_timestmap」)及び複数組のXY座標(例えば「float x, y」)を含む。インクデータ生成部120は、タイムスタンプ情報を受信する代わりに、例えばペンイベントデータを受信する時刻を、タイムスタンプ情報として使用することとしてもよい。

【0548】

インクデータ生成部120はまた、ペンイベントデータを入力するために用いたアプリケーション又はオペレーティングシステムから、ペンイベントデータについてのコンテキスト情報(「入力2」)を受信する。例えば、描画アプリケーションを用いて描かれたストロークの場合、コンテキスト情報(「入力2」)は、描画アプリケーションによってストロークを描くために設定される様々なパラメータを含むことができる。すなわち、コンテキスト情報(入力2)は、ストロークを生成(描画)する目的のために定義される構成

情報を含むことができる。コンテキスト情報（入力２）は、もしタイプ１のペンイベントデータとタイプ２のペンイベントデータとが２つの異なるアプリケーションによって生成されるのであれば、その２つのアプリケーションからのタイプ１及びタイプ２のペンイベントデータそれぞれのために提供され得る。或いは、タイプ１及びタイプ２のペンイベントデータのためのコンテキスト情報（入力２）は、そのタイプ１及びタイプ２のペンイベントデータを生成するために共通に使用される同じアプリケーション又はオペレーティングシステムによって提供されてもよい。

【０５４９】

第１の実施の形態で図７のストロークデータオブジェクト操作部１２２を参照して上述したように、インクデータ生成部１２０は、受信したタイプ１又はタイプ２のペンイベントデータ及び受信したコンテキスト情報（入力２）に基づき、それぞれがポイントに関するXY座標（位置）及び半径及びアルファデータを含む一連のポイントオブジェクトを出力する。図４８Ｌのデータオブジェクト「point」を参照して上述したように、半径はポイントに関連付けられる属性であり、圧力又はペン先の力に関連付けられる必要はないが、本発明の実施の形態によるインクデータモデルの定義によれば、むしろ、圧力やペン先の力より低いレベルの概念を包含する、一般化された、装置に依存しない、高度な、意味を有する概念である。

【０５５０】

インクデータ整形部１４０は、ポイントのXY座標及び半径及びアルファデータを含むポイントごとにデータを受信し、入力されたデータを、例えば図４８Ｌに記述したポイントオブジェクトのデータ構造に対応するデータ構造に整形し、整形したデータを出力する。整形されたポイントオブジェクト内のデータは、描画処理アプリケーションで典型的に使用されるfloat型のポイントデータ型（「float」）ではなく、「int」（整数）、「ushort」、「uchar」などのデータ型のものとなる。

【０５５１】

図５０Ａは、図４９のインクデータ生成部１２０によって実行される、本発明の実施の形態によるポイントオブジェクトの属性として半径及びアルファ情報を出力する処理を説明するフロー図である。典型的には、インクデータ生成部１２０はそれが受信するXY座標データをインクデータ整形部１４０に単にパスするに過ぎないので、XY座標データ（位置（X，Y））を出力することの記述は省略する。

【０５５２】

ステップＳ１２０１で、インクデータ生成部１２０は、処理されるべきポイントを含むストロークごとに、必要なコンテキスト情報（入力２）を取得する。コンテキスト情報（入力２）のサンプルが、図５０Ｂを参照して後述されるであろう。

【０５５３】

ステップＳ１２０２では、入力されたデータが筆圧データを含むか否かが決定される。

【０５５４】

ステップＳ１２０３では、入力されたデータが筆圧データを含むとステップＳ１２０２で決定された（「TRUE」）後、ポイントの筆圧データがその点でのパラメータ（vn・・・速度）を導出するために使用され得る。より大きな筆圧があるポイントに印加されるとき、そのポイントでの速度が遅くなると認められる。したがって、vnは一般に圧力データと反比例の関係にあり得る。

【０５５５】

ステップＳ１２０４では、入力されたデータが筆圧データを含まないとステップＳ１２０２で決定された（「FALSE」）後、そのポイントの時刻情報が取得される。時刻情報は、ストロークを構成するポイントごとの入力情報として受信されてもよいし、または、インクデータ生成部１２０（又はインクデータ処理部１００Ｔ）がそのポイント情報を受信する時刻として設定されてもよい。

【０５５６】

ステップＳ１２０５では、図５１を参照して後により完全に記述するように、ポイント

及び隣接する 1 又は複数のポイントの時刻情報に基づいて、そのポイントの速度が導出される。

【0557】

ステップ S 1 2 0 7 では、図 5 2 を参照して後により完全に記述するように、 v_n （速度）、フェイズ情報、及びペンタイプ情報に基づき、ポイントの半径情報が取得される。

【0558】

ステップ S 1 2 0 9 では、図 5 5 を参照して後により完全に記述するように、 v_n （速度）、フェイズ情報、及びペンタイプ情報に基づき、ポイントのアルファ（透明度又は不透明度）情報が取得される。

【0559】

図 5 0 B は、インクデータ生成部 1 2 0 に関連するコンテキスト情報（入力 2）のいくつかを設定するために使用され得るアプリケーション又はオペレーティングシステムの G U I のサンプルを説明する。インクデータ生成部 1 2 0 に対して提供されるコンテキスト情報（入力 2）は、例えば、最大及び最小の半径値（「C n t x 1」）、図 5 1 を参照して後述するように最大及び最小の半径値に関連付けられた最大及び最小の速度値（「C n t x 2」）、半径又はアルファ値を導出するために使用される機能（「C n t x 3」）、ストロークの始まり及び終わりのフェイズにあるポイントのために設定され得る例外的な値（図 5 3 参照）（「C n t x 4」）、及び、ペンタイプ情報（図示せず）を含み得る。コンテキスト情報（入力 2）は、インクデータ生成部 1 2 0 のために予め定義されていてもよいし、図 5 0 B に示すように、設定 G U I を介してユーザにより明示的に設定されるものとしてもよい。

【0560】

図 5 1 は、本発明の実施の形態による、図 5 0 A のステップ S 1 2 0 5 において時刻情報に基づいて速度を導き出す処理を説明する図である。

【0561】

速度（V e l o c i t y）は、距離を時間によって除することによって導出される。除算の分母は、現在の点座標が取得された時刻と、以前の点座標が取得された時刻との間の差分であり得る。除算の分子は、現在の点座標と以前の点座標との間の差分（距離）であり得る。サンプリングレートが固定されているとき（分母が固定されているとき）、現在の点座標と以前の点座標との間の変位が速度を示すものとして使用され得る。

【0562】

本発明の実施の形態において、速度は、0. 0 の最小値と 1. 0 の最大値とを有するパラメータ値（ v_n ）として、すなわち、最小と最大が規格化された速度値として出力される。最小値及び最大値は、図 5 0 B の「C n t x 2」に設定され得る。

【0563】

図 5 2 は、図 5 0 A のステップ S 1 2 0 7 において、ポイントオブジェクトの属性である半径を導出する処理を説明するフロー図である。一般に半径は、上述したステップ S 1 2 0 5 で計算された規格化された速度値であるパラメータ v_n から導出される。又は、本発明の実施の形態によるステップ S 1 2 0 3 の筆圧データから導出されることも可能である。

【0564】

ステップ S 1 2 0 7 _ 0 1 では、現在ポイントの「フェイズ」が判定される。本発明の実施の形態によるストローク内のポイントの「フェイズ」の定義を説明する図 5 3 をさらに参照すると、フェイズは現在のポイントがストローク内でどこ（どの位置）に位置するかを示す概念である。例えば、ストロークの最初のポイントは始まりのフェイズにあり、ストロークの最後のポイントは終わりのフェイズにある。最初のポイントと最後のポイントの間のポイントは中間のフェイズにある。各ポイントのフェイズは、ペンダウン、ペンムーブ、ペンアップの各タイプのような各ペンイベントタイプのために判定され得る。ペンダウンイベントは、ユーザがポインタ（ペン、指など）を用いてストロークを描画し始めるイベントであり、ペンアップイベントは、ユーザがポインタを用いるストロークの描

画を終えるイベントであり、ペンムーブイベントは、ペンダウンイベントとペンアップイベントの間に生ずるイベントである。

【0565】

ステップS1207_1において、処理対象のポイントが始まり又は終わりのフェイズにある、すなわちそのポイントがストロークの最初のポイント又は最後のポイントであると判定された後のステップS1207_4では、1以上のポイントがストロークの最初（最初のポイントの前）及びストロークの最後（最後のポイントの後ろ）に付加され得る。

【0566】

ステップS1207_2では、ストロークの最初又は最後に追加したポイントのそれぞれのために、図72を参照して後により完全に記述するように、0である半径や通常の半径により大きい半径（例えば2倍）のような図50Bの「Cntx4」に設定された例外的な半径の値が、そのポイントの半径として設定される。

10

【0567】

ステップS1207_1において、処理対象のポイントが中間のフェイズにある、すなわちそのポイントがストロークの最初のポイント及び最後のポイントのいずれでもない判定された後のステップS1207_03では、ペンイベントデータを入力するために使用されるペンのペンタイプが判定される。

【0568】

ステップS1207_03においてペンタイプが通常（デフォルト）であると判定された後のステップS1207_05では、図54の指数減衰（又はダンピング）関数（図54の「減衰」を参照）のような通常（デフォルト）の関数を用いてパラメータvnから半径が導出される。

20

【0569】

ステップS1207_03においてペンタイプが、特に柔らかいペン先を有するペンのように、特殊なものであると判定された後のステップS1207_05では、半径は、図54の「S字」及び「周期」関数のような特別な関数を用いてパラメータvnから導出される。通常及び特別な関数のいずれも、設定GUIを介し、図50Bのコンテキスト情報「Cntx3」の一部として明示的に定義又は修正され得る。

【0570】

図54は、本発明の実施の形態による、ステップS1207_05及びS1207_07で使用される、パラメータvn（速度）から半径を導出する3つの関数を説明するグラフである。

30

【0571】

水平の軸は、パラメータvn（速度）を示し、垂直な軸は半径を示す。

【0572】

「減衰」と称される実線は、図52のステップS1207_05で使用される通常の減衰（又はダンピング）関数を示している。この関数は、vnが増加すると半径が指数的に減衰するという関係を定義する。通常関数は、通常（デフォルト）タイプのペンのために使用される。そのようなvn－半径変換を引き起こすこの関数の使用は、以下の観測に基づいている。

40

【0573】

〔観測A〕筆圧に起因して増加する線幅は、インクが紙の中に染み込む面積に対応する。

【0574】

〔観測B〕ペンが速く動くほど、各ポイントにおいてインクを染み込ませるためにペンが有する時間が短くなる。

【0575】

上記観測A及びBに基づき、ペンがゆっくり動くほど線幅が増加し、一方で、ペンが速く動くほど線幅が減少する、ということが理論化される。この理論は、ペンが速く動くほど、接触する各ポイントで線（ストローク）から紙の中にインクを染み込ませるためにペ

50

ンが有する時間が短くなる、ということに基づいている。線幅は、一連のポイントがそれぞれ有する半径であると考えられる。したがって、本発明の実施の形態によれば、通常のペンに対しては、パラメータ v_n を半径に変換するために減衰（又はダンピング）関数を使用され、それにより、速度が増加すると半径が指数的に減衰することになる。

【0576】

筆圧データを取得する能力を有しない装置に関してさえも、本発明のインクデータ処理部 100T は、タイムスタンプ情報を用いて速度情報を算出又は取得することに注意が必要である。例えば、インクデータ処理部 100T は、タイムスタンプ情報を算出又は取得するために、そのような装置からペンストローク情報を受信するローカルなタイミングを使用することができ、それに基づいて、速度 v_n が各ポイントに対して決定され得る。したがってインクデータ処理部 100T は、圧力データを取得する能力を有する装置、及び、圧力データを取得する能力を有しない装置を含む様々なタイプの装置に関して、速度 v_n に基づいて各ポイントの半径を高い信頼性で決定し、出力することができ得る。

【0577】

図 54 において、「S 字」と称される破線は、減衰が指数的ではなく段階的に発生する特別な関数を示し、「周期」と称される破線は、周期的である他の特別な関数を示している。これらの特別な関数はともに、図 71 を参照して後により完全に記述するように、図 52 のステップ S1207_07 において、特殊なタイプのペンのために速度から半径を導出するために用いられ得る。

【0578】

速度をポイントの半径に変換するための上述したこれらの関数のいずれのアプリケーションもリアルタイムであり得る、ということに注意が必要である。或いは、パラメータ v_n の半径への変換は予め実行されてもよく、結果として得られるデータは、インクデータ処理部 100T によってアクセス可能であるルックアップテーブルに格納されてもよい。

【0579】

図 55 は、本発明の実施の形態による、図 50A のステップ S1209 で各ポイントの透明度（又は不透明度）を示すアルファ値を導出する処理を説明するフロー図である。

【0580】

ステップ S1209_03 では、ペンイベントデータを入力するために使用されるペンのペンタイプがコンテキスト情報（入力 2）から判定される。

【0581】

ステップ S1209_03 においてペンタイプが通常（デフォルト）であると判定された後のステップ S1209_05 では、図 56 に示す「冪」関数のような通常（冪）関数を用いて、パラメータ v_n からアルファが導出される。

【0582】

ステップ S1209_03 において、コンテキスト情報（入力 2）により提供されたペンタイプが、特に柔らかいペン先を有するペンのように、特殊なものであると判定された後のステップ S1209_07 では、図 56 に示す「S 字」関数のような特別な関数を用いて、パラメータ v_n からアルファが導出される。

【0583】

図 56 は、本発明の実施の形態による、図 55 のステップ S1209_05 及び S1209_07 で使用される、パラメータ v_n からアルファ（透明度／不透明度）を導出するための 2 つの関数を説明するグラフである。

【0584】

水平の軸は、パラメータ v_n （速度）を示し、垂直な軸は、透明度を示すアルファを示す。例えば、アルファ 0.0 は完全に透明であることを意味し、アルファ 1.0 は完全に透明性なし、すなわち不透明であることを意味することができる。

【0585】

「冪」と称される実線は、図 55 のステップ S1209_05 で使用される通常の冪関数を示している。この関数は、速度 v_n が増加すると半径が指数的に増加するという関係

10

20

30

40

50

を定義する。通常（冪）関数は、通常（デフォルト）タイプのペンのために使用される。そのような v_n から半径への変換を引き起こすこの関数の使用は、以下の観測に基づいている。

【0586】

〔観測 C〕筆圧に起因して増加するインクの濃度は、インクが紙の中に染み込む面積に対応する。

【0587】

〔観測 D〕ペンが速く動くほど、各ポイントでペンから染み出すインクの量が少なくなる（なぜなら、ペンが各ポイントに接触している時間が短くなるから）。

【0588】

上記観測 C 及び D に基づき、インクの濃度はペンがゆっくり動くほど増加し（不透明度が増加し）、一方で、ペンが速く動くほどインクの濃度が減少する（透明度が増加する）ことが理論化される。この理論は、ペンがゆっくり動くほど、接触している各ポイントにおいてペンから染み出すインクが増して濃い線（ストローク）を形成し、ペンが速く動くほど、接触している各ポイントにおいてペンから染み出すインクの量が減って明るい線（ストローク）を形成する、ということに基づいている。したがって、本発明の実施の形態によれば、通常のペンに対しては、パラメータ v_n をアルファに変換するために冪関数が使用され、それにより、速度が増加するとアルファ（透明度）が指数的に増加することになる。

【0589】

筆圧データを取得する能力を有しない装置に関してさえも、本発明のインクデータ処理部 100 T は、タイムスタンプ情報を用いて、高い信頼性で速度情報を算出又は取得することに注意が必要である。例えば、インクデータ処理部 100 T は、タイムスタンプ情報を算出又は取得するために、そのような装置からペンストローク情報を受信するタイミングを使用することができ、それに基づいて、速度 v_n が決定され得る。したがってインクデータ処理部 100 T は、圧力データを取得する能力を有する装置、及び、圧力データを取得する能力を有しない装置を含む様々なタイプの装置に関して、速度 v_n に基づいてアルファを高い信頼性で決定し、出力することができる。

【0590】

図 56 において、「S 字」と称される破線は、図 55 のステップ S 1209 __ 07 において、特殊なタイプのペンのために速度からアルファを導出するために用いられ得る増加関数の一例である特別な関数を示している。

【0591】

図 56 を参照して上述したこれらの関数のいずれのアプリケーションもリアルタイムであり得る、ということに注意が必要である。或いは、パラメータ v_n のアルファへの変換は予め実行されてもよく、結果として得られるデータは、インクデータ処理部 100 T によってアクセス可能であるルックアップテーブルに格納されてもよい。

【0592】

上述したように、インクデータ処理部 100 T のインクデータ生成部 120 は、筆圧データを含み又は含まない入力ペンイベントデータに基づいて、各ポイントオブジェクトの半径とアルファの値を決定する。図 49 に示すように、ポイントデータ導出部 120 は、それ自身の内部メモリ内に「float」データ型で半径とアルファの情報を出力する。その後、インクデータ整形部 140 は、ポイントオブジェクトの属性として半径とアルファの情報（float 型）を受信し、ストロークファイルフォーマット（SFF、図 48 L 参照）又はストロークメッセージフォーマット（SMF）の中にそれらを出力する。

【0593】

図 57 は、入力された X 及び Y 座標データとともに、入力された半径及びアルファの情報を、ストロークファイルフォーマット（SFF）内又はストロークメッセージフォーマット（SMF）内に整形するインクデータ整形処理を説明するフロー図である。一般的に、整形処理 S 140 は、インクデータ 200 を生成する処理の一部として、インクデータ

10

20

30

40

50

処理部100Tのインクデータ整形部140によって実行される。

【0594】

<データオブジェクトInkDataFileのシリアルライズ>

SFF内のInkDataFileをシリアルライズするステップS1401では、インクデータと、SFFファイル内のInkDataFileを永続的なものとする目的のために必要となる情報とがシリアルライズされる。そのような必要となる情報の例として、decimalPrecision値がシリアルライズされ、符号化される。図示した例では、値100が「符号なし整数」データ型のdecimalPrecision値として使用され、値100が、ASN.1、BERといった符号化方法、又は、第1の実施の形態の図10のスキーマファイル内に示される「sint32型」及び「float」のような符号化方法によって符号化される。これに対し、リアルタイム伝送又はメッセージングに適したSMF内のInkDataFileをシリアルライズするときには、SFF内のインクデータを永続的なものとする目的のために必要となる情報は必要とされず、したがって、SMF内のインクデータを整形するときにはステップS1401は省略され得る。ステップS1402では、InkDataFileのためのstrokesCount値がSFF内に符号化される（図48L参照）。これに対し、SMFに整形するときには、strokesCount値が含まれないので、ステップS1402は省略されることができ、プロセスは代わりに、処理されるすべてのストロークの中の最後のものを示すデータを符号化することとしてもよい。

10

【0595】

<データオブジェクト「stroke」のシリアルライズ>

図57において点線で書かれたより大きな四角に含まれるステップS1403から始まる以下のステップは、InkDataFile内に含まれるN(strokesCount)個のストロークのそれぞれに対して実行される。結果として、N個のストロークが規定の符号化方法を用いて整形され、出力される。

20

【0596】

ステップS1405では、variableAlpha値が、処理されるストロークオブジェクト210内において符号化される。上述したように、このvariableAlpha値(TRUE/FALSE)は、ストロークのアルファ値がストロークの長さに沿って可変かを示している。

30

【0597】

ステップS1406では、ストロークオブジェクト210内に含まれるポイントオブジェクトの数を示すpointsCount値が符号化される。例えばリアルタイム型のアプリケーションのケース（すなわち、SMFに整形するとき）において、もしpointsCount値が利用可能でなければ、ステップS1406は省略されることができ、プロセスは代わりに、処理されるストロークの終了を示すデータを符号化することとしてもよい。

【0598】

<データオブジェクト「point」のシリアルライズ>

図57において点線で書かれたより小さな四角に含まれるステップS1407から始まる以下のステップは、整形されるストローク内に含まれるpointsCount個のポイントのそれぞれに対して実行される。結果として、pointsCount個のポイントオブジェクトが整形され、出力される。

40

【0599】

ステップS1409では、処理されるポイントを含むストロークのアルファ値が可変か否かが判定される。すなわち、variableAlpha値がTRUEかFALSEかが判定される。

【0600】

ステップS1409において、そのストロークのためのアルファがその長さに沿って可変であり（「TRUE」）、したがってアルファはポイントから他のポイントまで変化し

50

得ると判定された後のステップ S 1 4 1 1 では、図 5 8 の上部を参照して後により完全に記述するように、ポイントに対して X Y 座標の値が半径及びアルファの値とともに符号化される。

【0601】

ステップ S 1 4 0 9 において、そのストロークのためのアルファが可変ではないと判定された後のステップ S 1 4 1 3 では、図 5 8 の下部を参照して後により完全に記述するように、ポイントに対して X Y 座標の値及び半径の値のみが符号化され、アルファは符号化されない。

【0602】

この時点で、本発明の実施の形態による規定のデータ構造内に配置されたインクデータ 2 0 0 は、適当なファイルフォーマット（例えば S F F）又はメッセージフォーマット（例えば S M F）にて、様々なタイプのメディアに対して出力され得る。

【0603】

図 5 8 は、本発明の実施の形態による、上述したステップ S 1 4 1 1 及び S 1 4 1 3 の実施例を説明している。

【0604】

図 5 8 の上部の 0 1 行～0 7 行は、アルファ値がストロークの長さに沿って可変である場合に、ポイントについて X Y 座標の値が半径及びアルファの値の両方とともに符号化されるときである図 5 7 のステップ S 1 4 1 1 に対応する擬似コードである。

【0605】

図 5 8 の下部の 0 8 行～1 3 行は、アルファ値がストロークに対して可変でない場合に、ポイントについて X Y 座標の値及び半径の値が符号化され、アルファの値が符号化されないときである図 5 7 のステップ S 1 4 1 3 に対応する擬似コードである。

【0606】

図 5 8 において、「A」及び「B」により示されるセクションは、本発明の実施の形態の実行に際し、上述した `decimalPrecision` 値がどのように利用されるかを示している。

【0607】

好ましくは、ポイントの X Y 座標の値及び半径の値は、プロセッサ内の値のために可能な最高の精度を維持するため、出力タイミングの直前まで浮動データ型又はダブルデータ型で維持される。これに対し、色々なデータ解釈方法によって広く（共通に）インクデータ 2 0 0 を理解可能とする目的、及び、結果として得られるインクデータ 2 0 0 を効率的に圧縮する目的のためには、各値を表すために最小数のビットを用いることが望まれ得る。

【0608】

したがって、図 5 8 の上部に示すように、ステップ S 1 4 1 1 では、入力された X、Y 及び半径の値は、図 5 8 の「A」に示すように、まず、それらの単位を標準化するための解像度（倍率）を示す `decimalPrecision` 値によって増加せられ、その後、「B」に示すように、`int`（整数）データ型に変換（キャスト）される。

【0609】

図 5 8 の下部に示すステップ S 1 4 1 3 でもまた、ステップ S 1 4 1 1 と同様、入力された X、Y 及び半径の値が `decimalPrecision` 値によって増加せられ、その後、`int`（整数）データ型にキャストされる。

【0610】

この点について、図 5 9 は、本発明の実施の形態による、図 5 7 のステップ S 1 4 1 1 及びステップ S 1 4 1 3 で使われる整数データ型への浮動データ型の変換を説明している。

【0611】

図 5 9 において、浮動データ型のものとして記憶される入力データ 1 6 1 は、IEEE 7 5 4 標準による例である。入力データ 1 6 1 において、「s」は符号化ビットであり、

10

20

30

40

50

「`exponent`」は浮動小数点の数の冪指数であり、「`fraction`」は浮動小数点の数の仮数である。

【0612】

入力データ161には、図59の増加入力データ163を生成するため、図58の「A」によって示されるとともに上述したように、`decimalPrecision`値が乗算される。増加入力データ163もまた、冪指数及び仮数を含む浮動小数点の数である。

【0613】

増加入力データ163は、図58の「B」によって示されるとともに上述したように、「`float`」から「`int`」（又は「`short`」又は「`ushort`」）へ変換（キャスト）され、それによって絶対値165が生成される。絶対値165はもはや浮動小数点の数ではない。図示した実施の形態では、XY座標の値と半径の値はすべて`int`（整数）にキャスト（変換）されるが、任意の浮動小数点でないデータ型にキャストされることとしてもよい。

10

【0614】

図60は、本発明の実施の形態による、上述した「`float`」から「`int`」へのデータフォーマットの変換（キャスト）の結果としての圧縮の効率化を説明している。

【0615】

図60では、説明と理解の容易のために、十進数が使用される。図示した例では、円周率 π に色々な10の指数（0、+1、+2の冪乗）が乗算される。

20

【0616】

図60の左手側は、図58のステップS1411及びS1413の処理に先立って取得される値を示している。

【0617】

図60の右手側は、図58のステップS1411及びS1413の処理の後に取得される値を示している。

【0618】

左手側の四角の中に含まれる先頭の3つの値及び最後の値は、処理の前における色々な浮動小数点の数（3.0、3.1、3.14）であるが、これらはすべて、右手側に示す処理の後には、3に変換される。各数の正確さは変換によっていくらか妥協されており、値の効率的な圧縮のようにプロセッサにおける値の効率的な処理を促進するために、同じ値（例えば、この例では3）の使用頻度が増している。例えば、データタイプ整数で表現されるデータは、ASN.1、BER、又はDERといった符号化方法を用いて、出力対象のファイル又はメッセージプロトコルに符号化され得る。

30

【0619】

図61は、本発明の実施の形態による、規定されたインクデータオブジェクトの属性を圧縮するために図49の「インクデータ整形部」140で実行され得る処理を説明するフロー図である。

【0620】

ステップS181では、予備的なステップとして、図57のステップS1411及びS1413で実行されたように、圧縮対象となる生成されたインクデータ200が整数精度の値として量子化される。

40

【0621】

ステップS183では、インクデータ整形部140がデータ圧縮のタイプを判定する。例えば、この判定は、出力フォーマットのタイプに依存し得る。また、この判定は、圧縮がリアルタイムデータを要求するアプリケーションのためのものであるのか、記憶装置タイプのメディア内でデータを利用するアプリケーションのためのものであるのか、に基づくものであり得る。もし、圧縮が実行されない（「NONE」）のであれば、プロセスは、整数データ型のインクデータ値を「そのまま」出力する。例えば図60に記述される先頭の3つの値を用いると、32ビットで表される値「+3」が3回出力され得る。

50

【0622】

ステップS183において第1のタイプの圧縮方法が選択された後のステップS185では、プロセスは、X座標値、Y座標値、半径値などのようなデータシーケンスをそれらの属性に従うストリームへとカテゴリイズし、生成されたインクデータ200に対して第1のタイプの圧縮方法を適用する。例えば、第1のタイプの圧縮方法は、ランレングス符号化方法である。例えば図60に記述される先頭の3つの値を用いると、値「+3」が3回繰り返されることを示す符号が使用される。提案される符号化技術の効率性は、ランレングス符号化を適用する前に値に対してデルタ符号化（データ差分）を実行することのような、いくつかの符号化方法を用いて改良され得る。これは、値間の変化がどちらかと言えば絶えず続くときには、繰り返される値の数を増加させるであろう。

10

【0623】

ステップS183において第2のタイプの圧縮方法が選択された後のステップS185では、プロセスは、生成されたインクデータ200に対して第2のタイプの圧縮方法を適用する。例えば、第2のタイプの圧縮方法は、指数ゴロム符号を用いるエントロピー符号化方法である。これは、+3のようなより小さな絶対値を有する値に対し、より大きな絶対値を有する値と比べて短いビット長を用いる可変長の符号化方法である。

【0624】

＜インクデータ処理部100T及びインクデータ生成方法に対する変形例＞

上述したように、本発明の実施の形態によるインクデータ処理部100Tは、各ストロークを構成するポイントの属性として半径及び／又はアルファ（透明度／不透明度）の情報を導出するために、いくつかは筆圧データを含み、他は筆圧データを含まない、様々なタイプの装置から受信されるデータを処理し、それによって非装置依存型のインクデータ200を生成する能力を有する。インクデータ処理部100Tは、ファイルフォーマット（例えばSFF）又はメッセージフォーマット（例えばSMF）のような様々なフォーマットで、生成したインクデータ200を出力する。

20

【0625】

図50Aの例では、入力データが筆圧データを含むケース（ステップS1202の結果がTRUEであるとき）において、半径と透明度の情報が、タイムスタンプ情報を使用することなく、筆圧データから導出される。ただし、本発明はそのような実施例に限定されない。

30

【0626】

タイミング情報及び筆圧データを入力として受信し、半径を出力する関数を使用され得る。この場合、筆圧データだけでなくペンの速度にも基づいて、ストローク幅及び／又はストロークの透明度を変更することが可能になる。

【0627】

上述した様々な実施の形態において速度が導出される一方で、例えばペンが加速度センサを含むときや、加速に対応する値のみが取得可能であるときには、加速度が使用され得る。例えば、速度を導出するために加速度を積分することにより、上記に似た処理が利用され得る。

【0628】

図62は、インクデータの属性として半径情報を出力するため、図49のインクデータ生成部120において、図52を参照して上述した方法に代えて実行される処理の他の例を示すフロー図である。

40

【0629】

ステップS192では、図50AのステップS1202と同様に、入力されたデータが筆圧データを含むか否かが判定される。

【0630】

ステップS192において入力されたデータが筆圧データを含まないと判定された後（「FALSE」）のステップS196では、上記図54を参照して説明したように、タイムスタンプ情報から導出される速度と半径との間の関係を用いて半径が算出される。すな

50

わち、速度が増加するときに半径が減衰する、という関係が使用される。

【0631】

ステップS192において入力されたデータが筆圧データを含むと判定された後（「TRUE」）のステップS194では、入力された圧力データとタイムスタンプ情報の両方が半径を導出するために使用される。ステップS192は、筆圧が固定されているケースでは（i）速度が増加するときに半径が減少し、速度が固定されているケースでは（ii）筆圧が増加するときに半径が増加する、という2つの変数を含む偏微分関数を使用する。したがって、ポイントの速度と筆圧の両方に基づいて、インクデータ200の各ポイントの半径を符号化することが可能になる。

【0632】

ステップS198では、ステップS196又はステップS194で導出された半径が出力される。

【0633】

図63は、インクデータの属性として可変のアルファ情報を出力するため、図49のインクデータ生成部120において、図55を参照して上述した方法に代えて実行される処理の他の例を示すフロー図である。

【0634】

ステップS2002では、図50AのステップS1202と同様に、入力されたデータが筆圧データを含むか否かが判定される。

【0635】

ステップS2002において入力されたデータが筆圧データを含まないと判定された後（「FALSE」）のステップS2006では、上記図56を参照して説明したように、タイムスタンプ情報から導出される速度とアルファとの間の関係を用いてアルファ（透明度／不透明度）が算出される。すなわち、速度が増加するときにアルファもまた増加する（より透明になる）、という関係が使用される。

【0636】

ステップS2002において入力されたデータが筆圧データを含むと判定された後（「TRUE」）のステップS2004では、入力された圧力データとタイムスタンプ情報の両方がアルファ値を導出するために使用される。ステップS2004は、筆圧が固定されているケースでは（i）速度が増加するときにアルファが増加し（より透明になり）、速度が固定されているケースでは（ii）筆圧が増加するときにアルファが減少する（より不透明になる）、という2つの変数を含む偏微分関数を使用する。したがって、ポイントの速度と筆圧の両方に基づいて、インクデータ200の各ポイントのアルファ（透明度）を符号化することが可能になる。

【0637】

ステップS2008では、ステップS2006又はステップS2004で導出されたアルファが出力される。

【0638】

図62及び図63の処理は、半径とアルファの両方の値が入力された圧力データ及びタイムスタンプ情報から導出され得るよう、一緒に用いられ得る。或いは、半径の値のみ又はアルファの値のみが、入力された圧力データ及びタイムスタンプ情報から導出されることとしてもよい。

【0639】

速度（ v_n ）を半径又はアルファに変換する関数のサンプルについては、図54及び図56を参照して上に述べた。一般に、速度（ v_n ）を半径及び／又はアルファに変換するために何の関数が使用されるべきか、ということは、ペンのタイプ（又はペン先のタイプ）、及び、センサ表面がシミュレートすることを期待される「紙」のタイプ（例えば、和紙、カーボン紙、普通紙、コピー用紙、写真用紙、インク吸収紙など）に依存する。したがって、速度を半径及び／又はアルファに変換するために選択される関数のいくつかは、ペンのタイプ及び／又は紙のタイプによって調整され得る。別の言葉で言えば、速度から

10

20

30

40

50

導出される半径及び／又はアルファは、ペンのタイプ及び／又は紙のタイプによって変化し得る。

【0640】

本発明のさらなる側面に従えば、上述したように本発明の様々な実施の形態で利用される筆圧データは、ペン角度（又はペンの傾き）データやペンの回転（ペンの転がり）データなどのような、様々なタイプの入力装置から受信され得る他のタイプの属性値によって置き換えられ得る。ペンの角度／傾きデータのようなこれらの属性値は、本発明の様々な実施の形態による半径及び／又は透明度（アルファ）の情報を導出するために、上述した例において使用した筆圧データに代えて使用され得る。例えば、いくつかのペン型入力装置は、ペン軸によりセンサ表面又はセンサ表面の法線に対して作られる角度を示すペン角度（ペンの傾き）データを生成する能力を有する。センサ表面に対して垂直に保持されるペンは、傾いているためにセンサ表面に対してより平行な方向に延在するペンより、センサ表面に対して大きな圧力を加える傾向にある。したがって、筆圧データがパラメータ v_n （速度）と相互に関連付けられ得るのと同様、ペンの角度／傾きデータも v_n と相互に関連付けられ得る。

【0641】

例えば、ペンがセンサ表面に対してよりまっすぐに（すなわち、センサ表面の法線に近づくように）保持されるほど（すなわち圧力が大きくなるほど）、速度（ v_n ）が遅くなるという関係を規定する関数を使用され得る。一旦 v_n が角度／傾きデータから導出されると、 v_n を半径及び／又はアルファ情報に変換するために使用される上述した関数と同様の関数を使用され得る。すなわち、ペンの角度／傾きデータは、後に半径及び／又はアルファ情報に変換される v_n と相互に関連付けられることができる。これに対し、センサ表面に垂直に保持されるペンが、おそらくは傾いたペン先とセンサ表面との間の増加した接触面積に起因してより広いストロークを生成（描画）する傾向にある傾いたペンよりも、狭いストロークを生成する傾向にあることもまた、観測され得る。この場合、ペンがセンサ表面に対してまっすぐに保持されるほど、速度（ v_n ）が速くなって狭いストロークが生成されるようになる、という関係を規定する適切な関数を使用され得る。一旦 v_n が角度／傾きデータから導出されると、 v_n はその後、上述した関数を用いて半径及び／又はアルファ情報に変換され得る。角度／傾きデータを v_n に変換するために何の関数を使用されるか、ということは、ペンのタイプ（又はペン先のタイプ）、及び、センサ表面がシミュレートすることを期待される「紙」のタイプ（例えば、和紙、カーボン紙、普通紙、コピー用紙、写真用紙、インク吸収紙など）に依存する。さらに、センサ表面に対して垂直に保持されたペンは、より狭くより細かい（より透明な）ストロークを生成する傾向にある傾いたペンよりも、より広くより濃いストロークを生成する傾向にあることが観測され得る。そこで、いくつかの実施の形態では、最初に角度／傾きデータを v_n と相互に関連付け、 v_n をポイントごとの半径及び／又はアルファ情報に変換するのではなく、ペンの角度／傾きデータを直接的に半径及び／又はアルファ情報に変換する適切な関数を使用され得る。同様に、いくつかの実施の形態では、もし利用可能であるならば筆圧データを、最初に圧力データを v_n に変換し、その後に v_n をポイントごとの半径及び／又はアルファ情報に変換することに代え、直接的に半径及び／又はアルファ情報に変換する適切な関数を使用され得る。

【0642】

[4] インクデータ処理部／復号器（図64～図66）

図64は、本発明の実施の形態による、インクデータ処理部100Rと様々なアプリケーション300-1, 300-2, 300-nとの間の関係を説明する図である。インクデータ処理部100Rは、第1の実施の形態の図22における100Rに対応する。

【0643】

図45に示されるように、インクデータ処理部100Rは本質的に、様々なアプリケーション300からの要求に応じて、メモリ（「RAM」）や様々なタイプのメディア（例えばハードディスク）内にあるファイル又はメッセージフォーマット内に格納されている

インクデータ200を、あるメモリ位置に、アプリケーションによって使用可能なデータフォーマットで取り出すソフトウェアライブラリである。例えば、アプリケーション300-1が(図5のグラフィック処理部300を利用する)描画処理アプリケーションであるとき、インクデータ処理部100Rは、必要なデータオブジェクト「ストローク」、「ポイント」などを(各ストローク及び/又はポイントが半径及び/又はアルファ情報に関連付けられる)グラフィック処理部300に対して出力するが、必要でないデータオブジェクト「著者」などは出力しない。他の例として、例えばアプリケーション300-2が電子会議の参加者を判定する目的で著者情報のみを要求するとき、インクデータ処理部100Rは、メタデータオブジェクト250内のデータオブジェクト「著者」を出力する。本発明の実施の形態では、インクデータ処理部100Rは、図65に記述されるように、プロセッサによって実行されるインクデータ再生処理S200として実現される。以下、アプリケーション300が描画アプリケーション300-1である描画処理に関して、インクデータ再生処理S200が記述されるであろう。

10

【0644】

図65は、本発明の実施の形態による、生成されたインクデータを再生し(取り出し)することにより、半径及びアルファの情報をX及びY座標とともに取得するインクデータ再生処理S200と、取得した情報及びデータを描画アプリケーション300-1からの要求に応じて出力することとを説明するフロー図である。描画アプリケーション300-1はその後、ストローク上に描画/レンダリングするときに、よりリアルな外観と表現の微妙な違いとをストロークに与えるために、半径及びアルファの情報を使用することができ(図68~図72参照)。本質的に、インクデータ再生処理S200は、上記図57を参照して記述したインクデータS140を生成する(整形する)処理の逆処理である。

20

【0645】

<データオブジェクトInkDataFileの再生又は取り出し>

ステップS2001では、decimalPrecision値のような、処理対象のストロークを含むインクデータ200のためのコンテキスト情報(入力2)(又は構成情報)が取り出される。この再生処理は、図57のステップS1401における符号化処理に逆対応する。

【0646】

ステップS2002では、データオブジェクトInkDataFile内に含まれるstrokesCount値が、図48Lに示すように、取り出される。例えばもしstrokesCount値が利用可能でなければ、ステップS2002は省略されることができ、プロセスは代わりに、InkDataFile内に含まれるすべてのストロークのうちの最後のものを示すデータを再生することにより、いつ処理を終わらせるかを判定することとしてもよい。

30

【0647】

<データオブジェクト「stroke」の再生>

図65において点線で書かれたより大きな四角に含まれるステップS2003から始まる以下のステップは、InkDataFile内に含まれるN(strokesCount)個のストロークのそれぞれに対して実行される。結果として、N個のストロークが再生され、出力される。

40

【0648】

ステップS2005では、データオブジェクト「stroke」内のvariableAlpha値(図48L参照)が取り出される。上述したように、このvariableAlpha値(TRUE/FALSE)は、ストロークの長さに沿って可変なアルファ値を処理されるストロークが含むかを示している。

【0649】

ステップS2006では、ストロークオブジェクト210内に含まれるポイントオブジェクトの数を示す「pointsCount値」が取得される。例えばリアルタイム型のアプリケーションにおいて、もしpointsCount値が利用可能でなければ、ステ

50

ップ S 2 0 0 6 は省略されることができ、プロセスは代わりに、処理されるストロークの終了を示すデータを再生することにより処理の終了を判定することとしてもよい。

【0650】

＜データオブジェクト「point」の再生＞

図 6 5 において点線で書かれたより小さな四角に含まれるステップ S 2 0 0 7 から始まる以下のステップは、再生されるストロークオブジェクト 2 1 0 内に含まれる points Count 個のポイントのそれぞれに対して実行される。結果として、points Count 個のポイントオブジェクトが再生され、出力される。

【0651】

ステップ S 2 0 0 9 では、処理されるストロークのアルファが可変か否かが判定される。すなわち、variable Alpha 値が TRUE か否かが判定される。

10

【0652】

ステップ S 2 0 0 9 において variable Alpha 値が TRUE であると判定された後のステップ S 2 0 1 1 では、そのポイントについて、XY 座標値とともに半径及びアルファ値が復号され、出力される。このステップは、再生されるインクデータ 2 0 0 を要求している特定のアプリケーションによって要求される（使用される）データフォーマットに、図 5 7 のステップ S 1 4 1 1 で符号化されたデータを逆転させる（変換する）ものである。

【0653】

ステップ S 2 0 0 9 において variable Alpha 値が FALSE であると判定された後のステップ S 2 0 1 3 では、図 6 6 の下部で後により完全に記述するように、アルファ値がストロークの全体に対して固定値として設定される一方で、そのポイントについて XY 座標値及び半径が復号及び再生され、出力される。このステップは、再生されるインクデータ 2 0 0 を要求している特定のアプリケーションによって要求される（使用される）データフォーマットに、図 5 7 のステップ S 1 4 1 3 で符号化されたデータを逆転させる（変換する）ものである。

20

【0654】

したがって、インクデータ再生処理 S 2 0 0 は、データオブジェクト「point」から、XY 座標値及び半径値、及び、ある場合にはアルファ値を取り出す。

【0655】

図 6 6 は、本発明の実施の形態による、上述した図 6 5 のステップ S 2 0 1 1 及び S 2 0 1 3 の実施例を説明している。

30

【0656】

図 6 6 の上部の 0 1 行～0 7 行は、ポイントを含むストロークの長さに沿ってアルファ値が可変である（すなわち、ストロークを構成する色々なポイントが色々なアルファ値を有することができる）場合に、そのポイントについて XY 座標の値が半径及びアルファの値の両方とともに取り出されるときである図 6 5 のステップ S 2 0 1 1 に対応する擬似コードである。

【0657】

図 6 6 の下部の 0 8 行～1 4 行は、ポイントを含むストロークの長さに沿ってアルファ値が可変でない（すなわち、ストロークを構成するすべてのポイントが同じアルファ値を有する）場合に、ポイントについて XY 座標の値及び半径の値が取り出される一方、アルファの値として固定値（例えば、図示した例では「1. 0」）が設定されるときである図 6 5 のステップ S 2 0 1 3 に対応する擬似コードである。

40

【0658】

図 6 6 において、「A」及び「B」により示されるセクションは、本発明の実施の形態の実行に際し、上述した decimal Precision 値がどのように利用されるかを示している。具体的には、図 6 6 の「INV__B」に示すように、図 5 8 に示した関数の逆関数を用いて、ステップ S 2 0 1 1 において、入力された X、Y 及び半径及びアルファのデータが int（整数）から float（浮動小数点）データ型にまず逆変換（キャ

50

スト)される。その後、図66の上部の「INV_A」に示すように、解像度を示す decimal Precision 値によって X, Y 及び半径のデータが除される。

【0659】

図66の下部に示すステップS2013では、ステップS2011と同様に、入力された X, Y 及び半径の値が float (浮動小数点) データ型にキャストされ、その後、 decimal Precision 値によって除される。これに対し、アルファ値は、図示した例における「1.0」のような固定値として設定される。

【0660】

したがって、アプリケーション300が、例えば入力データが「float」データ型であることを要求する描画アプリケーションであるとき、整数型のデータを含む生成されたインクデータ200は、要求された浮動小数点データ型、又は、アプリケーション300によって要求された任意の他の非整数の元のデータ型に戻る方向で再生(復号)される。

10

【0661】

[5] インクデータ描画処理(図67)

図67は、本発明の実施の形態による、スクリーン上にストロークを描画(レンダリング)するためにインクデータ200を利用する描画アプリケーション300-1(及び図5のグラフィック処理部300)によって実行される描画処理S300-1を説明するフロー図である。

【0662】

20

ステップS200において、プロセスは、ストロークと各ストロークに含まれるポイントに関する情報を取り出すため、上述したようにして、インクデータ処理部100Rに InkDataFile を取得させて再生させ、それにより、取り出された情報が描画処理S300-1の入力データとして使用可能となる。

【0663】

次に、処理されるストロークオブジェクト210に関連付けられた描画(ラスター化)スタイルオブジェクトが判定される。図48Bに示したように、インクデータ200は、(ストロークモデルサブドメイン内の)各ストロークオブジェクト210が、スクリーン上に描画(レンダリング、ラスター化など)されたときのストロークオブジェクトの外観を定義する1以上の(ラスター化サブドメイン内の)描画スタイルオブジェクトと関連付けられるように構造化されている。多くのタイプの描画スタイルオブジェクトが存在する一方で、図67の図示した実施の形態では、散乱スタイルオブジェクトと、シェイプフィルスタイルオブジェクトとの2つのオプションが利用可能である(図48参照)。

30

【0664】

散乱スタイルオブジェクトが選択されるとき、サブ処理S300-1Sにおいて、ストロークごとに頂点(ポイント)配列が導出される。この導出は、頂点配列が低密度に位置する一組の離散ポイントからなるようになされる。頂点配列を導出する処理は、「間隔」及び「範囲」の値のようなインクデータ処理部100によって生成された属性値を使用する。頂点配列を導出する処理はまた、アプリケーション又はオペレーティングシステムから受信されるコンテキスト情報(入力2)を使用することができる。例えば、スクリーンがシミュレートすることを期待される紙タイプ(例えば、和紙、カーボン紙、普通紙、コピー用紙、写真用紙、インク吸収紙など)についてのコンテキスト情報が、頂点配列内において低密度で位置する離散ポイントの個数を増加させ又は減少させるために使用され得る。生成された頂点配列は、一連の粒子を表す。図示した実施の形態では、描画処理S300-1Sを制御するGPUが、各ポイントの「半径」値に基づいて各粒子に規定サイズを与えるために、生成された頂点配列に対して第1の頂点シェイダーを適用する。このGPUはまた、各ポイントの「アルファ」値に基づいて各粒子に透明度(又は不透明度)の規定レベルを与えるために、粒子の配列に対して第1の断片シェイダーを適用する。描画処理S300-1Sは、このようにして、「散乱」粒子のスタイルで所与のストロークを描画する(図48F参照)。

40

50

【0665】

シェイプフィルスタイルオブジェクトが選択されるとき、サブ処理S300-1Fにおいて、まず各ストロークに対してスプラインセグメントが導出される。この導出は、各スプラインセグメントが、連続するストローク曲線の部分ごとに定義される十分滑らかな多項式関数となるようになされる。すなわち、一組のスプラインセグメントは、頂点をつなぎ、連帯してストロークを表現する複数の曲線セグメントを定義する。GPUは、各頂点（ポイント）の「半径」値に基づき、ストロークに沿って存在する各頂点を中心とする各円に規定のサイズを与えるために、一組のスプラインセグメントに対し第2の頂点シェイダーを適用する。GPUはまた、各頂点（ポイント）の「アルファ」値に基づいて各円に透明度（又は不透明度）の規定レベルを与えるために、一組のスプラインセグメントに対し第2の断片シェイダーを適用する。描画処理S300-1Fは、このようにして、「シェイプフィル」のスタイルで所与のストロークを描画する（図48F参照）。

10

【0666】

〔6〕効果。インクデータ描画例（図68～図72）

図68～図72は、様々な描画レンダリング例を説明している。これらは、本発明の実施の形態による、インクデータ生成方法、インクデータ再生方法、及びインクデータ描画（レンダリング）方法を説明するために使用される。図68～図72において、「s」はストロークの開始位置を示し、「e」はストロークの終了位置を示す。すべてのケースにおいて、ペンの動きの速度は、「s」から「e」にかけて増加している（加速している）と仮定する。

20

【0667】

図68は、本発明の実施の形態による、図54に示した減衰（ダンピング）関数に基づいて生成されたインクデータの出力の結果としての描画レンダリング例を説明している。減衰関数を用いることで、速度が増加すると、半径が減少することになる。したがって、図68に示すすべての描画例において、ストロークの幅は「s」から「e」に向かって減少している。これらの例では、アルファ（透明度）は固定値として設定されている。

【0668】

このレンダリング例は、筆圧に応じて増加する線幅はインクが紙に染み出す面積に対応し〔観測A〕、ペンが速く動くほど、各ポイントにおいてペンがインクを染み出させるために有する時間が短くなる〔観測B〕、という上述した観測結果に従っている。本発明の実施の形態によるインクデータ処理部は、与えられたペンイベントデータ入力に圧力情報を含んでいないときでさえ、各ポイントについて速度情報を取得し、その速度情報に基づいて各ポイントの半径情報を算出する能力を有する。したがって、生成されたインクデータは、少なくともいくつかのポイントのそれぞれについて半径情報を含む。半径情報は、インクデータ200がスクリーン上にレンダリング（描画）されるとき、描画されるストロークに対し、紙上にインクで手書きされた本物のストロークの外観を厳密にシミュレートするリアルな外観及び表現の微妙な違いを与えるために使用され得る。

30

【0669】

図69は、本発明の実施の形態による、図56に示した冪関数に基づいて生成されたインクデータ200の出力の結果としての描画レンダリング例を説明する図である。冪関数を用いることで、速度が増加すると、アルファ（透明度）が増加することになる。したがって、図69に図示したすべての描画例において、ストロークは、「s」から「e」に向かってより明るく、より透明になっている（すなわち、濃度が減少している）。これらの例では、半径は固定値として設定されている。

40

【0670】

このレンダリング例は、筆圧に応じて増加するインクの濃度はインクが紙の中に染み出す面積に対応し〔観測C〕、ペンが速く動くほど（ペンが各ポイントに接触している時間が短くなるため）各ポイントでペンから染み出すインクの量が少なくなる〔観測D〕、という上述した観測結果に従っている。本発明の実施の形態によるインクデータ処理部は、与えられたペンイベントデータ入力に圧力情報を含んでいないときでさえ、各ポイントに

50

ついて速度情報を取得し、その速度情報に基づいて各ポイントのアルファ情報を算出する能力を有する。したがって、生成されたインクデータは、少なくともいくつかのポイントのそれぞれについてアルファ情報を含む。アルファ情報は、インクデータ200がスクリーン上にレンダリング（描画）されるとき、描画されるストロークに対し、紙上にインクで手書きされた本物のストロークの外観を厳密にシミュレートするリアルな外観及び表現の微妙な違いを与えるために使用され得る。

【0671】

図70は、本発明の実施の形態による、図54に示した減衰関数及び図56に示した冪関数の両方に基づいて生成されたインクデータ200の入力の結果としての描画レンダリング例を説明する図である。減衰関数を用いることで、速度が増加すると半径が減少することになり、冪関数を用いることで、速度が増加するとアルファ（透明度）が増加することになる。したがって、図70に図示したすべての描画例において、ストロークの幅が「s」から「e」に向かって減少する一方で、同時にストロークは、「s」から「e」に向かってより明るく、より透明になっている（すなわち、濃度が減少している）。本発明の実施の形態によるインクデータ処理部は、与えられたペンイベントデータ入力が圧力情報を含んでいないときでさえ、各ポイントについて速度情報を取得し、その速度情報に基づいて各ポイントの半径及びアルファ情報を算出する能力を有する。したがって、生成されたインクデータは、少なくともいくつかのポイントのそれぞれについて半径及びアルファ情報を含む。半径及びアルファ情報は、インクデータ200がスクリーン上にレンダリング（描画）されるとき、描画されるストロークに対し、紙上にインクで手書きされた本物のストロークの外観を厳密にシミュレートするリアルな外観及び表現の微妙な違いを与えるために使用され得る。

【0672】

図71は、本発明の実施の形態による、特にやわらかいペン先を有するペンのような特殊なタイプのペンのために図52のステップS1207__07で使用されるものとしての図54の他の関数（S字及び周期関数）の効果を示す描画レンダリング例を説明する図である。

【0673】

左手側の描画例は、減衰が、「減衰」関数における指數的とは対照的に段階的に発生する図54の「S字」関数の結果として得られる描画例である。したがって、結果としての描画（レンダリング）例のそれぞれにおいて、ストロークの半径（幅）は、図68の例において徐々に減少しているのとは対照的に、広い部分から狭い部分まで段階的に減少している。図56の「S字（増加）」関数においては、減衰が段階的に発生する図54の「S字（減少）」関数とは対照的に、増加が段階的に発生する。したがって、図56の「S字（増加）」関数の結果として得られる描画例は、図71の左手側の描画例のそれと類似の外観を有しつつも、「s」と「e」の位置が入れ替わったものとなる。

【0674】

図71の右手側の描画例は、半径の出力が周期的に変化する（増減する）図54の「周期」関数の結果として得られる描画例である。したがって、結果として描画（レンダリング）されるストロークのそれぞれにおいて、「s」から「e」に向かってストロークの半径（幅）が周期的に変化している。

【0675】

図72は、本発明の実施の形態による、図52のステップS1207__04で追加される開始点（s）及び終了点（e）の半径として特別な値を用いることの効果を示す描画レンダリング例を説明している。

【0676】

具体的には、図52のステップS1207__02において、ストロークの開始及び終了にそれぞれ追加される開始点（s）及び終了点（e）のそれぞれに対して、半径として特別な値が設定される。

【0677】

10

20

30

40

50

図 7 2 の左手側は、開始及び終了点の半径がゼロ（「0」）に設定される場合の描画例を説明している。これは、ストロークの開始又は終了においてユーザがいかにかにペンを速く動かしたり遅く動かしたりしようとも、ストロークの開始及び終了点の半径（幅）は、結果としての絵において本質的に無視されるということの意味する。

【0678】

図 7 2 の右手側は、開始及び終了点の半径が、通常算出される半径、すなわち、上述した様々な関数を用いて本発明の様々な実施の形態により算出される半径より大きく（例えば 2 倍に）設定される場合の描画例を説明している。示されるように、このことは、ユーザが紙上でペンを用いてストロークを描くときに（ペンストロークの開始及び終了でペンがしばしば休止するために）各ストロークの開始及び終了点の半径がしばしば紙の上で強調されるのと同様に、各ストロークの開始及び終了点の半径が強調される結果となる。

10

【0679】

上述したように、本発明の様々な実施の形態のインクデータ処理部、インクデータ生成方法、インクデータ再生方法、及びインクデータ描画方法によれば、リアルな外観を有するストロークをレンダリング（描画）するために、非装置依存型のインクデータが生成され、使用され得る。インクデータ 200 は、そのうちのいくつかは圧力データをサポートする一方、他は圧力データをサポートしない様々なタイプの装置及びアプリケーションによって共有され得るよう構造化される。このインクデータ 200 の構造は、各ストロークの各ポイントに対して半径及び／又はアルファの値を定義し、この半径及び／又はアルファの値は、スクリーン上に描画されるストロークに対し、圧力データの代わりに、紙上にインクで手書きされた本物のストロークの外観を厳密にシミュレートするリアルな外観及び表現の微妙な違いを与えるために使用され得る。

20

【0680】

上記の記述においては、大部分において圧力はペンによって印加される筆圧であるとして記述したが、例えば指によって印加された圧力を取得（測定）する能力を有する装置に関しては、圧力は指の圧力を意味することとしてもよい。したがって、本記述において、「筆圧」という用語は「圧力」の同義語と解されるべきであり、「ペン」という用語は、ペン、指、並びに他の任意の、ユーザが入力装置上で位置を指示するために利用できる器具、機器、及び要素を含み得る「指示体」の同義語と解されるべきである。

【0681】

上記の記述においては、アルファは透明度の度合いを示すもの（アルファが大きいほど透明度が大きくなる）として使用したが、その値が大きいほど不透明の度合いが大きくなるという不透明度の度合いを示すパラメータもまた使用可能である。

30

【0682】

インクデータ処理部は、インクデータ処理部から再生されたインクデータを要求する様々なアプリケーションとは別の実体であるとして一般に記述されるが、これらは、例えばライブラリリンクを介する接続に基づき、連带的に又は一体的に構成されてもよい。

【0683】

第 4 の実施の形態

本発明の第 4 の実施の形態は、ユーザの手書き動作に基づくペンイベントデータを受信するとともに、手書き動作をサポートするアプリケーション又はオペレーティングシステムによって提供されるコンテキスト情報（入力 2）を受信するシステム及び方法に向けられたものである。コンテキスト情報（入力 2）は、ペンのタイプや著者 ID などのようなペンイベントデータについての情報を含む。このシステム及び方法は、受信したペンイベントデータ及び受信したコンテキスト情報（入力 2）に基づいて、ストロークオブジェクト 210、メタデータオブジェクト 250、及び描画スタイルオブジェクト 230 を含むインクデータ 200 を生成する。このシステム及び方法はさらに、手書き動作をサポートするアプリケーション又はオペレーティングシステムから操作情報を受信し、受信したペンイベントデータ、受信したコンテキスト情報（入力 2）、及び受信した操作情報に基づいて、インクデータの一部を構成する操作オブジェクトを生成することができる。

40

50

【0684】

第4の実施の形態の背景

色々なオペレーションシステム、色々なアプリケーション、色々なサービス、色々な画像フォーマット、ストロークに関する既存の色々な標準などの間で、デジタル化された手書き入力データ又はインクデータが共有されることを可能にする枠組みが望まれている。要するに、ストロークデータモデルの統一が望まれている。

【0685】

ハイパーテキストマークアップ言語（HTML）は、成功した統一の枠組みの一例である。HTMLは、（1）本質的に決まった数の文字コードの組み合わせである「テキスト」、（2）このテキストが表示されるときにどのように特徴付けられ描写されるべきかを埋め込むメタタグ、を埋め込むための共通の言語として広く採用されてきた。例えば、メタタグはフォント、サイズ、色、列、行、グループ、表などを指示するもので、テキストの外観を特定するための色々なタイプのブラウザにより共通に解釈される。そのような共通言語は、色々なコンピューティング環境の色々な装置上で、実質的に同一の態様で表示され得るドキュメントの生成を可能にする（もっとも、例えば各ブラウザの実装に起因して、いくつかの小さな変種及び相違はあり得る）。

【0686】

同じことが、手書き入力データについても望まれる。すなわち、（1）（ユーザの手書き動作によって入力される）「ストローク」（又は「トレース」又は「パス」）、及び、（2）ストロークの色、テクスチャ、オフセット位置などのような「ストローク」を特徴付け又は描写する「オブジェクト」、を定義する共通言語が望まれる。以下では、そのような共通言語（又は情報モデル）を「ストローク言語（SL）」と称する。ストローク言語（SL）は、色々なコンピューティング環境の色々な装置上で、実質的に同一の態様（同一の外観）で表示され得るデジタルドキュメントの生成を可能にするであろう。もっとも、例えば各レンダリングエンジンの実装に起因して、いくつかの小さな変種及び相違はあり得るが。

【0687】

上記非特許文献1、2、5に記述されるように、InkML、ISF、及びJOTデータ構造のような色々なアプリケーションで共有可能な態様で手書きストロークを表現するように構成されたいくつかのデータ構造が知られている。

【0688】

簡単に言えば、InkML（非特許文献1）は、電子ペン又はスタイラスを用いて入力されたインクを、入力データを記述するマークアップ言語を用いることによって表現する目的のために提供されるものである。例えば、InkMLはストロークのためのデータ構造を定義しており、そこでは、データ構造<trace>が入力装置によって生成された一連のデータを含み、そしてこのデータのフォーマットは、多数の<channel>要素を使用する独自のデータ構造<traceformat>に定められる。

【0689】

ISF（非特許文献2）は、入力機構としてスタイラスを使用するPDA、タブレットPC、及びその他のようなモバイル装置で使用されることを意図するバイナリフォームでインクデータを記憶する目的のために提供されるものである。例えば、ISFはストロークのためのデータ構造を定義しており、そこでは、データ構造TAG__STROKEが入力装置によって生成された一連のデータを含み、そしてこのデータのフォーマットは、TAG__NO__X、TAG__BUTTONSその他の各種タグを使用する独自のデータ構造TAG__STROKE__DESC__BLOCKに定められる。ISFは、圧縮符号化を必要とし、データのタイプごとに最も適した圧縮方法を選び取る方法を用いて、静的な（永続的な）ストリームを生成する能力を有する。彼らは、例えば、入力座標、圧力レベル、及び他のスタイラスによって生成されるデータのためにデルタ符号化とハフマンアルゴリズムの調整されたバージョンとの組み合わせを用い、カスタム描画属性のようなカスタム特性のためにLZアルゴリズムを用いる。

【0690】

JOT（非特許文献5）は、電子ペン又はスタイラスによって入力されたデータを、様々なオペレーティングシステム及びアーキテクチャを用いる色々なマシンの間で交換する目的のために提供されるものである。例えば、JOTはストロークのためのデータ構造を定義しており、そこでは、データ構造 `tag__INK__POINT` が、単一のペニイベントと、その位置、力（圧力）、回転などのような特徴とを記述する。

【0691】

また、手書きストロークを処理することに限定されない色々な標準が、入力から独立した態様でベクトルグラフィックスを記述する目的で存在している。SVG1.1（非特許文献3）は、そのような例の1つである。SVGのバージョン1.1は、ストロークを表現することを目的に、直線及びベジエ曲線に依存するパス要素を含む。

10

【0692】

第4の実施の形態の概要

本発明の実施の形態は、3つの側面のうちの1以上、特に側面3を扱うものとして理解され得る。

【0693】

上述した本発明の1以上の側面を実現するために、新規な言語（又は情報モデル）で定義されるインクデータ200を生成し、変換し、その他の方法で処理するためのシステム及び方法が提供される。

【0694】

本発明の実施の形態は、メタデータオブジェクトによって静的に描写され、及び／又は、描画スタイルオブジェクト及び操作オブジェクトによって動的に制御又は操作されるストロークオブジェクトを含むインクデータ200を出力することに向けられたものである。ストロークオブジェクト、メタデータオブジェクト、描画スタイルオブジェクト、及び操作オブジェクトは、全体として、記録フォーマット（例えば、ストロークファイルフォーマット（SFF））又は伝送フォーマット（例えば、ストロークメッセージフォーマット（SMF））で記憶され得るインクデータ200を構成する。

20

【0695】

本発明の実施の形態によるストロークオブジェクトは、上述した第3の実施の形態で示したように、可変のストローク幅（すなわち、ストロークの長さに沿って変化する幅）、可変のストローク色又は透明度（アルファ）／不透明度（すなわち、ストロークの長さに沿って変化する色又は透明度／不透明度）を有することができる。本発明の実施の形態によるストロークオブジェクトは、上述した第1の実施の形態で示したように、キャットマールームスプライン法のような適切な補間方法を用いて定義され、ストロークの任意の部分の開始及び／又は終了を記述するために特別なパラメータを使用することができる。

30

【0696】

本発明の実施の形態は、ローカルなストロークオブジェクトを動的に操作するのと同様に、リモートな（リモートに位置する）ストロークオブジェクトを、例えばリモートホストとの接続／結合状態に応じて動的に操作することのできるインクデータ200を生成する方法に向けられたものである。

40

【0697】

第4の実施の形態の記述

図73は、本発明の実施の形態による、インクデータ200が使用されるシステムの全体を説明する図である。図1に記述したシステムと比較すると、図73のシステムは、追加で、装置10-1-2及び装置10-3によってアクセスされるアプリケーションサービス#2をサポートするサーバ#2を含んでいる。図73のアプリケーションサービス#1及びアプリケーションサービス#2はともに、インクデータ交換インフラ10を介して、インクデータ200を利用し交換することができる。図73では、装置10-1-1は筆圧データを出力可能に構成されたペン型の入力装置であり、アプリケーションサービス#1によって提供されるアプリケーション300-1を用いてインクデータ200を生成

50

する。アプリケーション 300-1 は、TCP/IP ライブラリと、本発明のインクデータ生成方法を実行するインクデータ処理部 100 及びグラフィック処理部 300（図示せず）のためのライブラリとにリンクする。生成されたインクデータ 200 はその後、宛先であるメディア（例えば、ネットワーク）に対応する適切な出力フォーム（例えば、パケット）で出力され得る。

【0698】

装置 10-1-2 は、ユーザの指による手書き入力を受信可能に構成されたタブレット型の入力装置である。装置 10-1-2 のセンサは、筆圧データを出力する能力を有しないが、アプリケーションサービス #2 によって提供されるアプリケーション 300-2 を用いてインクデータ 200 を生成することができる。アプリケーション 300-2 は、TCP/IP スタックのようなライブラリと、本発明のインクデータ生成方法を実行するサーバ #2 上のインクデータ処理部 100 のためのライブラリとにリンクし、又はこれらを利用する。生成されたインクデータ 200 はその後、宛先であるメディア（例えば、ネットワーク）に対応する適切な出力フォーム（例えば、パケット）で出力され得る。

【0699】

装置 10-3 は、アプリケーションサービス #2 に加入しているデスクトップ型の PC である。装置 10-3 は、アプリケーションサービス #2 によって提供されるアプリケーション 300-2 を用いて、装置 10-1-1 又は装置 10-1-2 から出力されるインクデータ 200 を処理する（例えば、ディスプレイスクリーン上にレンダリングしたり、再配布する）ことができる。アプリケーション 300-2 は、TCP/IP スタックのようなライブラリと、本発明のインクデータ生成方法を実行するサーバ #2 上のインクデータ処理部 100 のためのライブラリとに動的にリンクし、又はこれらを利用する。

【0700】

図 74 は、本発明の実施の形態によるインクデータ処理部 100 のブロック図である。このインクデータ処理部 100 は、図 6 に示したインクデータ処理部 100 に対応する。インクデータ処理部 100 は、図 6 に示したグラフィック処理部 300 を利用する描画アプリケーション 300-1 のようなアプリケーションに動的又は静的にリンクされるライブラリとして実装され得る。インクデータ処理部 100 は、インクデータ生成部 120 及びインクデータ整形部 140 を含む。インクデータ生成部 120 は一般に、1) ペンイベント型の入力情報（「入力 1」）、2) コンテキスト情報（「入力 2」）、及び 3) 操作情報（「入力 3」）という 3 つのタイプの情報を入力し、含み、受信する。

【0701】

入力 1。

「ペンイベント型の入力データ」又は単に入力データ又はペンイベントデータは、ペンタブレットセンサのような入力装置からデータを取得するための OS、デバイスドライバ、API から入力される。入力データは、図 92 の左手側に図示するように、様々な種類の入力装置からのものであり得る。入力データは、入力装置からの生データに限られず、InkML 及び ISF データのような、生データを処理することによって生成されるペンイベントデータを含むことができる。

【0702】

入力 2。

コンテキスト情報は、上述したペンイベント型の入力データの入力をサポートするために使用されるコンテキストを示す。コンテキスト情報は、例えば、ストロークに関する日時情報（例えば、いつストロークが入力されたか）、ペンタイプ、ペン色、ペン ID、著者 ID、入力装置の解像度及びサンプリングレートなどを含むことができ、これらはストロークを生成するために使用されるアプリケーション（アプリケーション 300-1）又は OS によって提供される。

【0703】

入力 3。

操作情報は、入力されるべき次のストロークが、通常のストロークオブジェクト 210

10

20

30

40

50

である代わりに、既存のストロークオブジェクト 210 を操作するために使用される操作オブジェクトを構成するためのものであることを示すコマンドである。そのようなコマンドは、入力装置に関連付けられたスイッチ又はボタンのユーザ起動によって入力されることができ、アプリケーション 300-1 からインクデータ生成部 120 に提供される。例えば、ユーザが既存のストロークオブジェクト 210 を「スライス」することを望むとき、ユーザは操作情報を発行し、既存のオブジェクトをスライスするための手書き動作を行う。操作情報の観点では、インクデータ生成部 120 は、ユーザのスライス動作に基づいて他のストロークを描画する代わりに、既存のストロークオブジェクト 210 をスライスするためにユーザのスライス動作を使用する。

【0704】

10

インクデータ生成部 120 は、これら 3 つのタイプの情報（入力 1、入力 2、及び入力 3）を入力／受信し、第 3 の実施の形態を参照して上述した図 48B に示したように、ストローク言語（SL）の定義に従う一群のオブジェクトを生成する。

【0705】

インクデータ整形部 140 は、様々なオブジェクトを含むインクデータを生成するインクデータ言語操作部 120 とは切り離されている。図 74 では、インクデータ生成部 120 によって生成された一群のオブジェクトが、下から上、すなわち出力の方向で、「記録フォーマット」又は「伝送フォーマット」でデータを出力するインクデータ整形部 140 に入力される。図 74 では、記録フォーマット又は伝送フォーマットで書かれたデータが、上から下、すなわち入力の方

20

向で、一群のオブジェクトを再生し、再生した一群のオブジェクトをインクデータ生成部 120 に提供するインクデータ整形部 140 に入力される。以下の図では、SL はストローク言語（図 48B 参照）を意味し、SFF は記録フォーマットの一種であるストロークファイルフォーマットを意味し、SMF は伝送フォーマットの一種であるストロークメッセージフォーマットを意味する。

【0706】

図 75 は、本発明の様々な実施の形態による、図 74 に示したインクデータ処理部のより詳細な機能ブロック図である。この図におけるインクデータ処理部 100 は、図 6 に示したインクデータ処理部 100 に対応する。

【0707】

インクデータ生成部 120 は、ストロークオブジェクト 210 の操作部 122、メタデータオブジェクト操作部 124、レンダリング（描画スタイル）オブジェクト操作部 126、及び操作オブジェクト操作部 128 を含む。

30

【0708】

ストロークオブジェクト操作部 122 は、入力（入力 1）としてペンイベント型の入力データを受信するとともにコンテキスト情報（入力 2）を参照して、ストローク言語の核を構成するストロークオブジェクト 210 を生成する。

【0709】

メタデータオブジェクト操作部 124 は、ペンイベント型の入力データ（入力 1）及びコンテキスト情報（入力 2）に基づき、ストロークオブジェクト 210 を描写するメタデータオブジェクト 250 を生成する。メタデータオブジェクトは、日時情報、著者 ID、及びペン ID のような、スクリーン上に描画されるときにストロークオブジェクト 210 の外観に影響を与えないストロークオブジェクト 210 についての非描画関連情報を含む。

40

【0710】

レンダリング（描画スタイル）オブジェクト操作部 126 は、ストロークオブジェクト操作部 122 によって生成されたストロークオブジェクトに基づくとともにコンテキスト情報（入力 2）を参照して、ストロークオブジェクト 210 のレンダリング（描画）を制御し、スクリーン上にレンダリングされたときにストロークオブジェクト 210 がどのような現れるかを定義する描画スタイルオブジェクト 230 を生成する。

【0711】

50

操作オブジェクト操作部 128 は、操作情報（図 75 の「入力 3」）の受信に応じて、ローカルに（「ローカル」）、又は、ネットワークの向こうにリモートに（「リモート」）存在する既存のストロークオブジェクト 210 の状態を操作するために構成される操作オブジェクト 270 を生成するために、入力 1 として受信される次の「ストローク」を使用する。

【0712】

したがって、インクデータ生成部 120 は、3 つのタイプの入力情報（入力 1、入力 2、及び入力 3）に基づいて、図 48B に示したように、ストローク言語に基づく一群のオブジェクトを生成する。

【0713】

インクデータ整形部 140 は、SFF、InkML、JPEG フォーマットのような記録フォーマットでファイルを出力するように構成される記録フォーマットデータ処理部 142 と、SMF フォーマットのような伝送フォーマットでメッセージを出力するように構成されたインクデータ通信部 144 とを含む。これらの様々なフォーマットの定義に従うストローク言語によって定義されるデータは、SFF 構造記述ファイル（スキーマファイル）（F142-1-1）、SVG 構造記述ファイル（F142-2-1）、及び SMF 構造記述ファイル（F144-1）などの形で出力される。したがって、ストローク言語に従って様々なオブジェクトを生成し再生することが可能になり、生成されたオブジェクトは、様々な記録フォーマット及び／又は伝送フォーマットで入出力され得る。図示したファイルインクデータ生成部 140 は 2 つの記録フォーマット SFF 及び SVG をサポートしているが、2 つ以上の記録フォーマットをサポートしてもよいし、アプリケーション 300-1 がすべての可能なファイルフォーマットをサポートする必要があるときには 1 つだけ記録フォーマットをサポートすることとしてもよい。

【0714】

操作オブジェクト操作部 128 からの出力（例えば、図 75 の操作オブジェクト 270）は SMF メッセージ内に配置され、リモートに存在する 1 以上の既存のストロークオブジェクト 210 を操作する（例えば、スライスする）ためにネットワークを介して伝送され得る。

【0715】

<言語オブジェクトを生成する方法>

図 76～図 83B は、本発明の実施の形態による、ストロークオブジェクト操作部 122、メタデータオブジェクト操作部 124、レンダリング（描画スタイル）オブジェクト操作部 126、及び操作オブジェクト操作部 128 の構造及び動作を説明する機能ブロック図及びフローチャートである。

【0716】

<SM（ストロークモデル）。ストロークオブジェクトを生成する装置／方法>

図 76 は、図 75 に示したストロークオブジェクト操作部 122 の機能ブロック図である。ストロークオブジェクト操作部 122 は、ペンイベントデータ（「入力 1」）、又は InkML データのようなペンイベント型のデータを入力し、ポイントオブジェクトを出力するストロークモデル処理部（122_1）を含む。各ポイントオブジェクトはポイントの x 及び y 座標を含み、また、上記第 3 の実施の形態を参照して記述したように、ポイントの属性として半径及びアルファの値を含むことができ、そうすることによって、出力されるポイントオブジェクトは（x, y, 半径、アルファ）と表現され得る。ストロークオブジェクト操作部 122 は、コンテキスト情報（「入力 2」）を受信し、ポイントオブジェクトの生成での使用のためにストロークモデル処理部（122_1）に対して出力されることになるコンフィギュレーションを準備するモデル生成部（122_2）を含む。ストロークオブジェクト操作部 122 はさらに、連帯して 1 つのストロークを構成する複数の生成されたポイントオブジェクトを、1 つのストロークオブジェクト 210 に組み立てるストロークモデルビルダ部（122_3）を含む。ストロークオブジェクト操作部 122 はその後、複数の生成されたストロークオブジェクトを出力する。

【0717】

図77Aは、ストロークオブジェクト210を生成する処理を説明するフローチャートである。S122_1では、モデル生成部122_2がコンテキスト情報（「入力2」）を処理し、入力レートのような、コンフィグレーションを定義するために使用されるパラメータを取り出す。S122_6では、規定のコンフィグレーションがストロークモデル処理部122_1にロードされ、ポイントオブジェクトを生成するために使用される。もしこのコンフィグレーションがストロークモデル処理部122_1に対し平滑化を適用することを命令していたら、ストロークモデル処理部122_1はS122_2において、平滑化されたポイントオブジェクトを生成するためにポイントの配列に対して平滑化を適用する。例えば、適切なウィンドウサイズ、平滑化データ因数、及び方向平滑化因数によって構成される二重指数平滑アルゴリズムが使用され得る。もしこのコンフィグレーションが平滑化が適用されないことを命令していたら、S122_2はスキップされる。もしこのコンフィグレーションがストロークモデル処理部122_1に対し追加のポイントを生成することを命令していたら、ストロークモデル処理部122_1はS122_3において追加のポイントを生成し、それらをストロークの始まり及び／又は終わりに付加する。S122_2及びS122_3のようなステップは、モデル生成部122_2内で処理されるコンテキスト情報に応じてS122_7の前に実行される。他の例としては、ストロークオブジェクト210を構成するポイントオブジェクトの位置の値が、S122_6でロードされたコンフィグレーション内に特定される補間方法（例えば、キャットマルロームスプライン法）に依存することとしてもよい。ステップS122_7では、ストロークモデル構築部122_3が生成されたポイントオブジェクトを組み立てて新たなストロークオブジェクト210を形成し、既存のストロークオブジェクト210を更新する。

【0718】

図77Bは、追加のポイントが生成され、ストロークの始まり及び／又は終わりに付加される図77AのS122_3の詳細なアルゴリズムを記述している。示されるように、処理されるポイントのフェイズに応じて、異なるアルゴリズムが実行される。ポイントのフェイズは、そのポイントが属するストロークに対するポイントの位置を示す。例えば、もしフェイズが、そのポイントがストロークの開始点であることを示す「始まり」である（「終わり」は相似のもので、ストロークの終了点を示す）ならば、S122_4（「終わり」フェイズに対してはS122_5）において、1以上（例えば3）のポイントが生成され、ストロークの始まり（終わり）を構成するために付加される。図77AのS122_7と同様、S122_4（S122_5）で生成され追加されたポイントオブジェクトの位置の値が、S122_6でロードされたコンフィグレーション内に特定される補間方法（例えば、キャットマルロームスプライン法）に依存することとしてもよい。もしフェイズがポイントがストロークの中間部にあることを示す「中間」であれば、そのポイントのための追加のポイントは生成されない。

【0719】

<MD（メタデータ）。メタデータオブジェクトを生成する装置／方法>

図78は、図75に示したメタデータオブジェクト操作部124の機能ブロック図である。メタデータオブジェクト操作部124は、コンテキスト情報（「入力2」）を受信し、そこから著者ID、位置などのようなメタデータを取り出す。そのようなメタデータの少なくともいくつかはコンフィグレーション内に配置され、ストロークオブジェクト操作部122から受信されるストロークオブジェクト210をコンフィグレーションに従って処理するメタデータ処理部124_2に送られる。メタデータオブジェクト250は、受信されたストロークオブジェクト210を描写するために生成されるものである。メタデータ処理部124_2は、受信したストロークオブジェクト210から時刻情報のようなメタデータを取り出し、取り出したメタデータをメタデータ構築部124_3に送る。メタデータ生成部124_1はまた、コンテキスト情報から取り出した静的なコンフィグレーションをメタデータ構築部124_3に送る。典型的には、静的なコンフィグレーションは絵の全体に共通である。メタデータ構築部124_3は、メタデータ生成部124_

1 及びメタデータ処理部 1 2 4 __ 2 から受信されるメタデータに基づいてメタデータオブジェクト 2 5 0 を構築する。

【0 7 2 0】

図 7 9 は、メタデータオブジェクト 2 5 0 を生成する処理を説明するフローチャートである。S 1 2 4 __ 1 では、メタデータ生成部 1 2 4 __ 1 及びメタデータ処理部 1 2 4 __ 2 がそれぞれの入力から、ペン I D、タイムスタンプなどのようなメタデータを取り出す。S 1 2 4 __ 2 では、アプリケーションコンテキスト情報として取り出されたメタデータがメタデータ構築部 1 2 4 __ 3 にロードされる。S 1 2 4 __ 3 では、処理されるポイントのフェイスが判定される。もしそのポイントがストロークオブジェクト 2 1 0 の開始点であったならば、メタデータオブジェクト 2 5 0 が生成され、そのポイントと関連付けられる。典型的には、ストロークオブジェクト 2 1 0 ごとに必要なメタデータオブジェクト 2 5 0 は 1 つだけであるから、開始点以外の残りのポイントについては、メタデータオブジェクト 2 5 0 は生成され関連付けられる必要はない。

10

【0 7 2 1】

< R (ラスター化)。レンダリング (描画スタイル) オブジェクトを生成する装置 / 方法 >

図 8 0 は、図 7 5 に示したレンダリング (描画スタイル) オブジェクト操作部 1 2 6 の機能ブロック図である。レンダリングオブジェクト操作部 1 2 6 は、コンテキスト情報 (「入力 2」) を受信し、最小 / 最大半径情報、最小 / 最大速度情報、最小 / 最大圧力情報、色情報などの情報を取り出す。そのような情報の少なくともいくつかはコンフィグレーション内に配置され、ストロークオブジェクト操作部 1 2 2 から受信されるストロークオブジェクト 2 1 0 をコンフィグレーションに従って処理するスタイル処理部 1 2 6 __ 2 に送られる。描画スタイルオブジェクト 2 3 0 は、受信されたストロークオブジェクト 2 1 0 をどのように描画 (レンダリング) するかを定義するために生成されるものである。スタイル処理部 1 2 6 __ 2 は、受信したストロークオブジェクト 2 1 0 から、(可変の) 半径、(可変の) 色、(可変の) アルファ (透明度)、及びアンチエイリアシングパラメータのようなスタイルに関係するパラメータを取り出し、取り出した情報をスタイル構築部 1 2 6 __ 3 に送る。スタイル生成部 1 2 6 __ 1 はまた、コンテキスト情報から取り出した静的なコンフィグレーションをスタイル構築部 1 2 6 __ 3 に送る。典型的には、静的なコンフィグレーションは絵の全体に共通である。例えば、1 つの絵の中のすべてのストロークが同じ合成混合モードを有していたら、そのモードの値は静的なコンフィグレーションである。

20

30

【0 7 2 2】

図 8 1 は、スタイルオブジェクトを生成する処理を説明するフローチャートである。ステップ S 1 2 6 __ 1 では、スタイル処理部 1 2 6 __ 2 が、アプリケーションから利用可能なストロークオブジェクト 2 1 0 及びコンテキスト情報に基づいて、圧力データ、タイムスタンプデータ、位置データなどのような入力特性を決定する。S 1 2 6 __ 2 では、ステップ S 1 2 6 __ 1 で決定された特性に応じて、適切なコンフィグレーションがロードされる。例えば、もしアプリケーション内で選択されたツールがボールペンであり、かつ、入力特性が圧力データを含むのであれば、圧力ベースのボールペンのためのコンフィグレーションがロードされる。他の例としては、もし選択されたツールがボールペンであり、かつ、入力特性が圧力データを含まずタイムスタンプを含むのであれば、速度ベースのボールペンのためのコンフィグレーションがロードされる (タイムスタンプ情報から速度が導出できるため)。ステップ S 1 2 6 __ 3 では、幅 (又は色) が可変であるか否かを判定するために、ロードされたコンフィグレーションが検査される。その後、ステップ S 1 2 6 __ 4 において、幅 (又は色) がポイントごとに可変か、すなわち、ストロークがその長さに沿って変化する幅又は色を有するかが判定される。もし「y e s」であれば、S 1 2 6 __ 5 において、カスケードリングスタイルプロパティである半径 (及び / 又は色) がポイントオブジェクトごとに生成される。さもなければ、ステップ S 1 2 6 __ 6 において、カスケードリングスタイルプロパティがストロークオブジェクト 2 1 0 のために生成される

40

50

。

【0723】

オプションのカスケーディングプロパティのための他の例は、強化プロパティである。強化プロパティは、入力装置がダウン状態で静止しているとき、特別なインクの流出量をシミュレートするために使用される。このプロパティに対する実世界の類似物は、柔らかい紙の上における水彩筆の振る舞いである。水彩筆が静止位置にあるとき、柔らかい紙は絵の具を吸い込み、したがって接触点はより濃く大きくなる。ステップS126__7では、もし入力されたストロークオブジェクト210がS126__2でロードされたコンフィグレーションによって特定される強化条件を満たすならば、各ポイントオブジェクトのために強化プロパティが生成される。例えば、もしユーザがダウン状態でいかなる動きも生じさせず、かつ、ロードされたコンフィグレーション内で強化アルゴリズムが活性化されていたならば、ステップS126__7では、現在の（最後に生成された）ポイントが、特定の位置におけるポイント密度を増加させるために複製又は更新される。

10

【0724】

S126__8の前に、S126__2でロードされたコンフィグレーションに応じて、先に説明したステップと同様の他のステップが実行され得る。ステップS126__8では、スタイルオブジェクトが生成される。ステップS126__5, S126__6, 及びS126__7他で更新されるカスケーディングスタイルプロパティのすべてが、S126__8でスタイルオブジェクト内に組み入れられる。

20

【0725】

<M（操作）。操作オブジェクトを生成する装置／方法>

図82は、図75に示した操作オブジェクト操作部128の機能ブロック図である。操作オブジェクト操作部128は、コンテキスト情報（「入力2」）を受信して処理することにより操作プロセッサ128__2による使用のためにコンフィグレーションを準備する操作者生成部128__1を含む。操作プロセッサ部128__2は、操作者生成部128__1から受信されるコンフィグレーションと、アプリケーションから受信される操作情報（「入力3」）を参照してペンイベントデータ（「入力1」）を処理し、それによってスライスの実体のような操作の実体を生成する。操作ビルダ128__3は、スライスの実体のような操作の実体を受信し、スライスオブジェクト274のような操作オブジェクトを構築する。操作オブジェクトは、既存のストロークオブジェクト210に対し、規定の動作を実行するように構成される。例えば、スライスオブジェクト274は、既存のストロークオブジェクト210を2つのスライスピースにスライスするために使用される。操作ビルダ128__3によって生成された操作（例えばスライス）オブジェクトのコレクションは、リモートに存在する既存のストロークオブジェクト210上で実行されるために、ネットワークを介して送られることができ、又は、ストロークオブジェクト操作部122によって生成され格納された既存のストロークオブジェクト210上でローカルに実行されることができる。図示するように、操作プロセッサ128__2はまた、操作の実体を生成するためのどれかに基づいて、ストロークオブジェクト操作部122からストロークオブジェクトを受信することができる。

30

【0726】

図83Aは、操作オブジェクトを生成する処理を説明するフローチャートである。ステップS128__1では、圧力、位置、及びタイムスタンプ情報のような入力特性が入力1から取り出され、また、操作情報（入力3）が受信される。ステップS128__2では、取り出された入力特性及びアプリケーションのコンテキスト情報（「入力2」）から決定される適切なコンフィグレーションがロードされる。ステップS128__3では、操作オブジェクトを形成するために、ロードされたコンフィグレーションに従って新たな入力が処理される。例えば、もし新たな入力が多角形の形状であったら、その多角形が操作領域を定義する。例えば、もし生成される操作オブジェクトが投げ縄ツールであったら、その多角形の形状が投げ縄ツールの範囲を定義する。S128__4では、複数の既存のストロークオブジェクトの中から1つの既存のストロークオブジェクト210が選択され、S1

40

50

28__5では、選択された既存のストロークオブジェクト210と、操作領域（例えば多角形の形状）との間の交点が算出される。もし、S128__6で交点が発見されなければ、他の既存のストロークオブジェクト210が選択され、既存のストロークオブジェクト210との交点が少なくとも1つ見つかるまで、ステップS128__4、S128__5、及びS128__6が繰り返される。既存のストロークオブジェクト210と操作領域との間の交点が発見された場合、S128__7において、既存のストロークオブジェクト210を交点で「スライス」するように構成されるスライシング操作オブジェクトが生成される。S128__14では、操作領域が交差し得る既存のストロークオブジェクト210がもっとあるかが判定される。もし「yes」であれば、処理はステップS128__4に戻る。既存のストロークオブジェクトのすべてについて操作領域とのそれらの交点がチェックされると、ステップS128__15において、生成されたスライスオブジェクトがスライスオブジェクトのコレクションに組み上げられる。

10

【0727】

図83Bは、スライスオブジェクトを生成する処理を説明するフローチャートである。S128__8では、S128__2でロードされたコンフィグレーションが「操作精度」を判定するために使用される。もし操作精度が「あるがままのストローク」であれば、S128__9において、既存のストロークオブジェクト210をスライスして2つのスライスチップを生成するスライスオブジェクト274が生成され、2つのピースはそれぞれ、既存のストロークオブジェクト210を構成する元のポイントオブジェクトのサブセットにより定義される。別の言葉で言えば、操作領域との交点が、既存のストロークオブジェクト210の隣接する2つのポイントオブジェクトの間に位置しているときであっても、交点の正確な位置が2つのスライスチップを定義するために使用されることはない。

20

【0728】

もし操作精度が「正確なポイント」であれば、S128__10において、隣接する2つのポイントオブジェクトの間の交差セグメント（曲線）が見出される。ここで、交差セグメントとは、操作領域が既存のストロークオブジェクト210と交差するところである。1以上の交差セグメントが見出される。S128__11では、各交差セグメントが、交点の正確な位置を見つけるために、例えば補間法を用いて処理される。S128__12では、各交差セグメントに対して、それぞれ終了点又は開始点として交点の正確な位置を有する2つのスライスチップが生成される。S128__13では、生成されたスライスチップが更新されてそれぞれ新たなストロークオブジェクトとなり、新たに生成されたストロークオブジェクトのそれぞれがメタデータオブジェクト250及び描画スタイルオブジェクト230に関連付けられる。

30

【0729】

上述したように、インクデータ処理部100と、その関連する方法とは、ストロークオブジェクトとともに、メタデータオブジェクト250、ラスター化描画スタイルオブジェクト230、操作オブジェクト270を含む、ストロークオブジェクト210に関連付けられるオブジェクトを生成する。

【0730】

以下、インクデータ整形部140の構成及び動作について、図84～図90Cを参照しながら詳細に記述する。

40

【0731】

図84は、図75に示したインクデータインクデータ整形部140の機能ブロック図である。図75を参照して説明したように、インクデータ整形部140は、SFF、InkML、及びJPEGフォーマットのような記録フォーマットでファイルを出力する記録フォーマットデータ処理部142と、伝送フォーマットで各種のオブジェクト（ストロークオブジェクト210、メタデータオブジェクト250、描画スタイルオブジェクト230、及び操作オブジェクト270）を出力するように構成されたインクデータ通信部144とを含む。したがって、ストローク言語に従って様々なオブジェクトを生成し再生することが可能になり、生成されたオブジェクトは、様々な記録フォーマット及び／又は伝送フ

50

フォーマットで入出力され得る。

【0732】

記録フォーマットデータ処理部142は、記録フォーマット内に、ストロークオブジェクト、メタデータオブジェクト、描画スタイルオブジェクトを配置するように構成される。サブセクション142-1, 142-2, ...は、それぞれの出力ファイルフォーマットに従ってオブジェクトを整えるように構成された処理部である。

【0733】

インクデータ通信部144は、スライスオブジェクト274のような操作オブジェクトを、ネットワークを介してリモート装置に（リアルタイム）伝送することに適したストロークメッセージフォーマット内に配置する。伝送フォーマット内に配置される操作オブジェクトは、ローカルに存在し、又は、ネットワークを関してリモートに存在する既存のストロークオブジェクト上で実行され得る。

【0734】

図85は、インクデータ整形部140において実行される処理を説明するフローチャートである。まず、オブジェクトを送信する必要があるかが判定される。この判定は、現時点でローカルに組み立てられたストロークオブジェクト210がリモート端末に共有されるか否か、ということに基づいて実行される。もし肯定なら、オブジェクトはステップS144において、ネットワークを介して伝送されるストロークメッセージフォーマット（SMF）内に配置される。これに対し、もしオブジェクトが伝送されないと判定されれば、オブジェクトは、複数の記録フォーマットの中から選択される適切な記録フォーマット内に配置される。もしストロークファイルフォーマット（SFF）が出力フォーマットとして使われるのであれば、ステップS142-1において、SFF生成処理が実行される。もしSVGやJPEGフォーマットのような他のフォーマットが使われるのであれば、ステップS142-1において、例えば、SVGの拡張に基づくストローク言語情報を含むSVGフォーマットのファイルを出力するために、SVG生成処理が実行される。

【0735】

図86は、ストロークファイルフォーマット（SFF）のデータを出力する処理を説明するフローチャートである。S142-1-1では、SFF構造体を生成するために、SFF構造記述ファイル（F142-1-1）が分析される。SFF構造体は、インターフェイス記述言語を用いて記述される。SFFを使用する各ソフトウェアアプリケーションは、SFFデータを適切に扱うために、記述ファイルを理解（分析）する必要がある。例えば、もしSFF構造体がプロトコルバッファIDLを用いて表現されていたら、プロトコルバッファのコンパイラによって生成されるスタブクラスがS142-1-1でロードされる。S142-1-2においては、図75のインクデータ生成部120で生成された様々なオブジェクトにより、SFF構造体を埋める。S142-1-3では、可変長整数符号化技術及び指数ゴロム符号技術のような抽象データ型の最適なメモリ表現のための各種技術を用いて、SFF構造体が処理される。S142-1-4では、生成されたSFF構造のメモリ表現がメモリストリーム内に詰め込まれる。生成されたメモリストリームは、ファイルシステム又はファイル構造内に保存され得る。

【0736】

S142-1-2は、SFFデータ構造を適切に満たすための複数のステップを含むことができる。処理対象であるインクデータに含まれる各ストロークオブジェクト210について（S142-1-5）、及び、処理対象であるそのストロークオブジェクト210内に含まれる各ポイントオブジェクトについて（S142-1-6）、圧縮動作が実行されるか否かが判定される。もし「yes」であれば、S142-1-7において、すべての浮動小数点値が固定小数点精度の値に変換され、整数として表現される。精度の損失は、ストロークオブジェクト操作部122内又はレンダリングオブジェクト操作部126内で、ポイントオブジェクトの浮動小数点値を所望の精度に丸めることによって補償される。S142-1-8では、生成された整数値に対し、データ符号化のような圧縮アルゴリズムが適用される。S142-1-9では、オブジェクト、もし圧縮が適用された場合に

は圧縮処理を受けたもの、が S F F 構造体を埋めるために使用される。

【0737】

図87は、J P E Gフォーマットのデータを出力する処理を説明するフローチャートである。処理対象であるインクデータ内に含まれる各ストロークオブジェクト210について（S142-2-1）、S142-2-2において、ストロークオブジェクト210に結合されているすべての描画スタイルオブジェクト230が読み出され、描画スタイルオブジェクト内に定義されているすべてのラスター化（描画）プロパティがロードされる（例えば、混合及びテクスチャプロパティ）。このとき、色及び半径の値のようなすべてのカスケーディングプロパティが分析される。S142-2-3では、ストロークの幾何学的形状がC P U又はG P Uを用いて生成される。S142-2-4では、生成された幾何学的形状に対し、C P U又はG P Uを用いて、色、テクスチャなどのすべてのラスター化／グラフィカル情報を適用することにより、ストロークがラスター化（描画）される。S142-2-5では、ラスター化されたすべてのストロークが一斉に組み上げられる。S142-2-6では、レンダリング（描画、ラスター化）されたすべてのストロークオブジェクト210を含むビットマップが生成される。S142-2-7では、このビットマップデータがJ P E Gアルゴリズムを用いて圧縮される。

10

【0738】

図88は、ストロークメッセージングフォーマット（S M F）のデータを出力する処理を説明するフローチャートである。S144-1では、図75に示したオブジェクト操作部122、124、126、又は128の中の1つから、オブジェクトが受信される。S144-2では、ストロークオブジェクト210、メタデータオブジェクト250、描画スタイルオブジェクト230、又は操作オブジェクトとしてオブジェクトタイプが決定される。S144-3では、オブジェクトと、処理対象であるインクデータ内の残りのオブジェクトとの間の関係を示すため、オブジェクトに対し識別子（例えば、ストロークI D、スタイルI D）が割り当てられる。S144-4では、S M F構造記述ファイル（F144-4）が分析され、決定されたオブジェクトタイプに対応するS M F構造体がロードされる。例えば、もしS M F構造体がプロトコルバッファを用いて表現されていたら、プロトコルバッファコンパイラによって生成されるスタブクラスがロードされる。S144-5では、決定されたオブジェクトに対して圧縮動作が実行されるか否かが判定される。もし「y e s」であれば、S144-6において、すべての浮動小数点値（例えば、x、y、半径、不透明度、透明度）が固定小数点精度の値に変換され、整数として表現される。S144-7では、生成された整数値に対し、データ符号化のような圧縮アルゴリズムが適用される。S144-8では、オブジェクト、もし圧縮が適用された場合には圧縮処理を受けたもの、がS M F構造体を埋めるために使用される。S144-9では、S M Fデータがメモリストリーム内に保存される。

20

30

【0739】

図89は、様々なファイルフォーマット及び伝送フォーマットで出力されたデータ（S F F／J P E G及びS M F）の入力処理を説明する機能ブロック図である。

【0740】

図89では、左手側に図示したインクデータ出力操作部140Tが、上述した出力処理を実行する。インクデータ200は、S F Fフォーマット及びJ P E Gフォーマットのような記録フォーマットで出力されるか、又は、S M Fフォーマットのような伝送フォーマットで出力される。

40

【0741】

各種フォーマットで出力されるこれらのファイル及び／又はメッセージは、その後、図89の右手側に図示したインクデータ入力操作部140Rによって入力（受信）される。様々な実施の形態において、インクデータ入力処理及びインクデータ出力処理は、同じサブセクション142-1（I N及びO U Tの両方）並びに同じサブセクション142-2（I N及びO U Tの両方）のような同じライブラリを共有する同じ処理部で実行される。

【0742】

50

インクデータ入力操作部 140R 内の記録フォーマットデータ処理部 142 は、入力されたデータからフォーマットに依存するデータを除去し、様々なタイプのインクデータオブジェクトに関する情報を取り出し、取り出したインクデータオブジェクトに関する情報を受信側のインクデータ生成部 120 に対して出力する。

【0743】

インクデータ入力操作部 140R 内のインクデータ通信部 144R は、受信したパケット又はメッセージから操作オブジェクトを取り出し、取り出された各操作オペレーションに対して、受信側のインクデータ生成部 120 内で既存のストロークオブジェクト 210 に対して実行（適用）されるよう命令する。

【0744】

図 90A は、SFF ファイル内に配置されたオブジェクトを解釈して再生するための処理のフローチャートである。S142-1 (IN)-1 では、SFF 構造体を生成するために SFF 構造記述ファイルが分析される。S142-1 (IN)-2 では、SFF 構造体が復元される。1 以上の SFF 構造体が復元され、復元された各 SFF 構造体について (S142-1 (IN)-3)、復元された SFF 構造体が圧縮されているか否かが判定される。もし「yes」であれば、S142-1 (IN)-4 において、復元された SFF 構造体が解凍され、S142-1 (IN)-5 において、整数として表現されている解凍された固定小数点値が浮動小数点表現に変換される。S142-1 (IN)-6 では、対応するストローク言語オブジェクトが生成される（例えば、ストロークオブジェクト 210、描画スタイルオブジェクト 230、メタデータオブジェクト 250）。

【0745】

図 90B は、InkML での入力に基づくオブジェクトを解釈して再生するための処理のフローチャートである。S142-2 (IN)-1 では、InkML ファイルが分析され、メモリ内にロードされる。S142-2 (IN)-2 では、トレースオブジェクトが、ポインタ入力イベントのサンプルに変換される。この処理は、位置、圧力、角度、傾き、及びタイムスタンプデータのような入力データを取り出し、取り出した入力データをポインタ入力イベントのシーケンスにモデリングすることを意味する。S142-2 (IN)-3 では、ポインタ入力イベントのシーケンスが、ストロークオブジェクト操作部 122 に渡される。ストロークオブジェクト操作部 122 はまた、InkML ファイルに含まれるデータ（例えば、圧力チャンネルがあるかないか）に基づいてコンテキスト情報を受信する。ストロークオブジェクト操作部 122 は、ストロークオブジェクトを生成する。S142-2 (IN)-5 では、メタデータオブジェクト操作部 124 が、メタデータオブジェクトを生成する。S142-2 (IN)-5 では、ラスター化（描画スタイル）オブジェクト 230 が、描画スタイルオブジェクトを生成する。

【0746】

図 90C は、SMF による操作（スライス）オブジェクトを受信して実行する処理のフローチャートである。S144-1 では、SMF で書かれたスライスオブジェクト 274 のコレクションが受信される。S144-2 では、スライスオブジェクト 274 が復元される。S144-3 では、S144-2 で復元されたスライスオブジェクト 274 により影響されるストロークオブジェクトの位置を特定するため、既存のストロークオブジェクトが検討される。S144-4 では、影響されるストロークオブジェクトが検討される。S144-5 では、影響されるすべてのストロークオブジェクト 210 が、対応するスライスオブジェクトを用いて修正（スライス）される。対応するスライスオブジェクト内であると特定されるポイント範囲内のすべてのポイントオブジェクトが削除（消去）される。S144-6 では、望むならば、1 つ又は 2 つの新たなストロークオブジェクトが生成される。例えば、もし削除されたポイントオブジェクトがスライスされたストロークオブジェクト 210 の中間にあるのであれば、元のストロークオブジェクト 210 の始まりの部分が新たなストロークオブジェクト 210 を構成し、元のストロークオブジェクト 210 の終わりの部分がもう 1 つの新たなストロークオブジェクト 210 を構成し得る。S144-8 では、影響されるストロークのスタイルプロパティが修正されるべきか否かが判

10

20

30

40

50

定される。もしスタイルプロパティが修正されるべきであるなら、S 1 4 4-9 が、新しいストロークオブジェクトに対して新しいスタイルプロパティの値を設定する。そうでなければ、S 1 4 4-7 が元のストロークオブジェクト 2 1 0 のスタイルプロパティの値を、新しく生成されたストロークオブジェクトに単にコピーする。同じ処理がメタデータにも適用される。もしメタデータが修正されるべきであるなら、S 1 4 4-1 1 が、新しく生成されたストロークオブジェクトに対して新しいメタデータを設定する。そうでなければ、S 1 4 4-1 0 が元のストロークオブジェクト 2 1 0 のメタデータを、新しく生成されたストロークオブジェクトに単にコピーする。S 1 4 4-1 2 では、元のストロークオブジェクト 2 1 0 の開始パラメータ 3 0 1 及び終了パラメータ 3 0 3 の値が、新しく生成されたストロークオブジェクトにコピーされる。上述した処理は、影響されるすべてのストロークについて繰り返される。S 1 4 4-1 3 では、現在のスクリーンに再描画する必要があるかないかを決定するための検査が実行される。もし「y e s」であれば、S 1 4 4-1 4 において、修正された領域内の、1 以上のスライスオブジェクト 2 7 4 によってスライスされたストロークオブジェクトが、スクリーン上に描画（レンダリング）される。

10

【0747】

第 1 乃至第 4 の実施の形態の効果

図 9 1 は、上述した側面 1 に関して、図 7 5 のインクデータ 2 0 0 の処理装置（1 0 1）を使用することの効果の説明する図である。本発明の様々な実施の形態による操作オブジェクトは、伝送フォーマットを用いる操作オペレーションの内容の伝送を許容し、それによって、複数の場所に位置しているストロークオブジェクトの状態を容易に同期させる。例えば、図 9 1 の左手側の 1 つの装置と、右手側の 2 つの装置（それぞれ、図 1 及び図 7 3 の装置 1 0-1-1, 1 0-1-2, 1 0-3 に対応する）とがリアルタイムコラボレーションアプリケーションを実行していると仮定する。さらに、これらの装置が、まだスライスされていない処理対象のストロークオブジェクト 2 1 0 を共有していると仮定する。すると、本発明の実施の形態によれば、以下の動作が可能になる。

20

1. まず、左手側の装置が、規定のストローク幅 W I D T H を有するストロークオブジェクト 2 1 0 に対してスライス操作オペレーションを実行する。

2. 次に、インクデータ 2 0 0 の処理装置 1 0 0（操作オブジェクト操作部 1 2 8）が、スライス操作オペレーションに基づいて操作オブジェクトを生成する。

30

3. 次に、インクデータ処理装置 1 0 1 が、そのローカルのストロークオブジェクト 2 1 0 に対してスライス操作オペレーションを実行することにより、ローカルのストロークオブジェクト 2 1 0 を修正する（図 7 5 の矢印「ローカル」参照）。この処理は、上記ステップ 2 に先行して実行されてもよいし、上記ステップ 2 と並行して実行されてもよい。

4. 次に、インクデータ処理装置 1 0 1（インクデータ通信部 1 4 4）が操作オブジェクトを S M F フォーマットに整形し、S M F データをネットワークに送信する（図 7 5 の矢印「リモート」参照）。

5. 装置 1 0-1-2 及び 1 0-3 は、S M F データである操作オブジェクトを受信して操作オブジェクトに関連付けられたストローク I D を取り出し、取り出されたストローク I D によって識別される複数のストロークオブジェクトのそれぞれに対し、操作オペレーション（スライスオペレーション）を実行する。結果として、取り出されたストローク I D によって識別されるストロークオブジェクトの（スライスされた）状態が、左手側の装置 # 1 と、右手側の装置 1 0-1-2 及び 1 0-3 との間で同期される。

40

【0748】

したがって、本発明の実施の形態によるインクデータ処理方法は、ローカルで、及び、2 つの遠隔に位置する装置間のネットワークを介してリモートで、ストロークデータを動的に操作する能力を有する。

【0749】

図 9 2 は、上述した側面 2 に関して、図 7 5 のインクデータ処理装置（1 0 1）を使用することの効果の説明する図である。

50

【0750】

図92の左手側は、入力側における装置依存型の生データを示しており、図92の右手側は、最終製品としての出力ファイル内に含まれることになるデータを示している。左手側は、ストロークを生成するために使用され得る入力データの以下に示す4つのタイプを示している。

1. 筆圧情報を取得する能力を有するタイプ1の装置によって取得される一連の点座標と一連の筆圧情報。

2. タイプ2の装置、すなわち、容量タイプのタッチセンサを組み込む装置のような比較的単純な装置によって取得される一連の点座標。

3. 例えばコンピュータグラフィックを生成するために使用されるプロフェッショナル級のハードウェア及びアプリケーションの結合によって取得される、ペンの回転角、筆圧、X方向のペンの傾き角、Y方向のペンの傾き角など様々な詳細を含むタイプNのデータ。

4. 方位、高さ、ペンの向きの情報を表現できる、InkMLのような標準化されたデータ。

【0751】

側面2を参照して上述したように、一般に、手書き入力データに基づいて再生される必要のある情報は、ペン（スタイラス）が何度の角度で保持されたか、どの程度の筆圧が印加されたか、などのような、「どのように」手書きデータが入力されたかということではない。むしろ、再生される必要のある情報は、そのようなペン操作の「結果」であり、それはペン操作によって生成された1以上のストロークを含む。したがって、できるだけ抽象化され一般化された手書き入力データを作るストロークモデルを使用すること、すなわち、可能な限り、図92の右手側で手書きデータを処理することが望まれる。そのようなストロークモデルは、様々な特殊な（抽象的でない）態様で「どのように」を記録する色々な装置の間に存在する違いを吸収することができる。

【0752】

図92の最も右手側には、インクデータ処理の最終製品としての画像ファイルに含まれるデータ構造又はファイルフォーマットを示している。図92の中間部は、本発明の実施の形態によるインクデータ200の処理の結果として得られる、テキスト化、署名認証、注釈、及びリアルタイムコラボレーションアプリケーションのような様々なアプリケーションでの使用に適した中間ベクトルデータを示している。中間ベクトルデータは、装置に依存しない態様で、すなわち、ペン入力に由来しない態様でベクトルグラフィックスを定義する既存のSVGデータ（非特許文献3）を含む。そのようなものとして、SVGは、ストローク幅、ストローク色、ストロークの透明度のようなペン由来のデータを変化又は調整することを容易には許容せず、結果として、ストロークデータをマークアップする（特徴付ける）には特に適してはいない。これに対し、本発明の実施の形態による中間ベクトルデータに基づくストローク言語（SL）は、生データから導出されるストロークオブジェクトをマークアップし、特徴付け、そのようなストロークオブジェクト上で動作する、メタデータオブジェクト、レンダリングオブジェクト、操作オブジェクトのような様々なオブジェクトを提供している。

【0753】

図93は、上述した側面3に関して、図75のインクデータ処理装置（101）を使用することの効果の説明する図である。特定のフォーマットに紐付けられてはいないが様々なフォーマットで利用できる共通のストローク言語（又は、言語の意味及び文法を定義する共通の情報モデル）の供給は、インクデータエコシステムのライフサイクルを延長する。図93では、100-1, 100-2, ..., 100-Nが、本発明の実施の形態によるインクデータ処理方法が埋め込まれた色々なアプリケーションを表している。生の「入力データ」がアプリケーション100-1に入力される時（図93の「ステップ1」）、アプリケーション100-1のインクデータ生成部120は、ストローク言語（又はストローク言語を定義する情報モデル）のオブジェクトに生データを抽象化する。オブジ

ェクトはその後、記録フォーマット又は伝送フォーマット（「第１のフォーマット」又は図示した例におけるＳＶＧ）に変換され、出力される（図９３の「ステップ２」）。アプリケーション１００－２は、レンダリング又は操作用としてストローク言語で書かれたオブジェクトを取り出すため、ＳＶＧで書かれたデータを受信して解釈する。アプリケーション１００－２は、他の記録フォーマット又は伝送フォーマット（「第２のフォーマット」又は図９３の「ステップ３－２」におけるＳＦＦ）にオブジェクトを整形することができる。ＳＦＦで書かれたデータはその後出力され、レンダリング又は操作用としてストローク言語で書かれたオブジェクトを取り出すためにＳＦＦで書かれたデータを解釈するアプリケーション１００－Ｎによって受信される。アプリケーション１００－Ｎは、さらに他の記録フォーマット又は伝送フォーマット（「第３のフォーマット」又はビットマップ）にオブジェクトを整形し、出力することができる。したがって、例えば非特許文献５における、単一のフォーマットを使用してインクデータ構造体を処理するＪＯＴと比較すると、本発明の実施の形態は、様々なフォーマットでインクデータを処理する能力を有しており、したがって、インクデータ２００のライフサイクルを延長している。図９３の図示した例では、インクデータ２００は、アプリケーション１００－１によって、アプリケーション１００－２によって、及び、最後のアプリケーション１００－Ｎを含むさらなるアプリケーションによって、使用可能とされている。

【０７５４】

上述した要素、装置、ステップ、及び処理のそれぞれを、実装及び応用のそれぞれに応じて、他の要素、装置、ステップ、及び処理と組み合わせることができ、又は、サブ要素、サブ装置、サブステップ、サブ処理にさらに分割できることが、当業者によって理解される。さらに、上記ステップ及び処理は、実装及び応用のそれぞれに応じて、単一のプロセッサで実行されることができ、又は、複数のプロセッサ内で分散的に実行されることができ、

【図１】

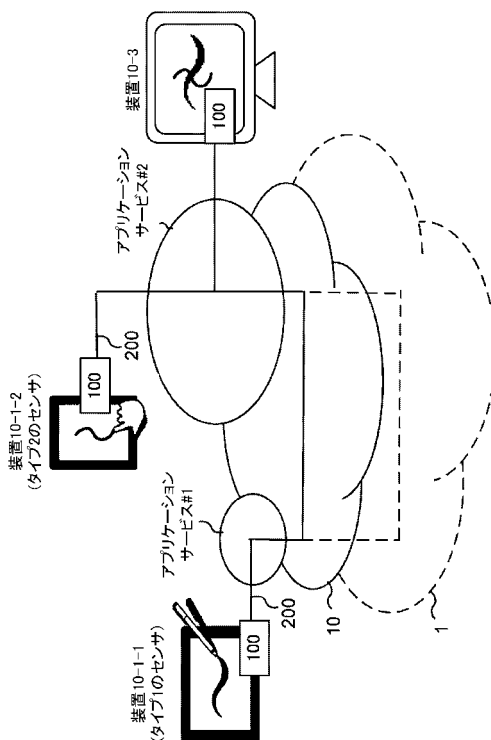


Fig. 1

【図２】

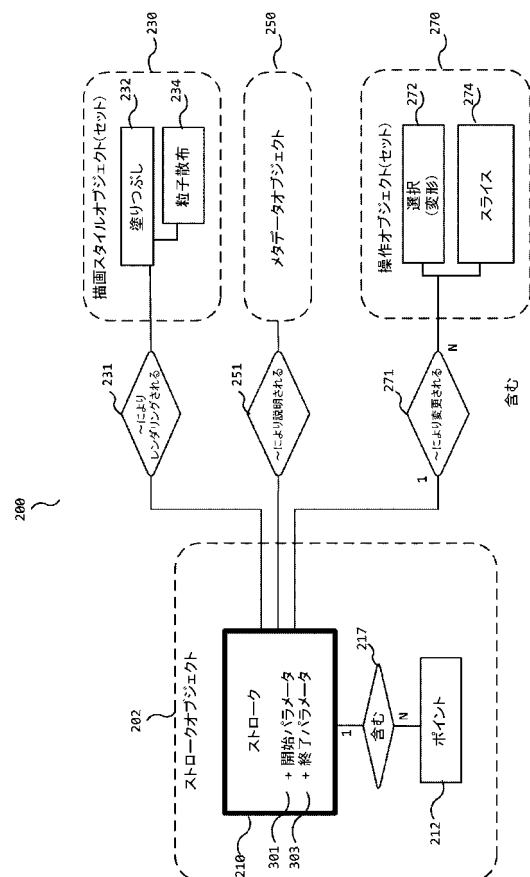


Fig. 2

【図 3 C】

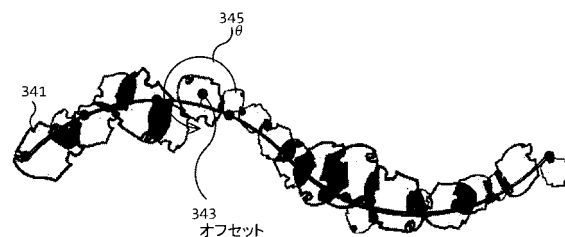


Fig. 3C

【図 4 A】

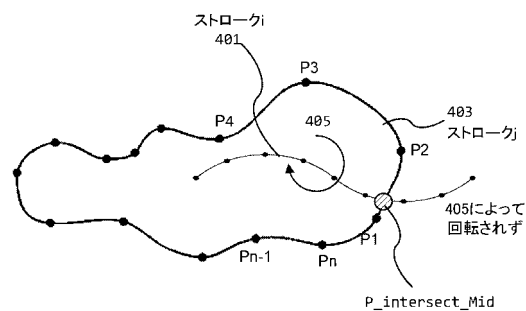


Fig. 4A

【図 5】

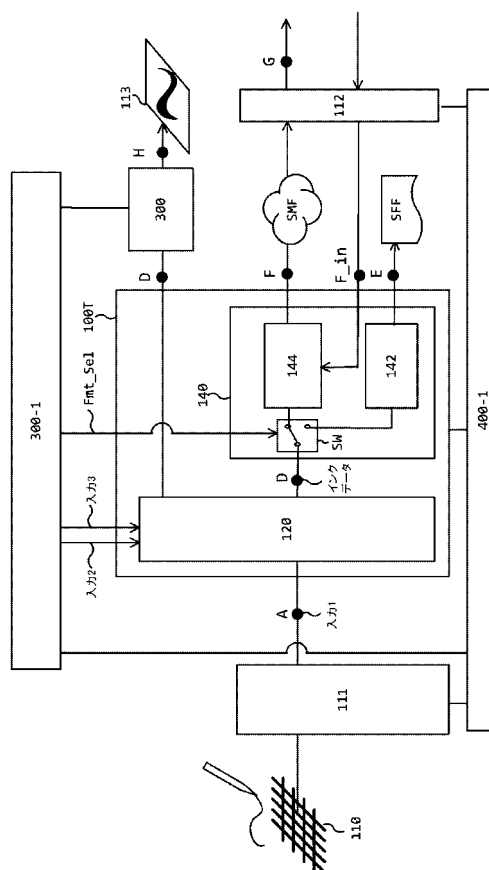


Fig. 5

【図 6】

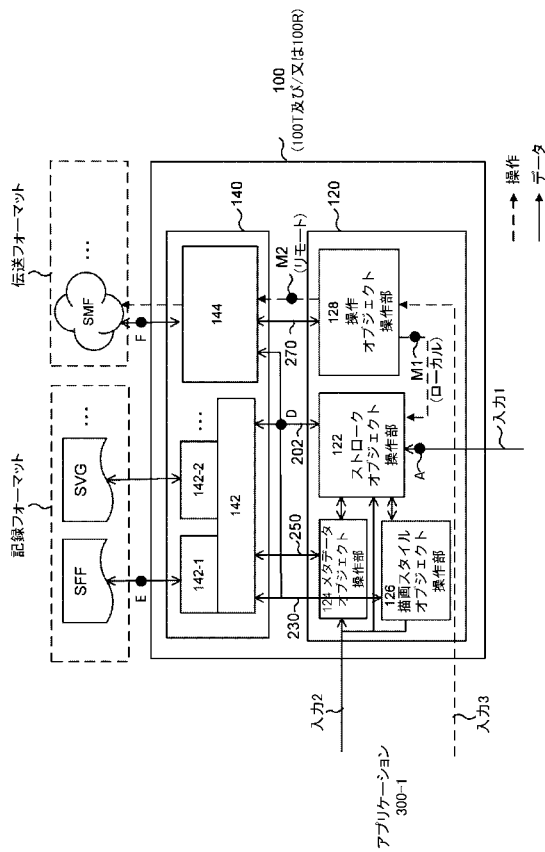


Fig. 6

【図 7】

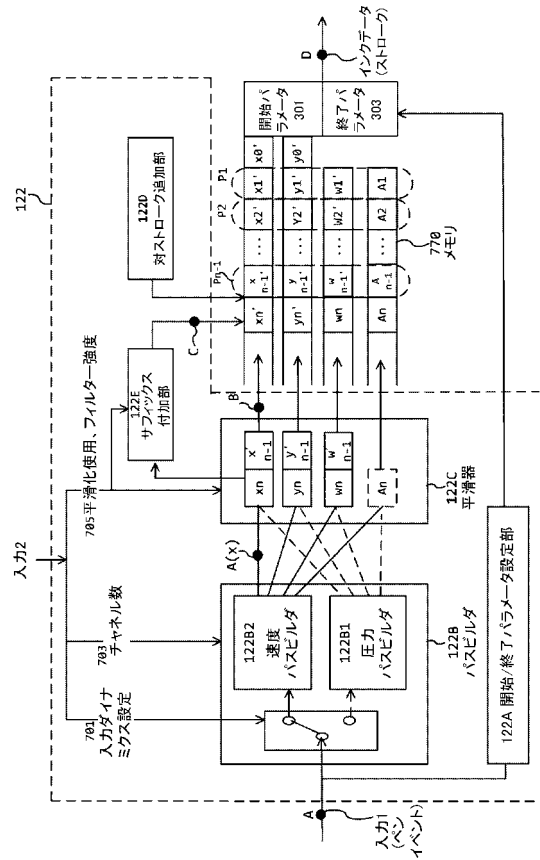


Fig. 7

【図 8】

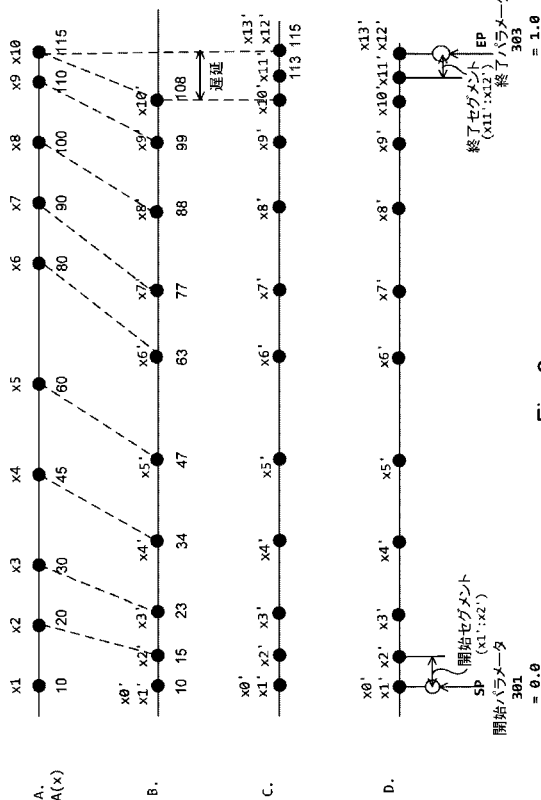


Fig. 8

【図 9】

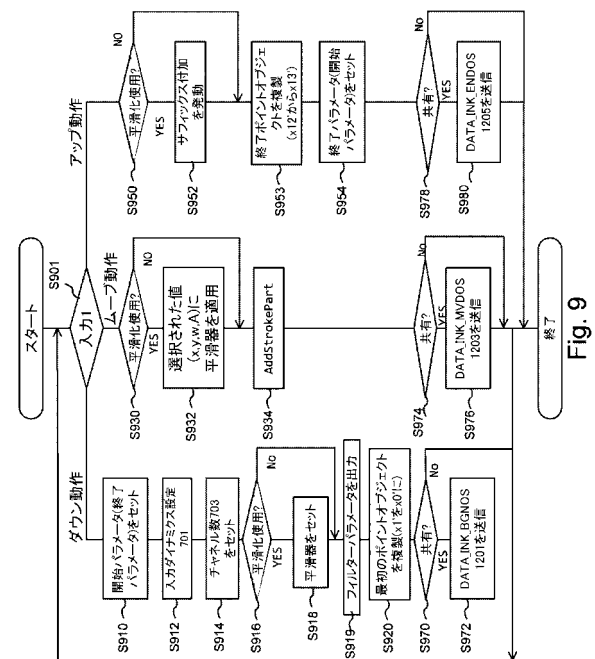


Fig. 9

【図 1 0】

```
01: package strokefileformat;
02: message Document {
03:   repeated float viewBox = 1;
04:   optional Uint32 decimalPrecision = 2 [default = 2];
05:   repeated Style styles = 3;
06:   repeated Stroke strokes = 4;
07: }

08: message Style {
09:   optional float strokeWidth = 1 [default = 4];
10: }

11: message Stroke {
12:   repeated sint32 point = 1 [packed = true];
13:   optional float startParameter = 2 [default = 0];
14:   optional float endParameter = 3 [default = 1];
15:   repeated sint32 variableStrokeWidth = 4 [packed = true];
16:   optional float strokeWidth = 5;
17: }
```

Fig. 10

【図 1 1】

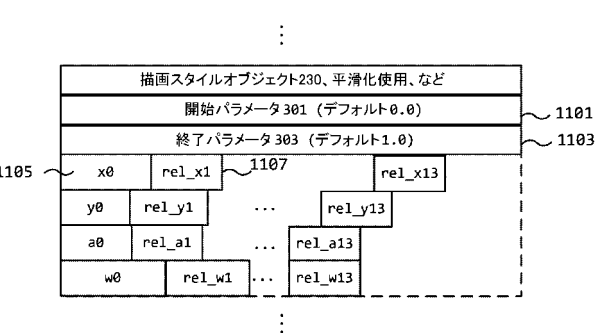


Fig. 11

【図 1 2】

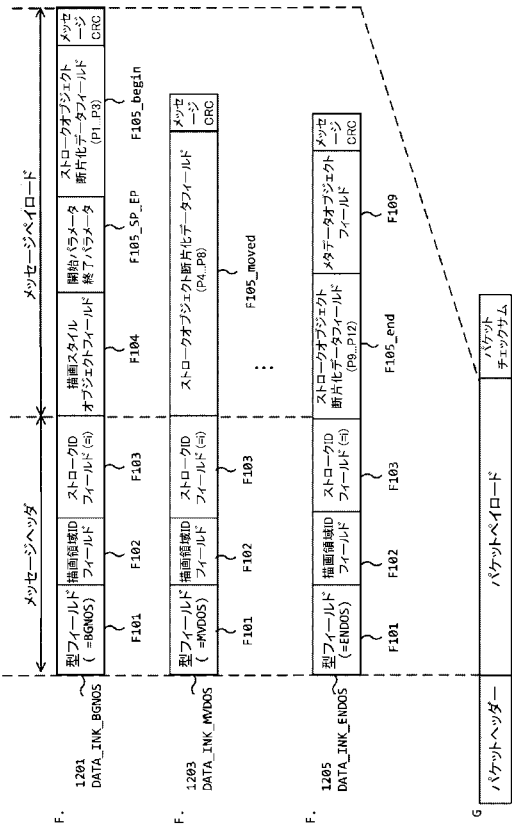


Fig. 12

【図 1 3 A】

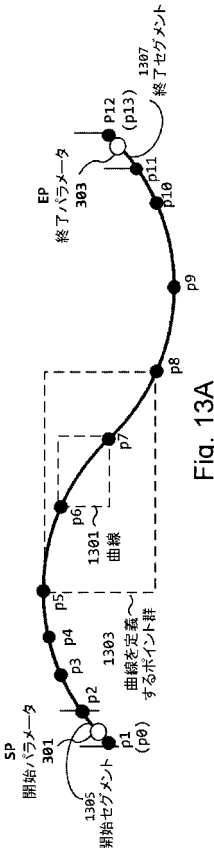


Fig. 13A

【図 13 B】

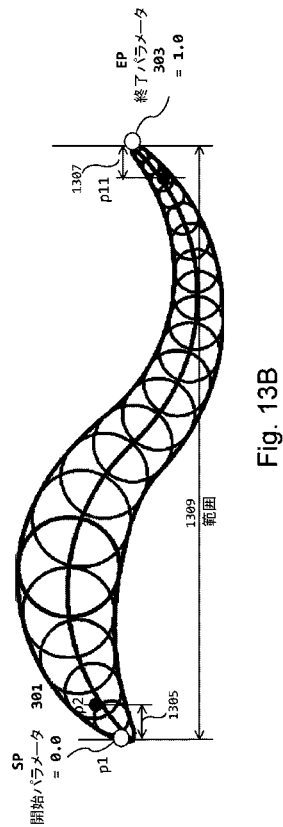


Fig. 13B

【図 14】

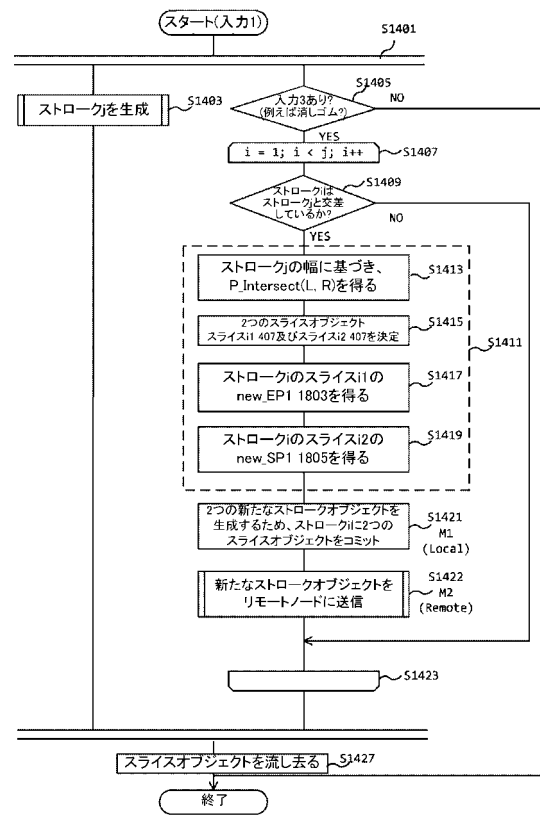


Fig. 14

【図 15 A】

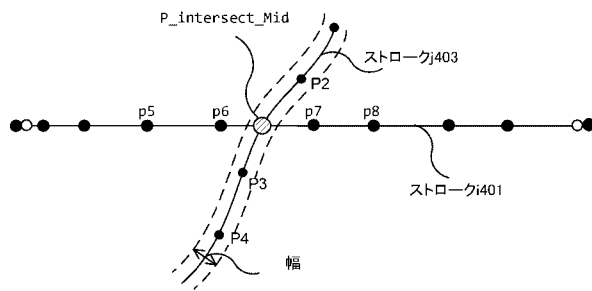


Fig. 15A

【図 15 B】

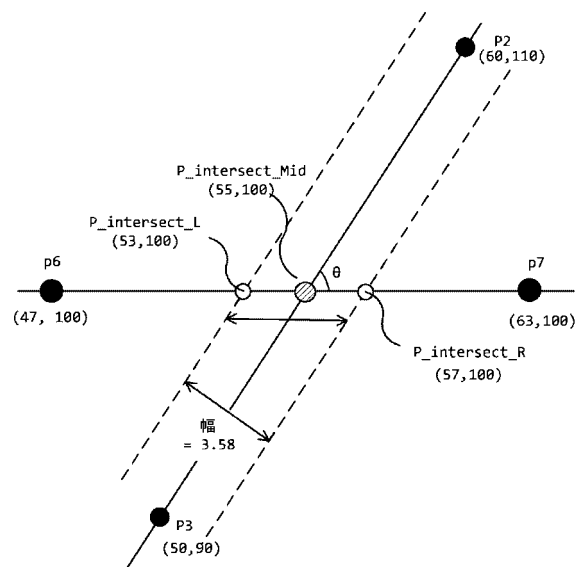


Fig. 15B

【図 16 A】

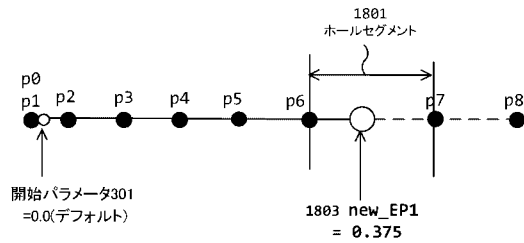


Fig. 16A

【図 16 C】

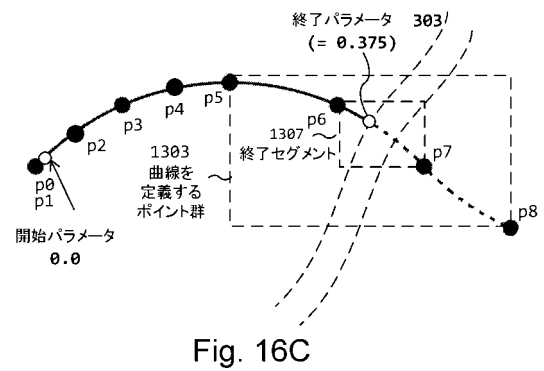


Fig. 16C

【図 16 B】

```

11: class Slice {
12:     int slice_fromIndex;           // 0
13:     int slice_toIndex;             // 8
14:     float slice_startParameter;    // 0.0
15:     float slice_endParameter;      // 0.375
16: }

```

Fig. 16B

【図 17 A】

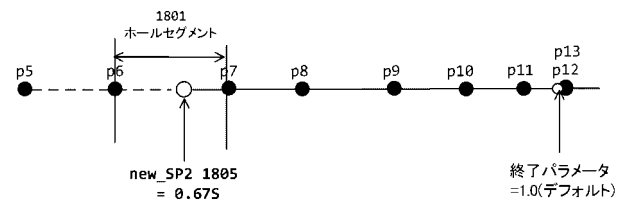


Fig. 17A

【図 17 B】

```

21: class Slice {
22:     int slice_fromIndex;           // e.g., 5
23:     int slice_toIndex;             // e.g., 13
24:     float slice_startParameter;    // e.g., 0.625
25:     float slice_endParameter;      // e.g., 1.0
26: }

```

Fig. 17B

【図 17 C】

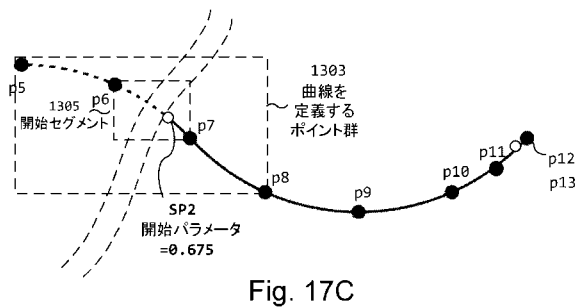


Fig. 17C

【図 18 A】

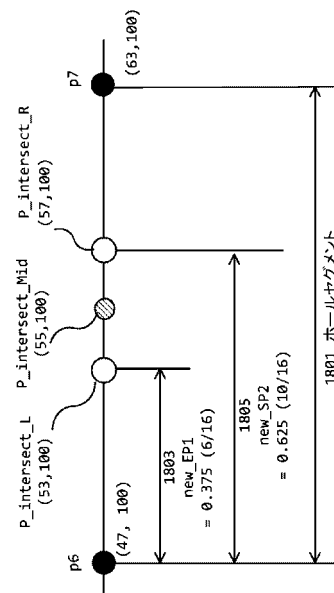


Fig. 18A

【図 18 B】

```

01: class Hole_segment {
02:     int hole_fromIndex;
03:     float hole_endParameter;
04:     int hole_toIndex;
05:     float hole_startParameter;
06: }
// e.g., 6
// e.g., 0.375
// e.g., 7
// e.g., 0.625

```

Fig. 18 B

【図 19】

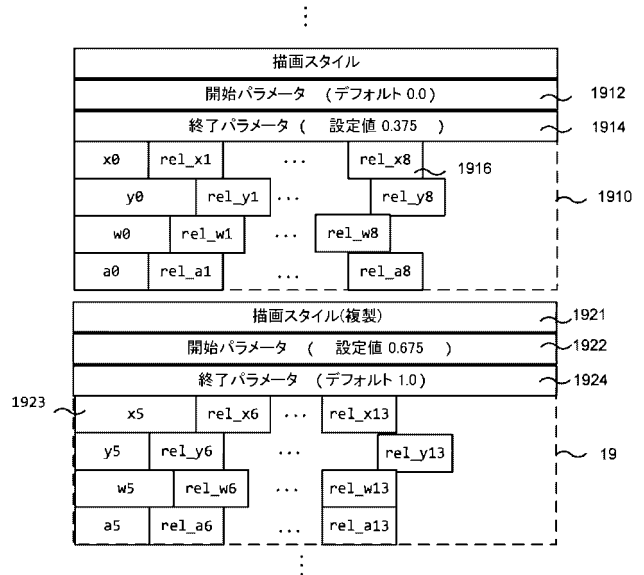


Fig. 19

【図 20】

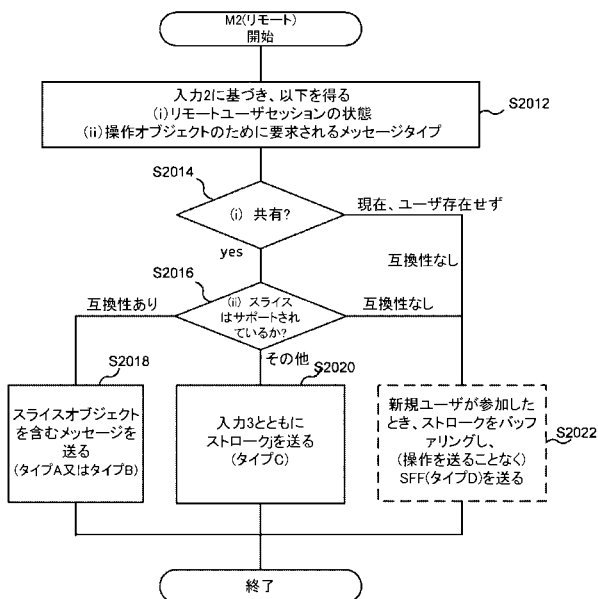


Fig. 20

【図 21 A】

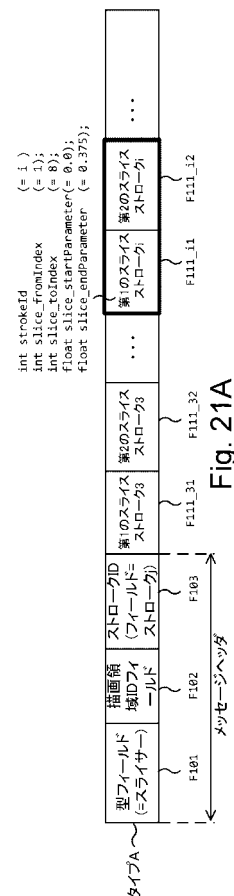
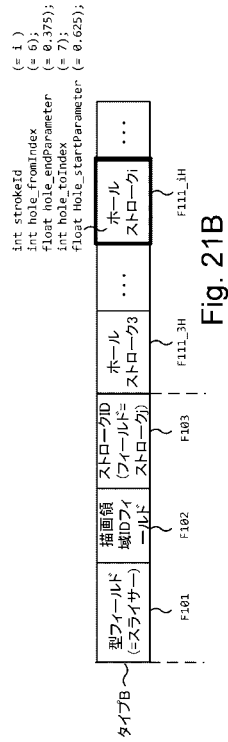
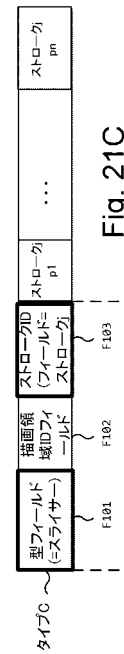


Fig. 21A

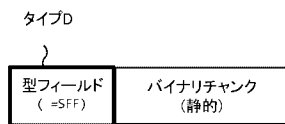
【 ㊦ 2 1 B 】



【图 2 1 C】



【図 2 1 D】



【 図 2 2 】

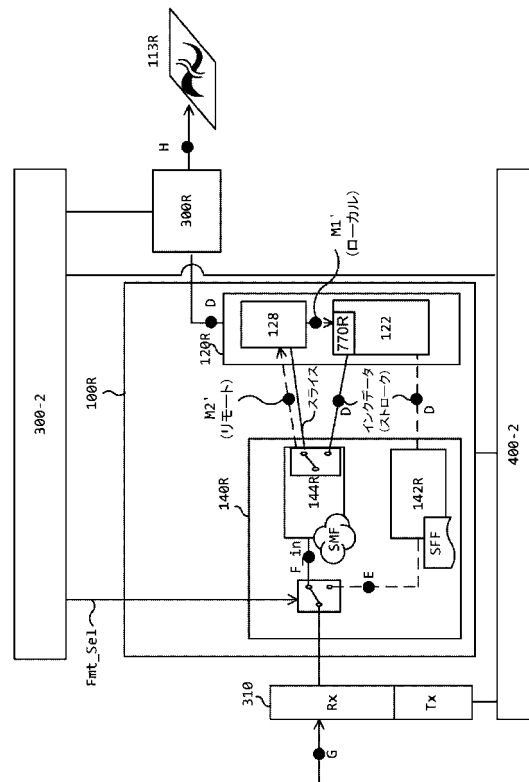


Fig. 22

【图 2 3】

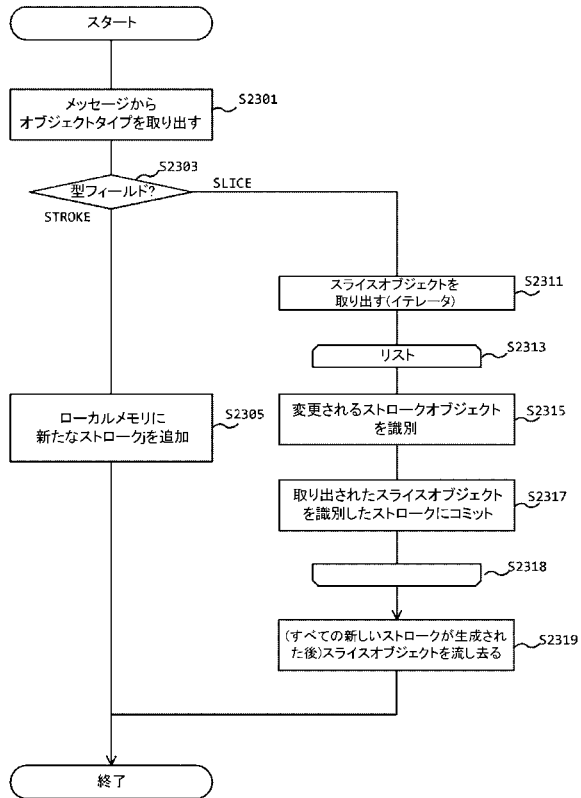


Fig. 23

【图 2 4 A】

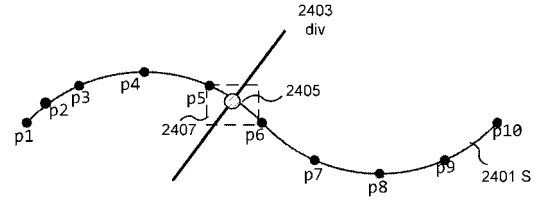


Fig. 24A
(関連技術)

【图 2 4 B】

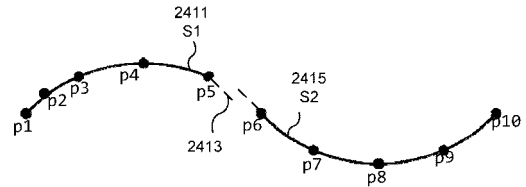


Fig. 24B
(関連技術)

【图 2 5】

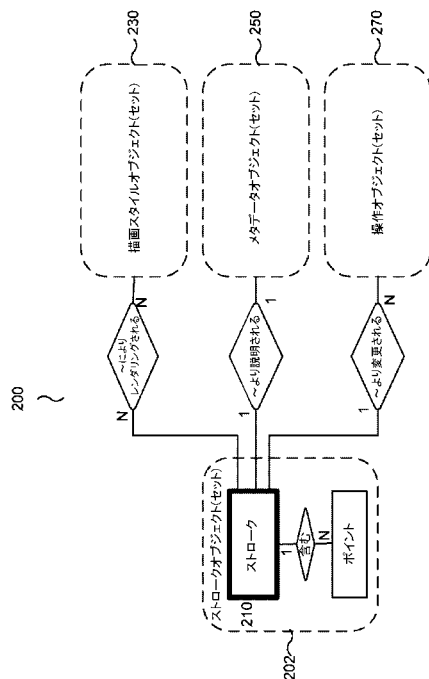


Fig. 25

【图 2 6】

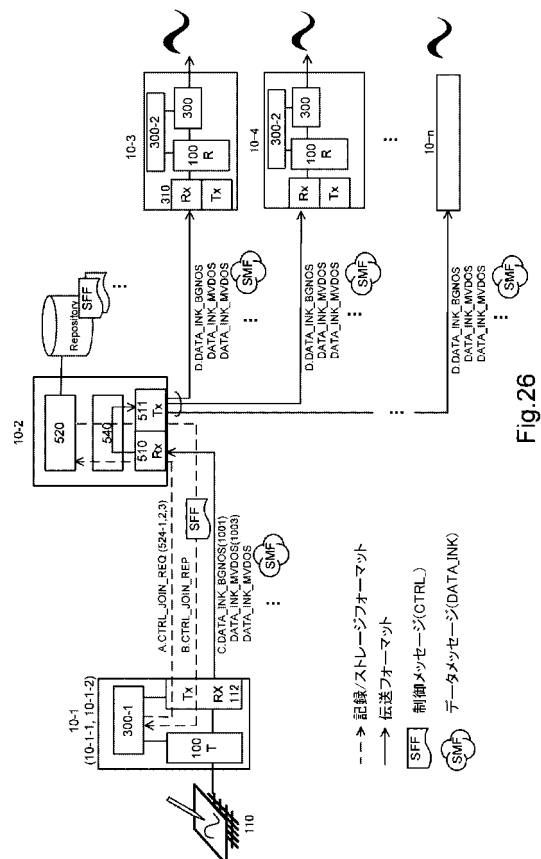


Fig. 26

【図 27】

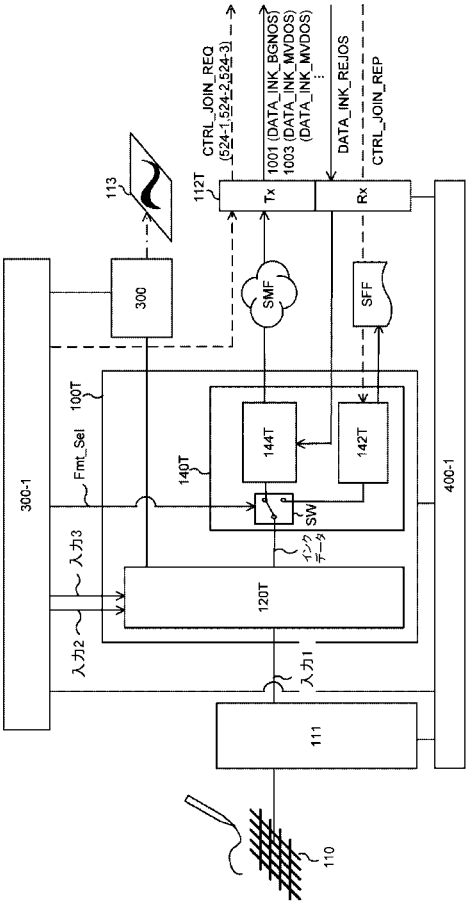


Fig. 27

【図 28】

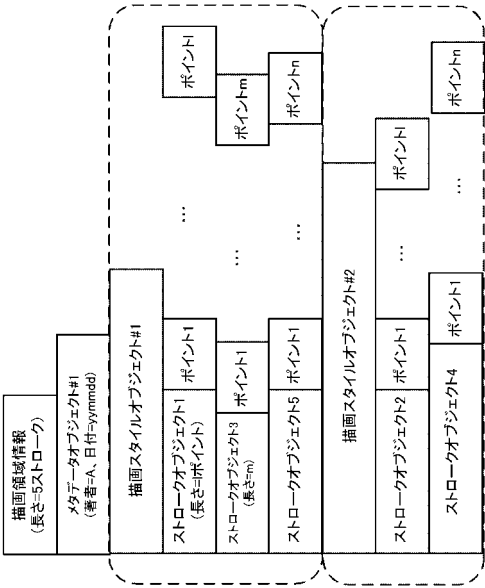


Fig. 28

【図 29】

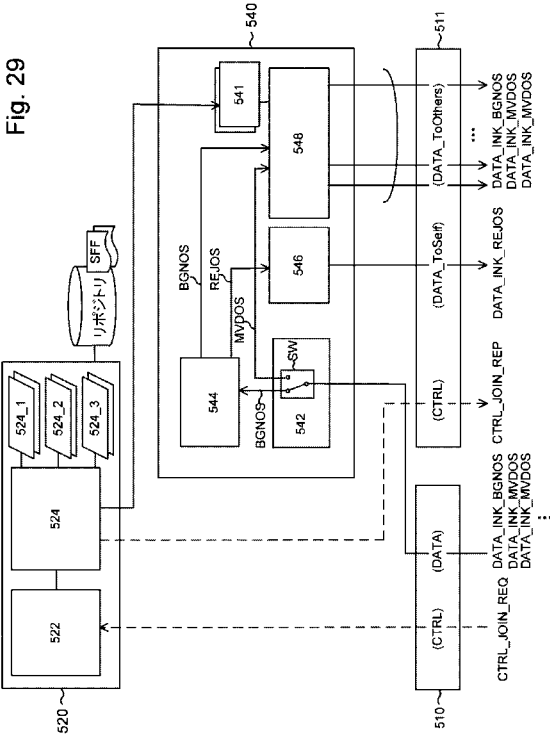


Fig. 29

【図 30 A】

タイプ	(長さ)	値	
		Tx	Rx
(524_1a)/パケット再送信	2	TRUE	FALSE
(524_1b) MTU, MSS	16	1460	1460
(524_1c)ストロークデータ断片化可否	2	TRUE	FALSE
(524_1d)最大Rx遅延	8	100 [msec]	
(524_1e)メッセージ暗号化	2	TRUE	FALSE
(524_1f)メッセージ再送信	2	ENABLE	DISABLED
(524_1g)音声同期可否	2	ENABLE	ENABLE
(524_1z)他パラメータ設定識別子	2	#15	

Fig. 30A

【図 3 0 B】

524_2		
タイプ	長さ	値
(524_2a) 描画領域ID	8	#123
(524_2b) ユーザーローカルキャンバス設定 (オフセット、回転、スケール)	24	(200, 30)
		0 [degree]
		x1
(524_2d) 描画ツールセットid	8	Default_pentool_set

Fig. 30B

【図 3 0 C】

524_3		
タイプ	長さ	値
(524_3a) ユーザーアカウント優先度	2	High
(524_3b) ブロックユーザーリスト	8n	DENY: User_10_m
		DENY: User_10_n
		ACCEPT: ALL

Fig. 30C

【図 3 1】

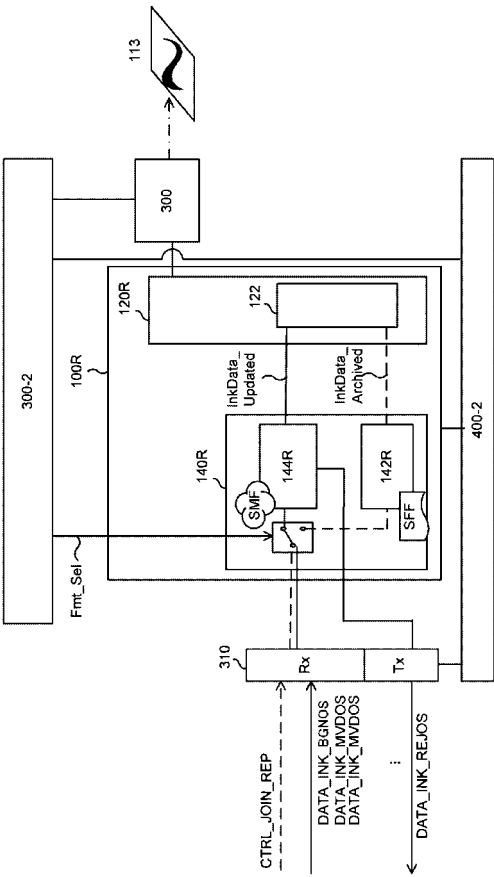


Fig. 31

【図 3 2】

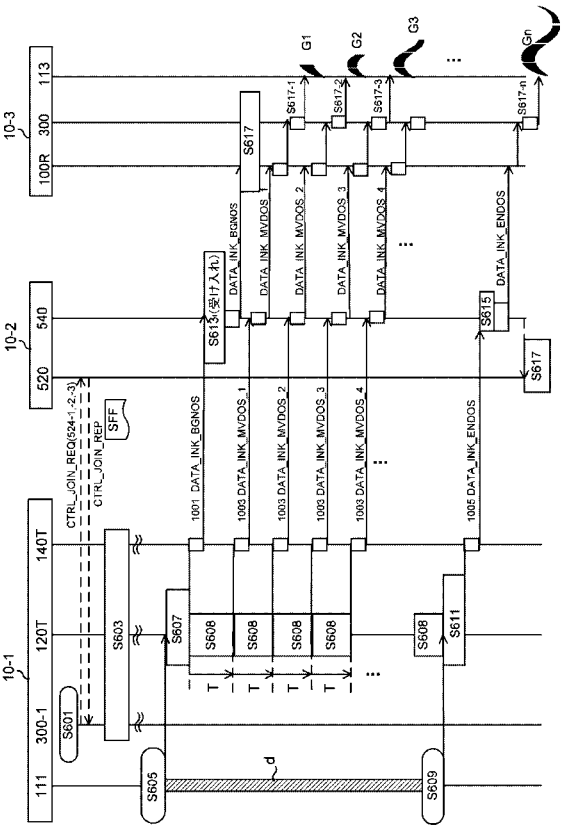


Fig. 32

【図 3 3】

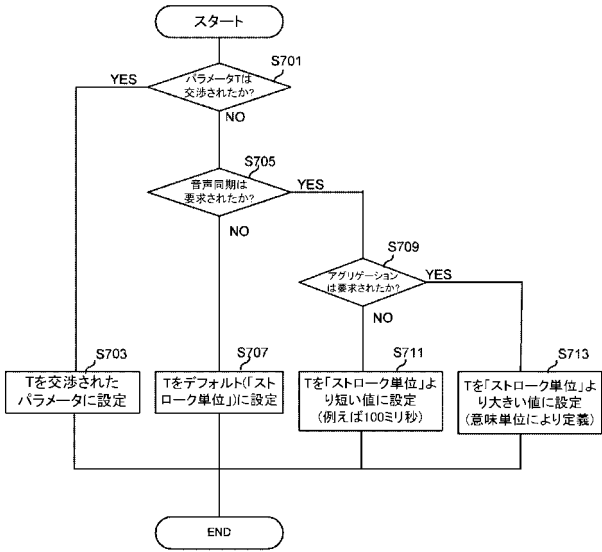


Fig. 33

【図 3 4】

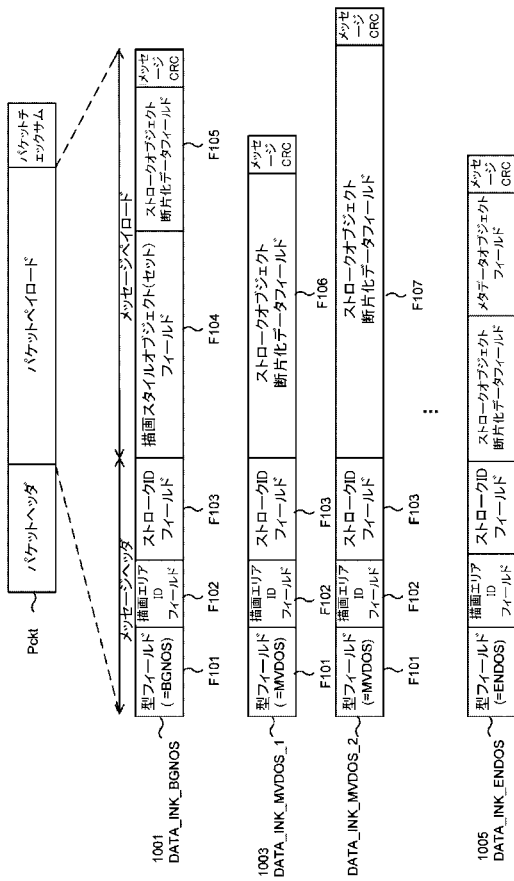


Fig.34

【図 3 5 A】

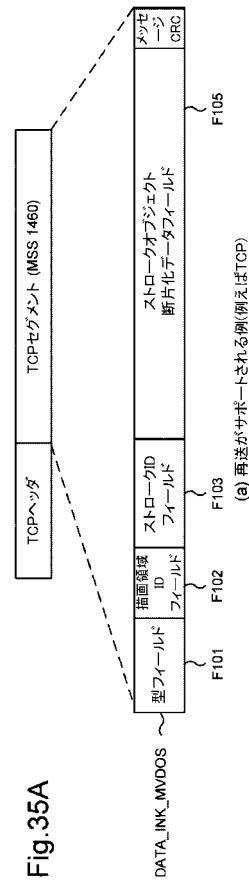


Fig.35A

【図 3 5 B】

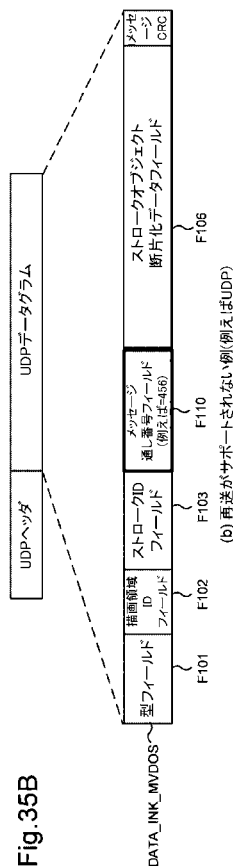


Fig.35B

【図 3 6 A】

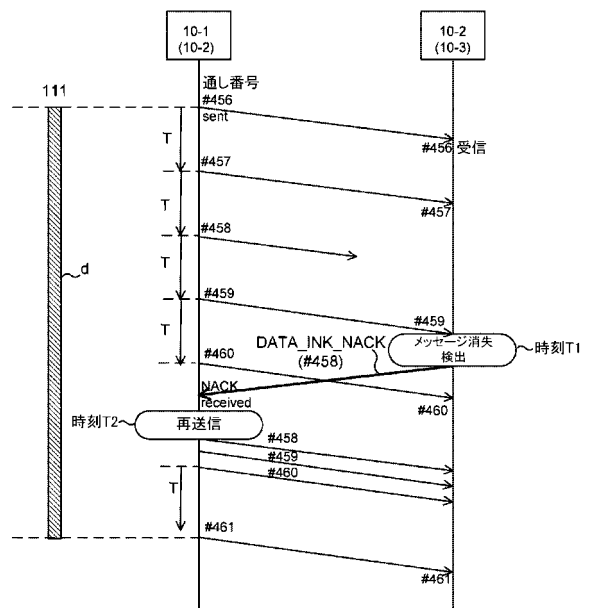


Fig.36A

【図 3 6 B】

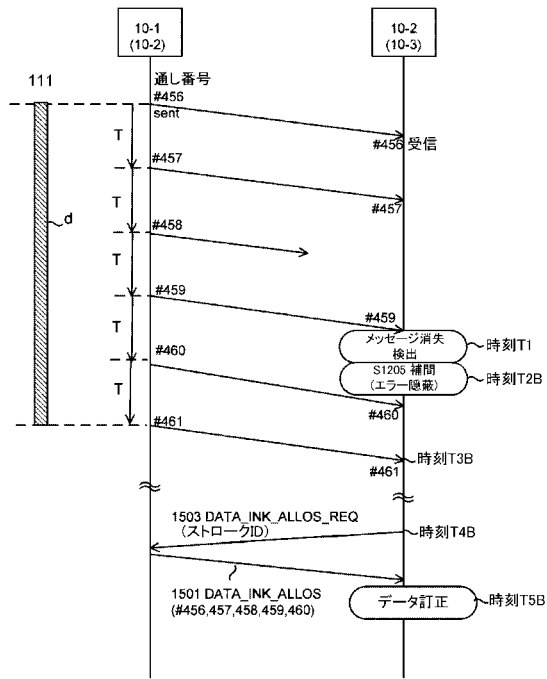


Fig.36B

【図 3 6 C】

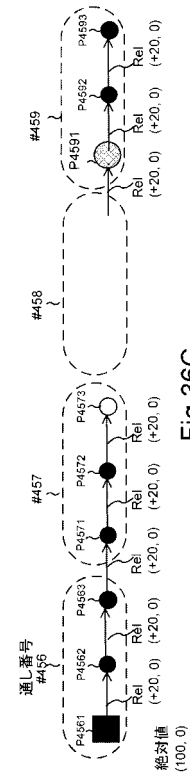


Fig.36C

【図 3 6 D】

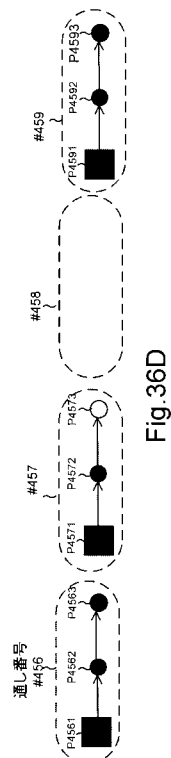


Fig.36D

【図 3 6 E】

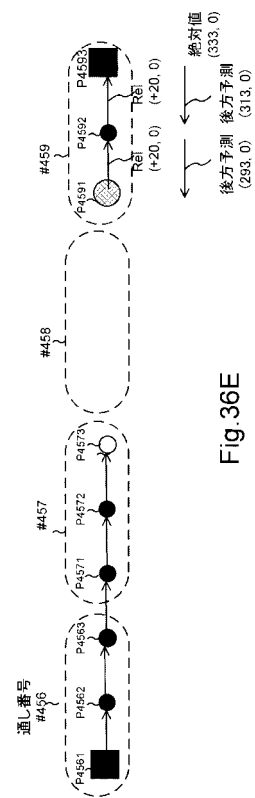
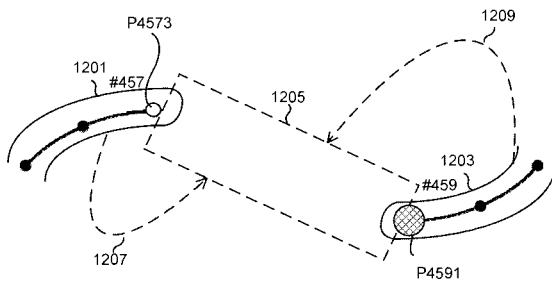
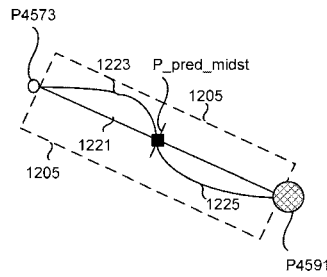


Fig.36E

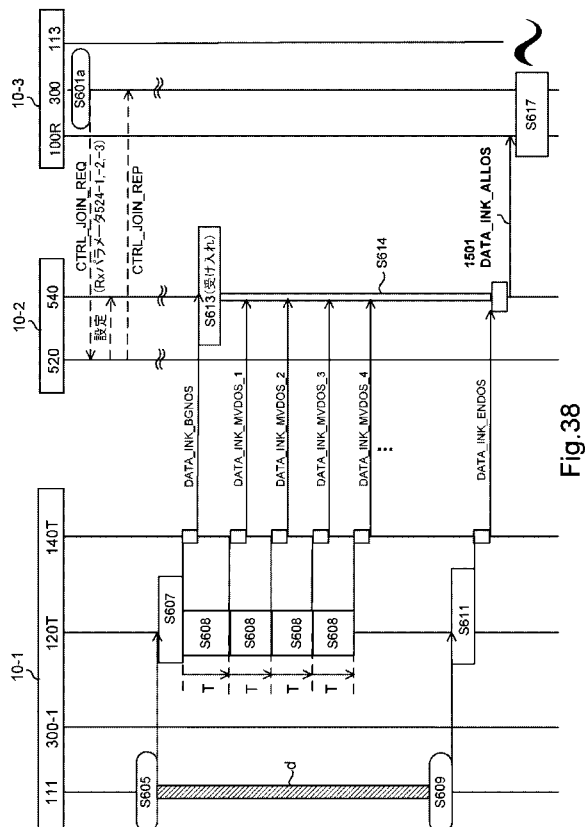
【 図 3 6 F 】



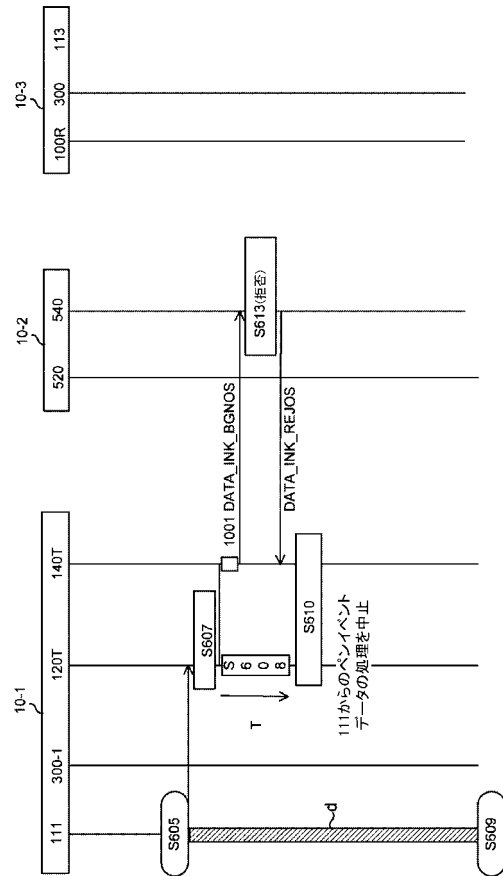
【 図 3 6 G 】



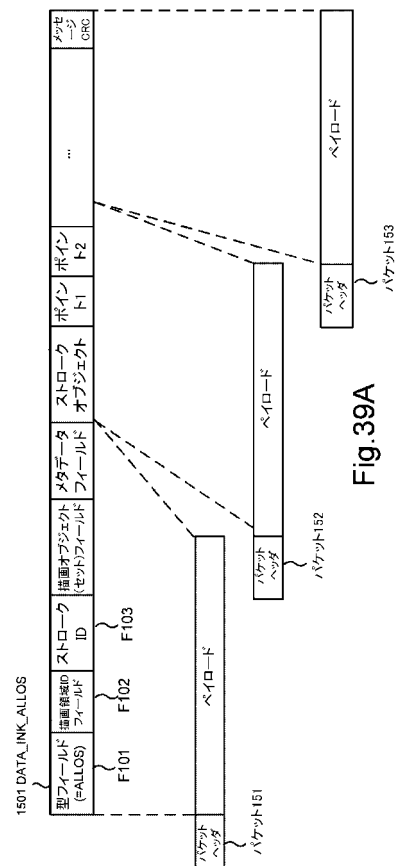
【图 3 8】



【図 3 7】



【図 3 9 A】



【図 39 B】



Fig.39B

【図 40】

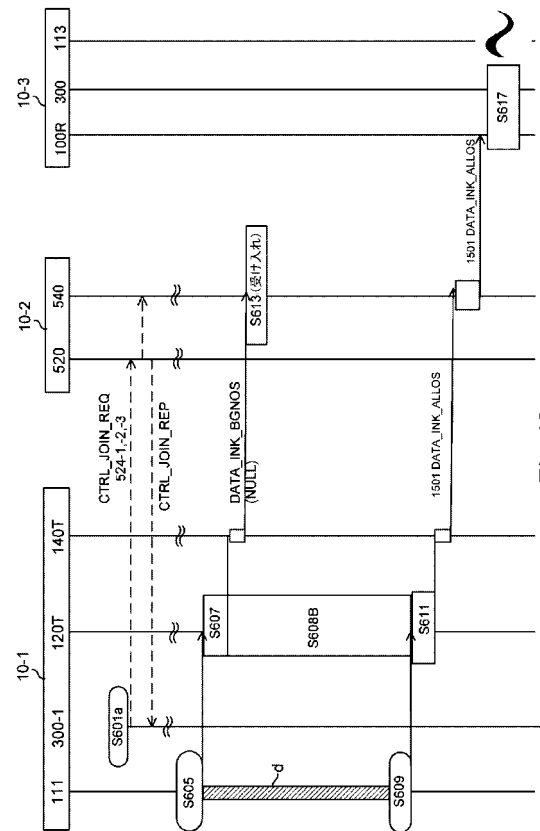


Fig.40

【図 41】

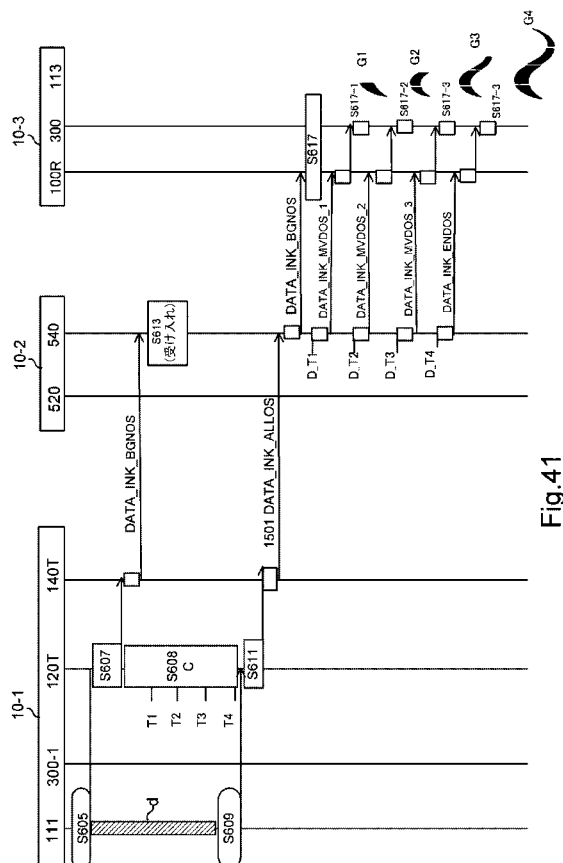


Fig.41

【図 42】

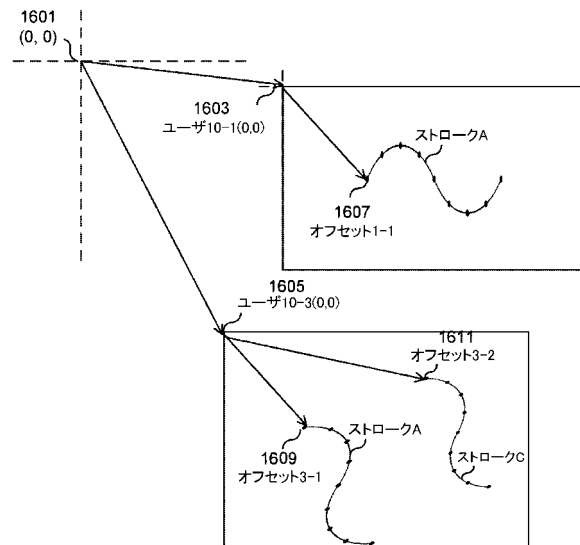


Fig.42

【图 4 3】

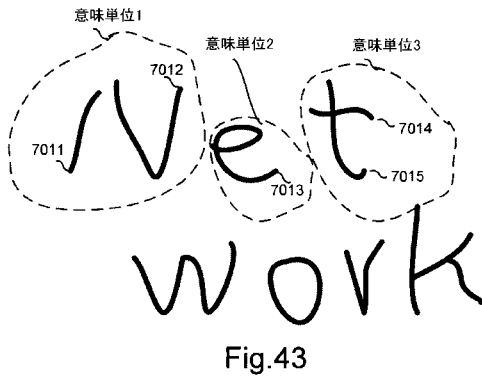


Fig.43

【 図 4 4 】

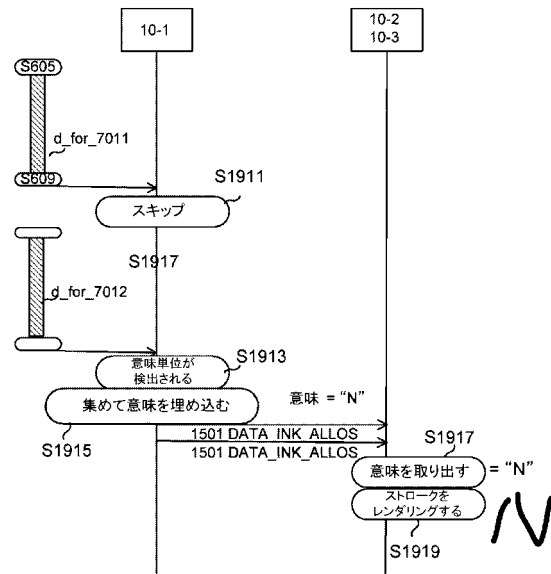


Fig.44

【图 4 5】

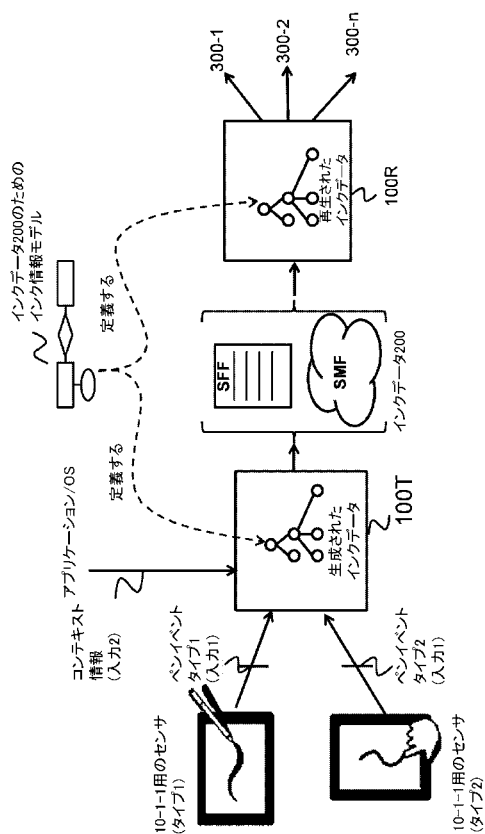


Fig. 45

【图 4 6 A】

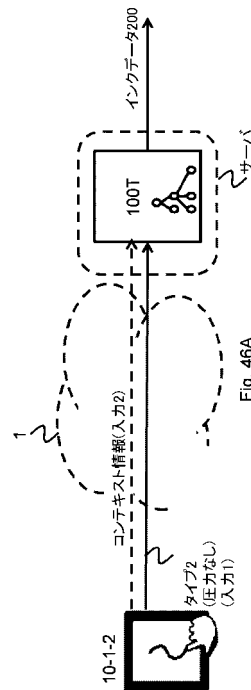
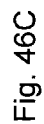
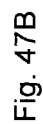
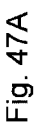


Fig. 46A

【図 4 6 C】



【图 4 7 B】



【図 48 A】

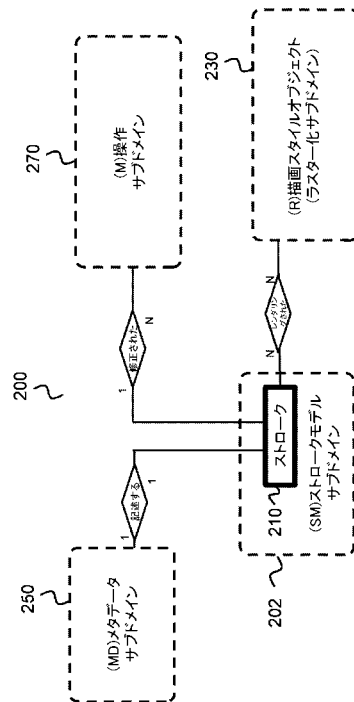


Fig. 48A

【図 48 B】

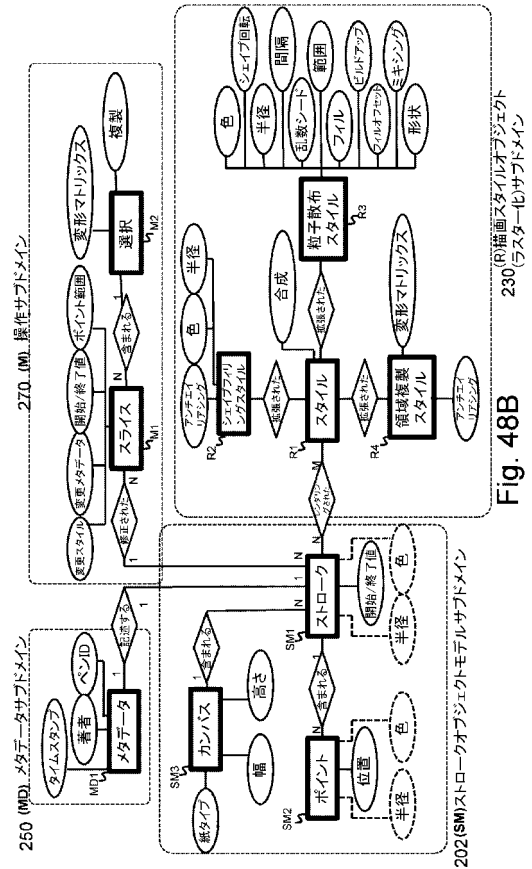


Fig. 48B

【図 48 C】

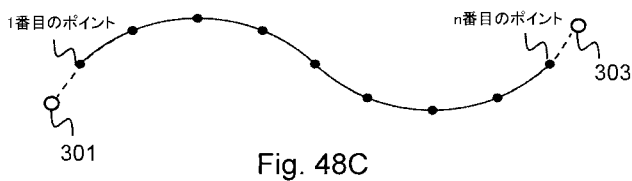


Fig. 48C

【図 48 E】

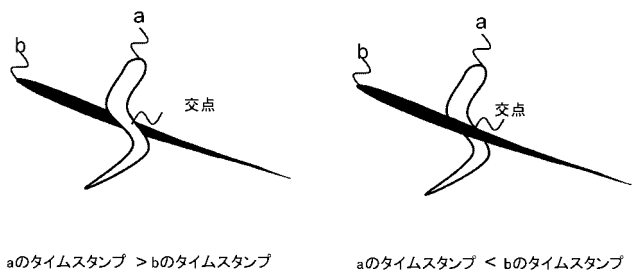


Fig. 48E

【図 48 D】

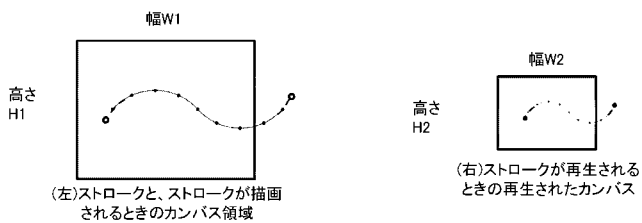


Fig. 48D

【 図 4 8 F 】

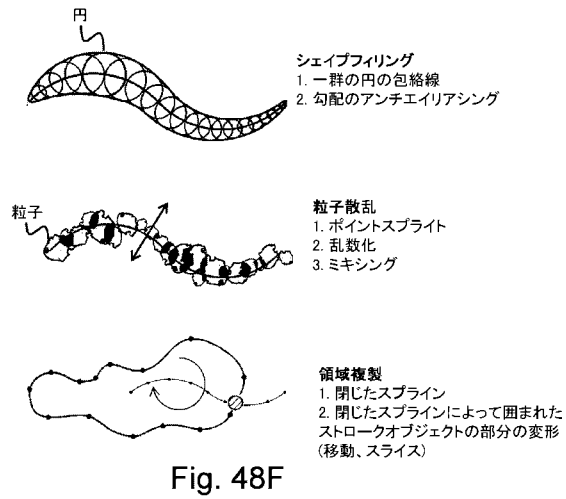


Fig. 48F

【图 4 8 G】

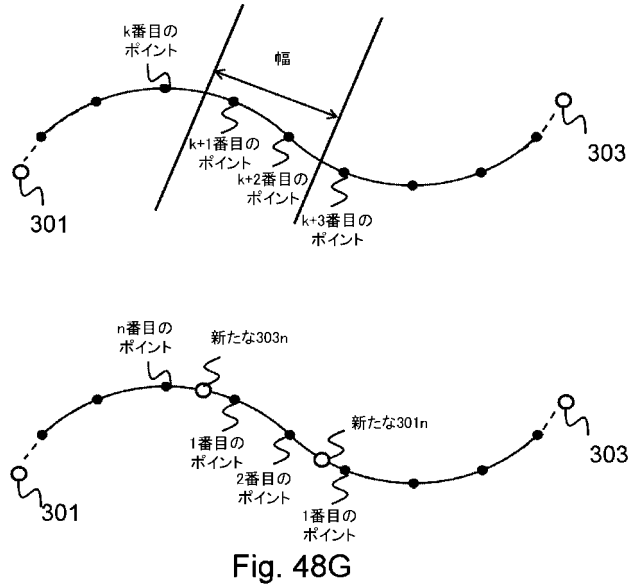


Fig. 48G

【図 4 8 H】

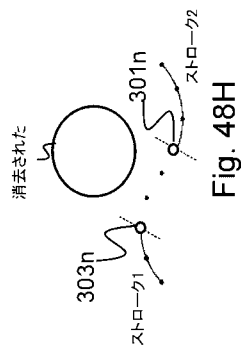


Fig. 48H

【図 4 8 I】

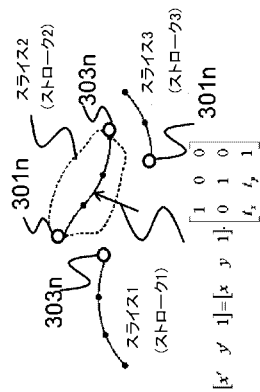


Fig. 481

【図 4 8 J】

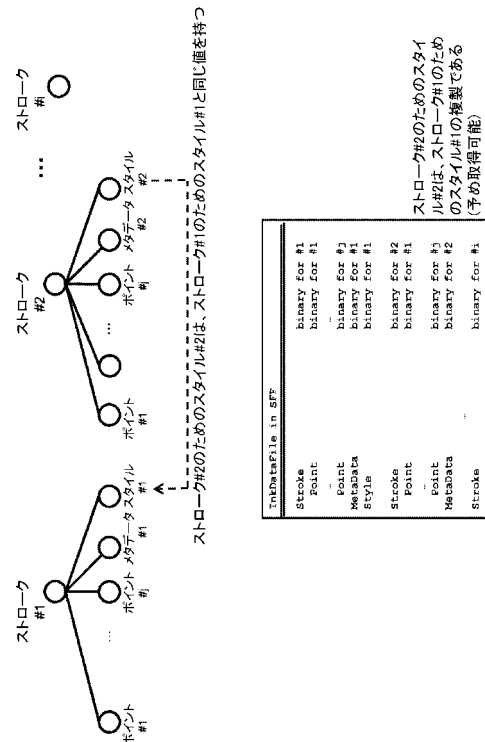
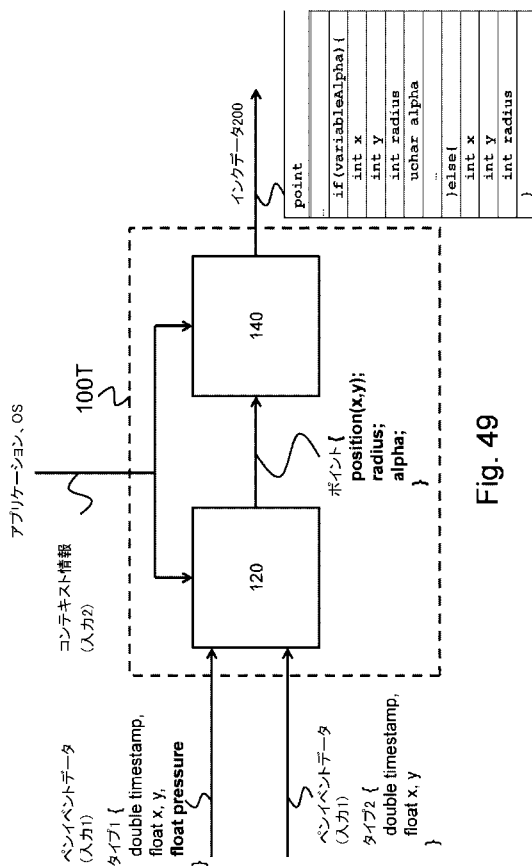
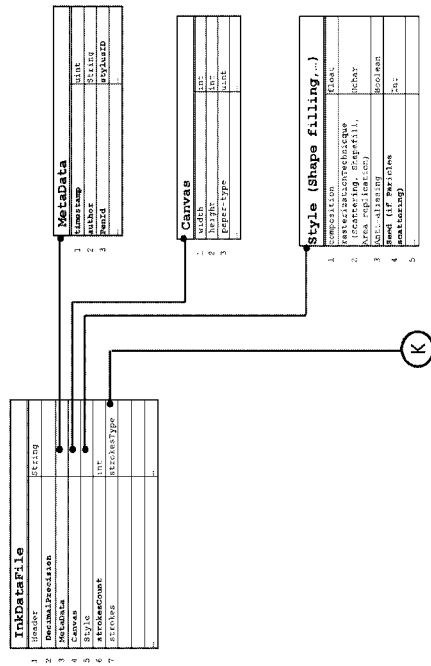


Fig. 48J

【図 4 9】



【図 4 8 K】



【図 50 B】

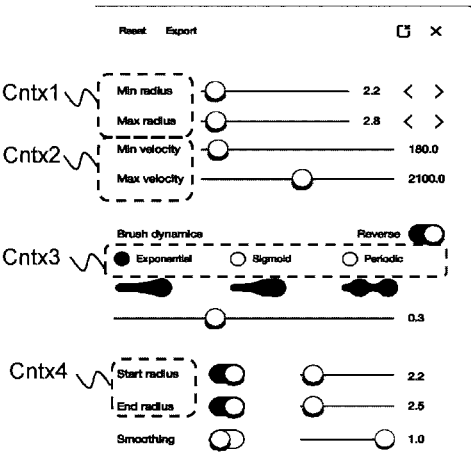


Fig. 50B

【図 5 1】

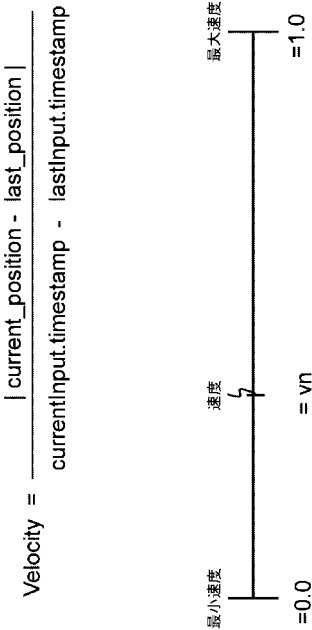


Fig. 51

【図 5 2】

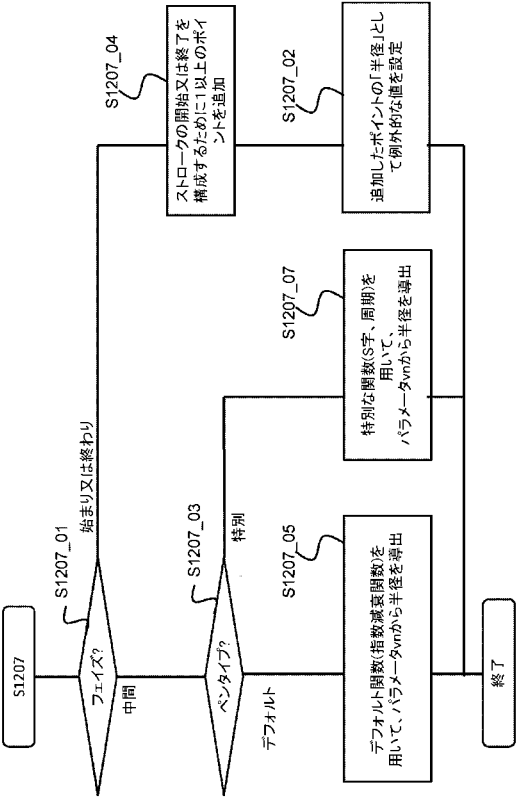


Fig. 52

【図 5 3】

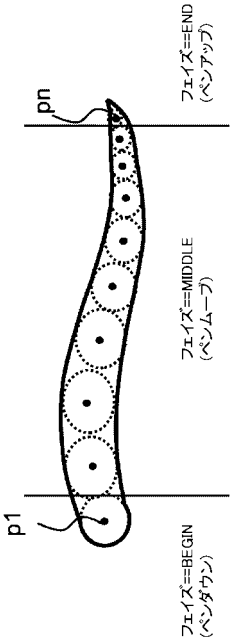


Fig. 53

【図 5 4】

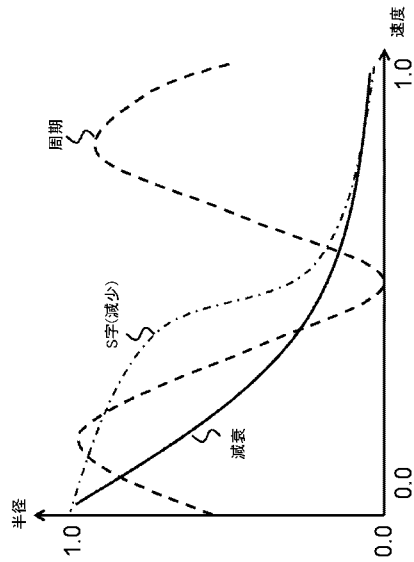


Fig. 54

【図 5 5】

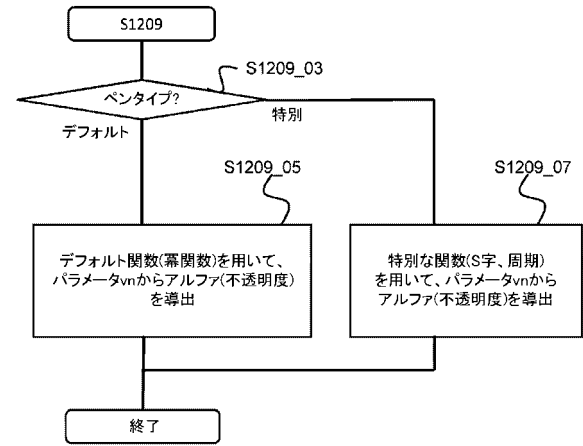


Fig. 55

【図 5 6】

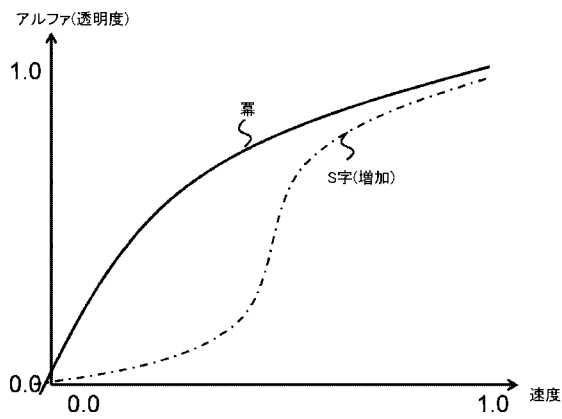


Fig. 56

【図 5 7】

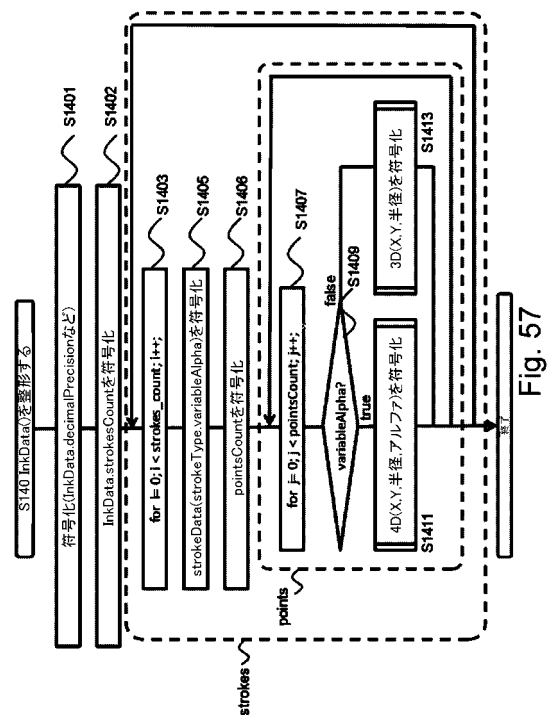


Fig. 57

S1411

```
01: void encode4D (inputPointType *input , Point4D *out, unsignedint decimalPrecision)
02: {
03:     out->x = (int)(input->x * decimalPrecision);
04:     out->y = (int)(input->y * decimalPrecision);
05:     out->radius = (int)(input->radius * decimalPrecision);
06:     out->alpha = (unsigned short)(input->alpha * TWO_TO_16_MINUS_1);
07: }
```

S1413

```
08: void encode3D (inputPointType *input, Point3D *out, unsignedint decimalPrecision)
09: {
10:     out->x = (int) (input->x * decimalPrecision);
11:     out->y = (int) (input->y * decimalPrecision);
12:     out->radius = (int) (input->radius * decimalPrecision);
13: }
```

Fig. 58

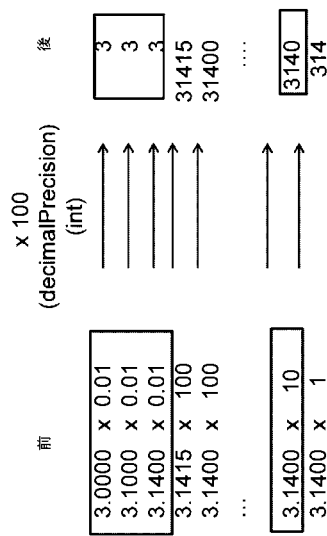


Fig. 60

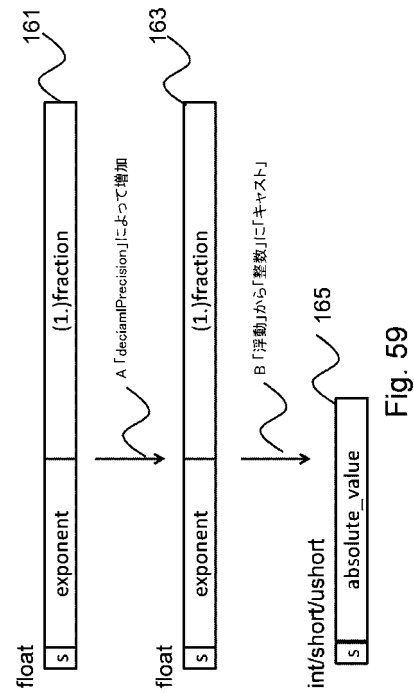


Fig. 59

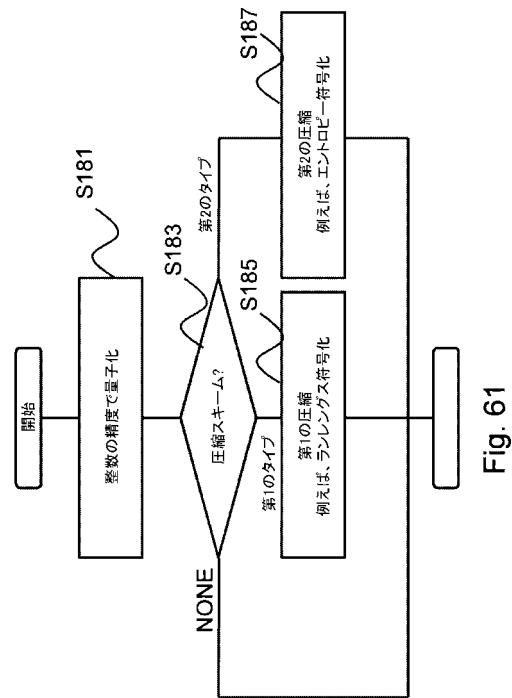


Fig. 61

【図 6 2】

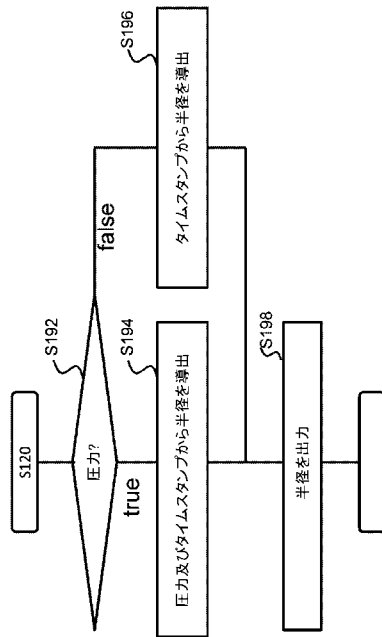


Fig. 62

【図 6 3】

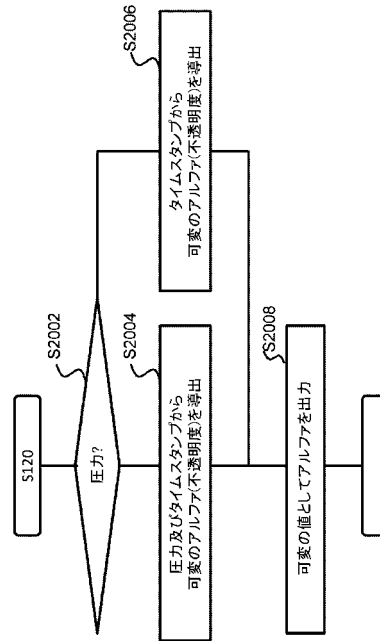


Fig. 63

【図 6 4】

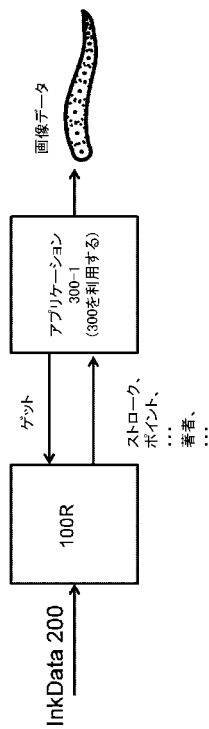


Fig. 64

【図 6 5】

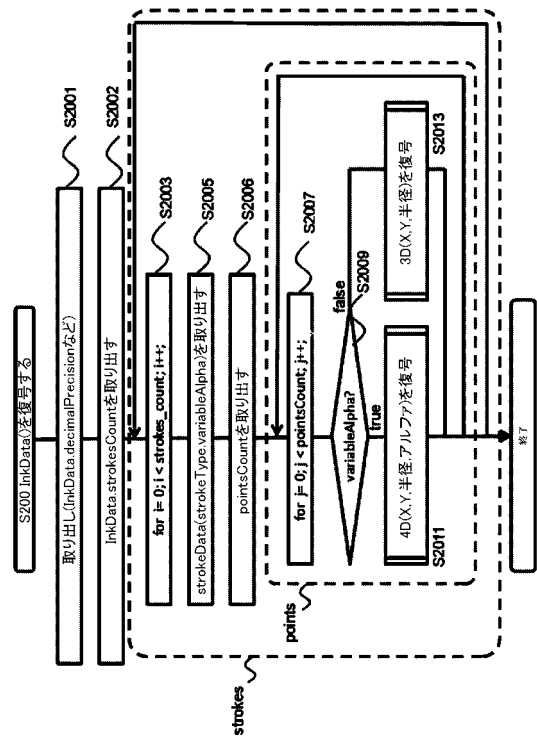


Fig. 65

【図 66】

```

S2011
01: point4D_Unpack()
02: {
03:   out->x = (float)input->x / decimalPrecision;
04:   out->y = (float)input->y / decimalPrecision;
05:   out->radius = (float)input->radius / decimalPrecision;
06:   out->alpha = (float)input->alpha / TWO_TO_16_MINUS_1;
07: }
INV_A

S2013
08: point3D_Unpack ()
09: {
10:   out->x = (float)input->x / decimalPrecision;
11:   out->y = (float)input->y / decimalPrecision;
12:   out->radius = (float)input->r / decimalPrecision;
13:   out->alpha = 1.0;
14: }

```

Fig. 66

【図 67】

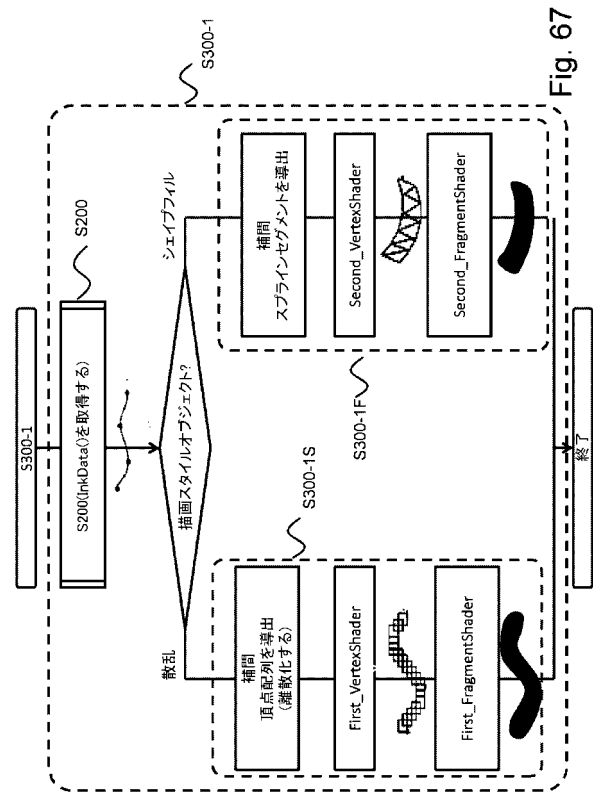


Fig. 67

【図 68】

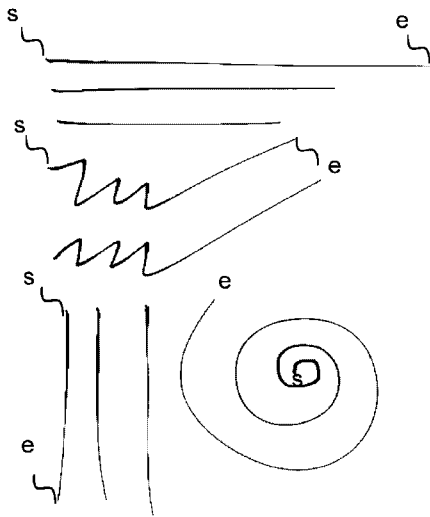


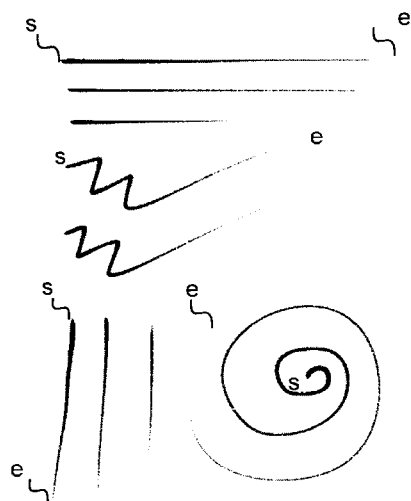
Fig.68

【図 69】

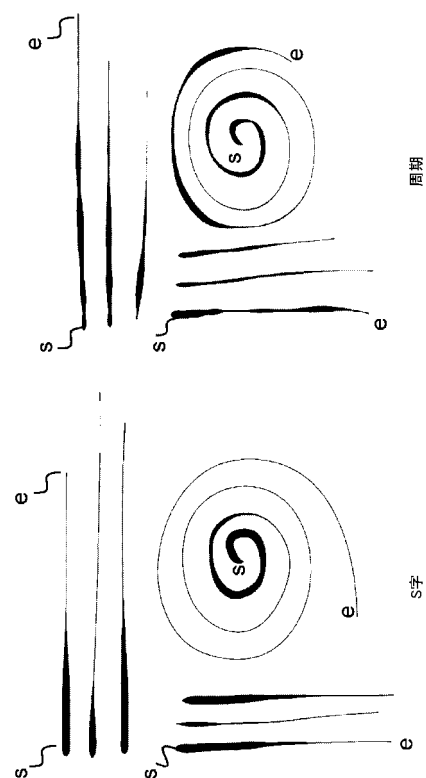


Fig.69

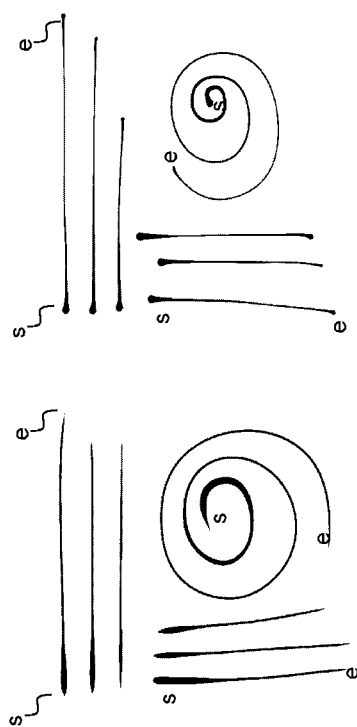
【图 7 0】



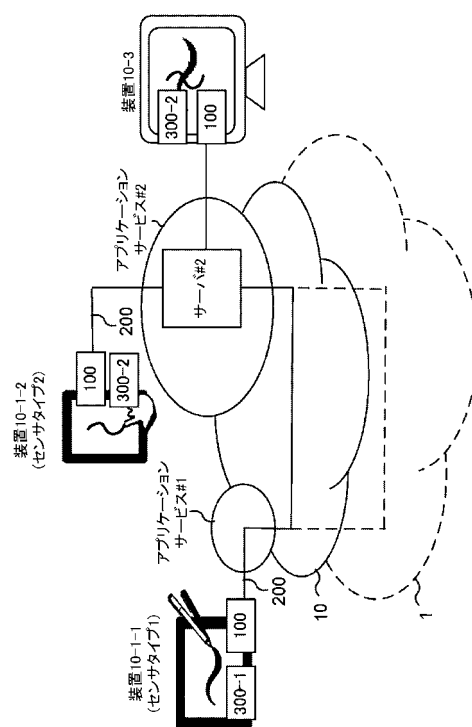
【図 7 1】



【图 7 2】



【図 7 3】



【図 7 4】

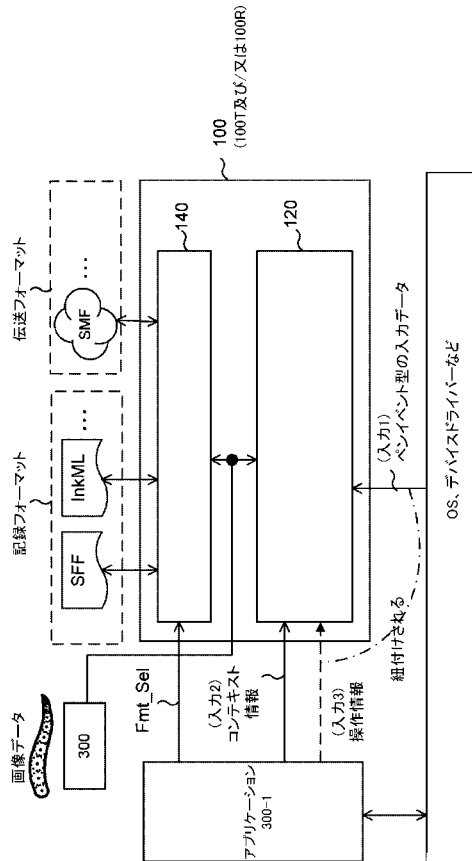


Fig. 74

【図 7 5】

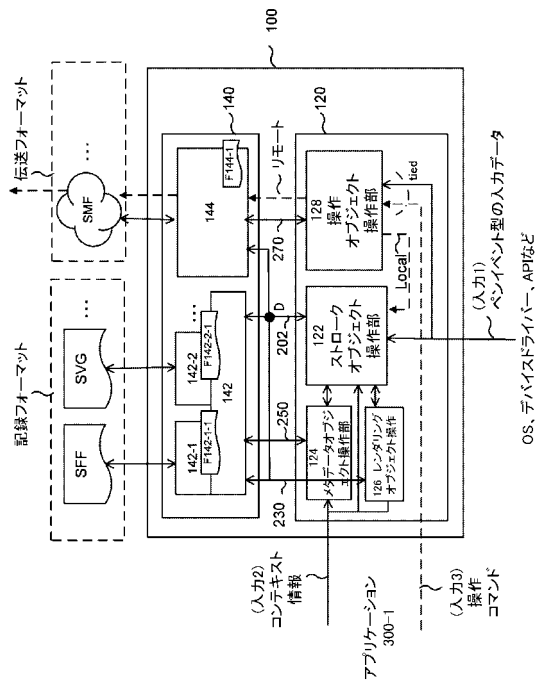


Fig. 75

【図 7 6】

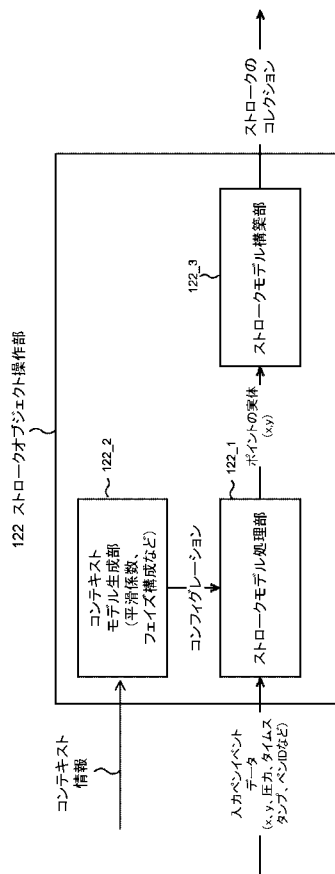


Fig. 76

【図 7 7 A】

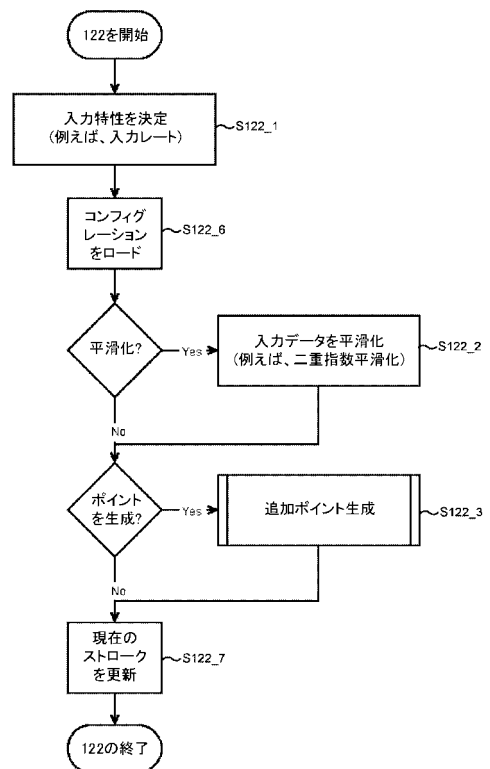


Fig. 77A

【図 77 B】

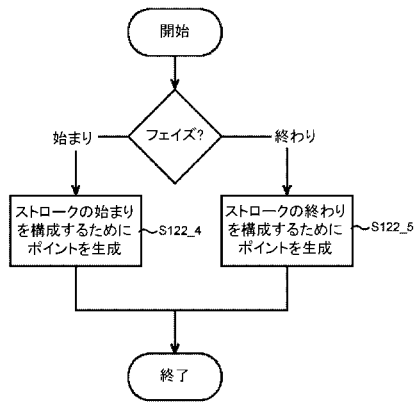


Fig. 77B

【図 78】

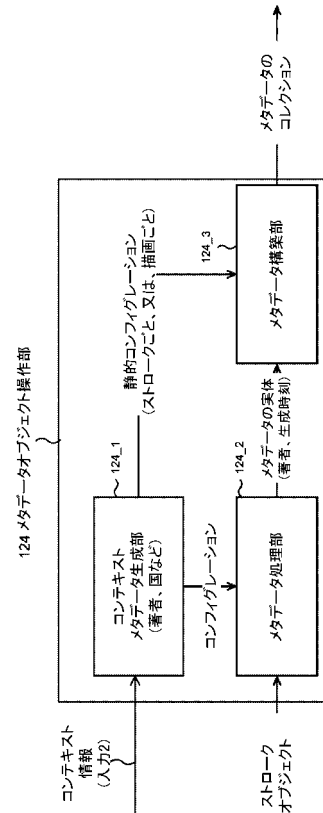


Fig. 78

【図 79】

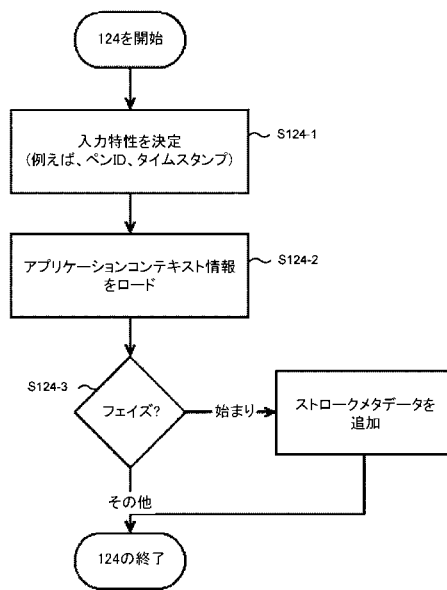


Fig. 79

【図 80】

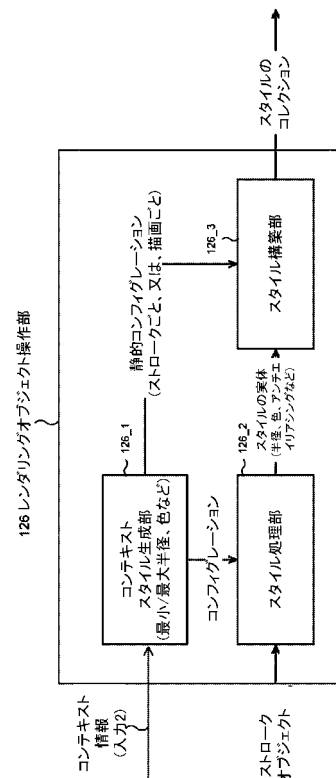


Fig. 80

【図 8 1】

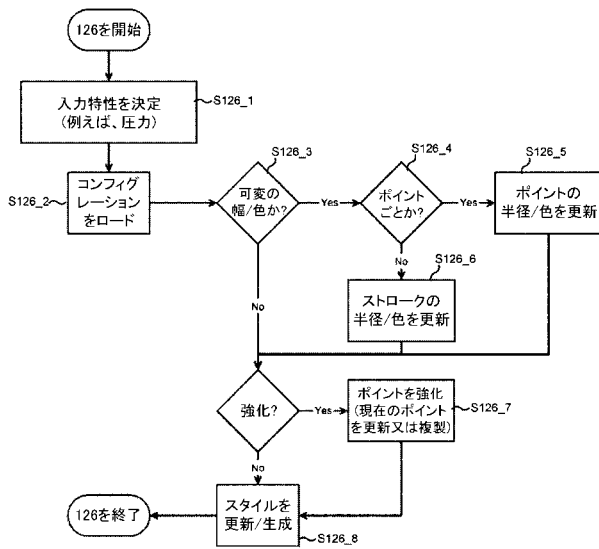


Fig. 81

【図 8 2】

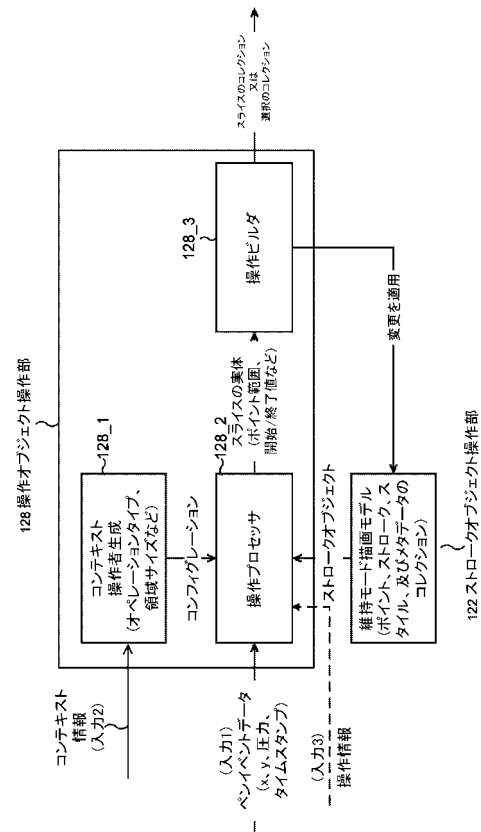


Fig. 82

【図 8 3 A】

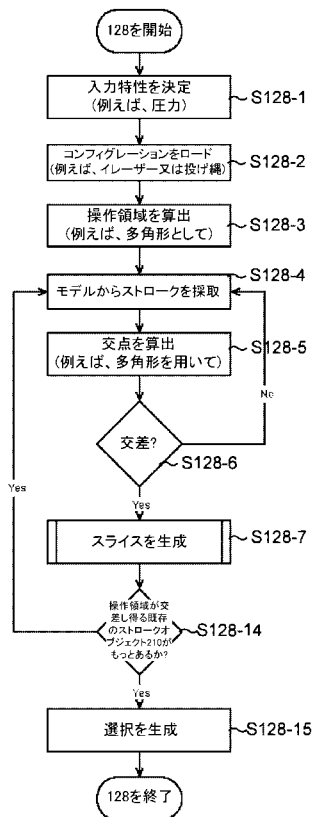


Fig. 83A

【図 8 3 B】

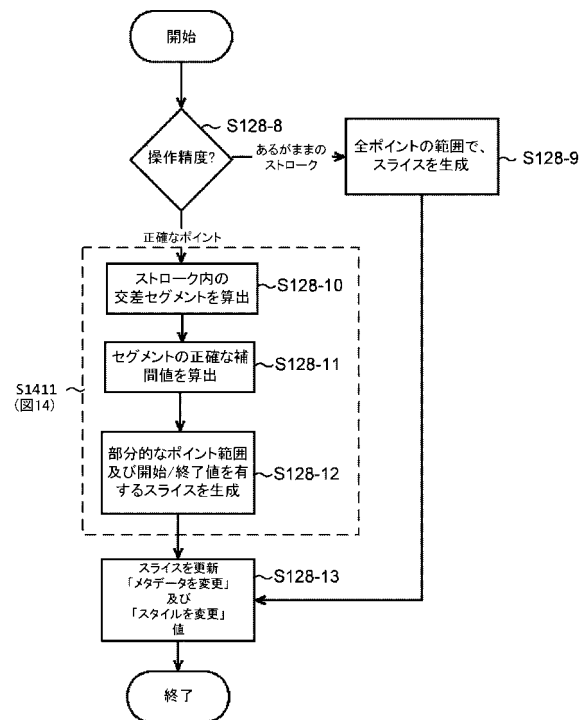


Fig. 83B

【図 84】

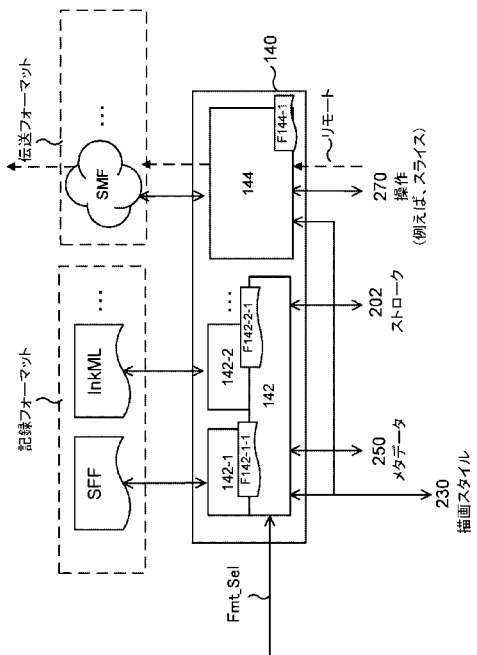


Fig. 84

【図 85】

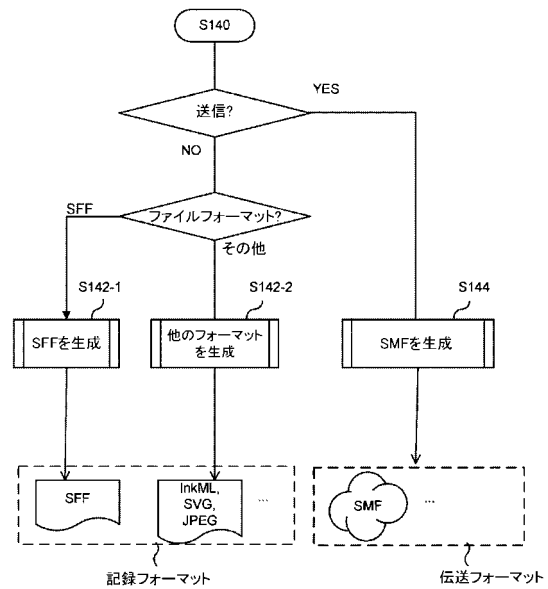


Fig. 85

【図 86】

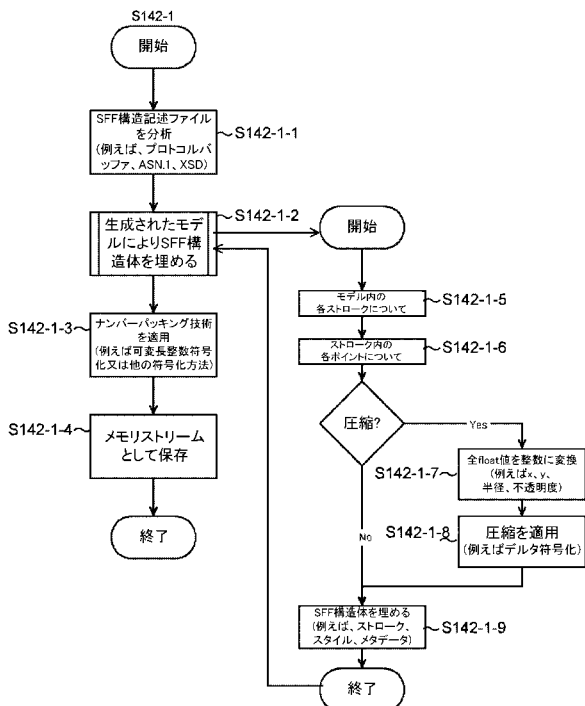


Fig. 86

【図 87】

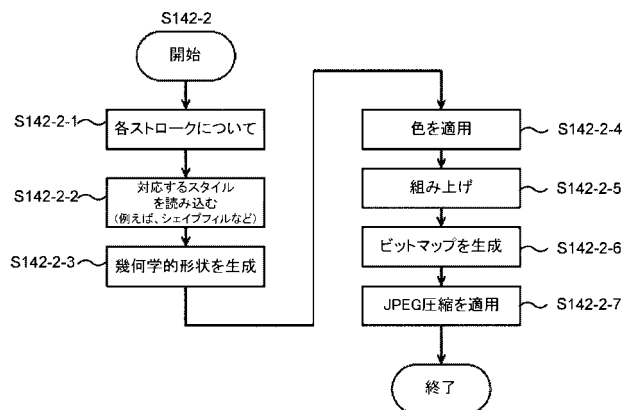


Fig. 87

【図 88】

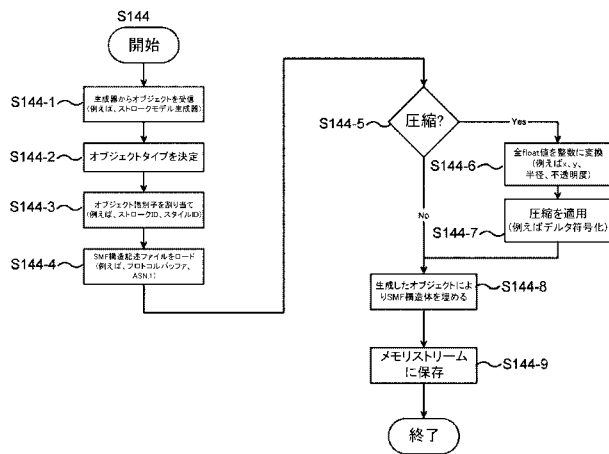


Fig. 88

【図 89】

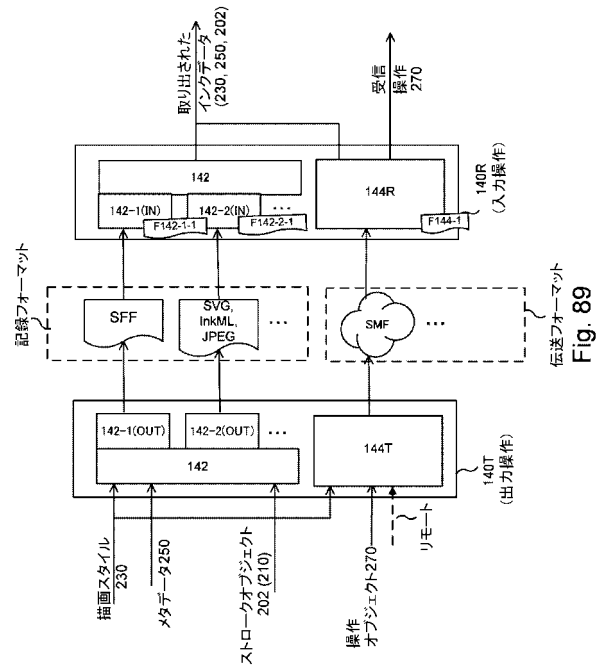


Fig. 89

【図 90 A】

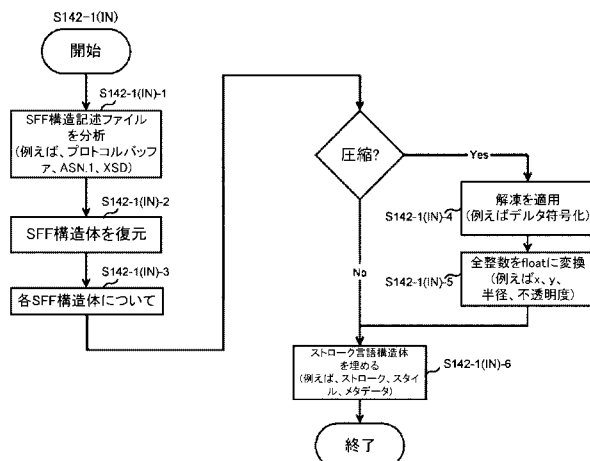


Fig. 90A

【図 90 B】

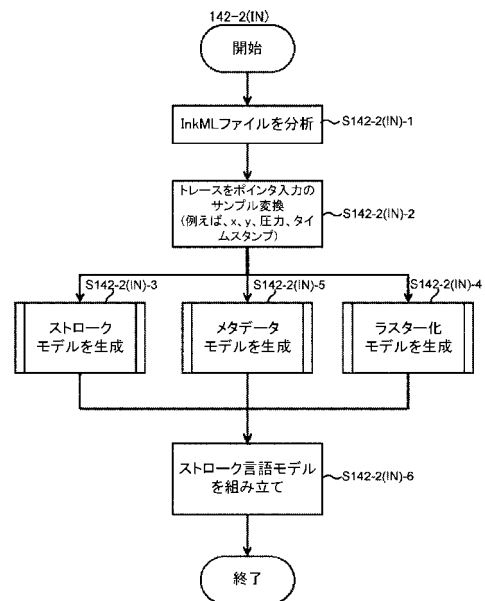


Fig. 90B

【図 90C】

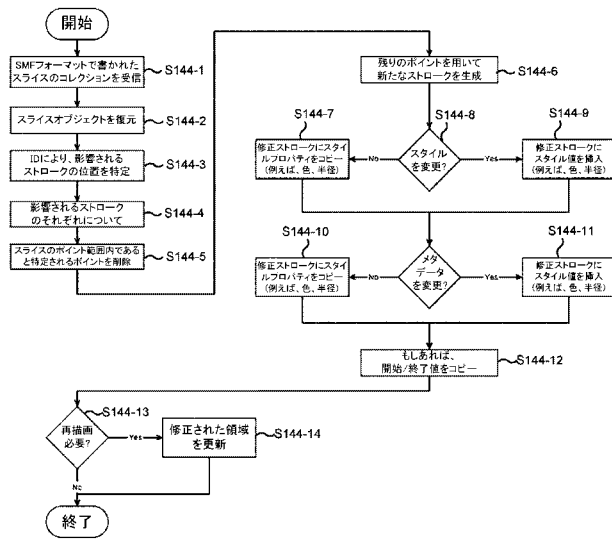


Fig. 90C

【図 91】

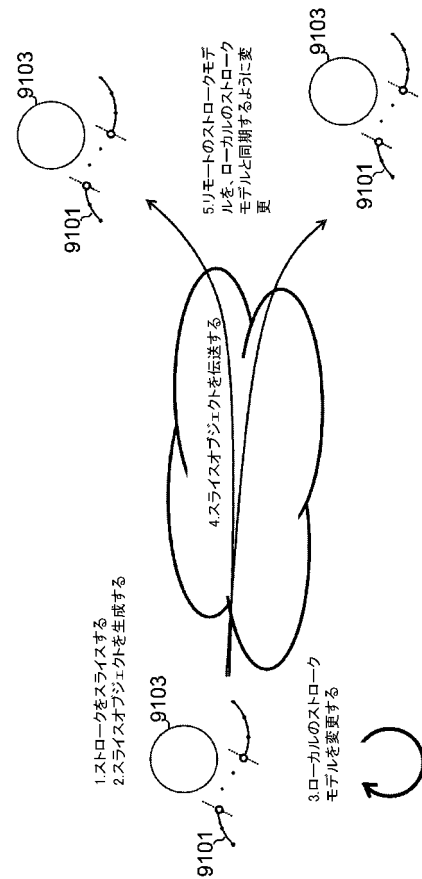


Fig. 91

【図 92】

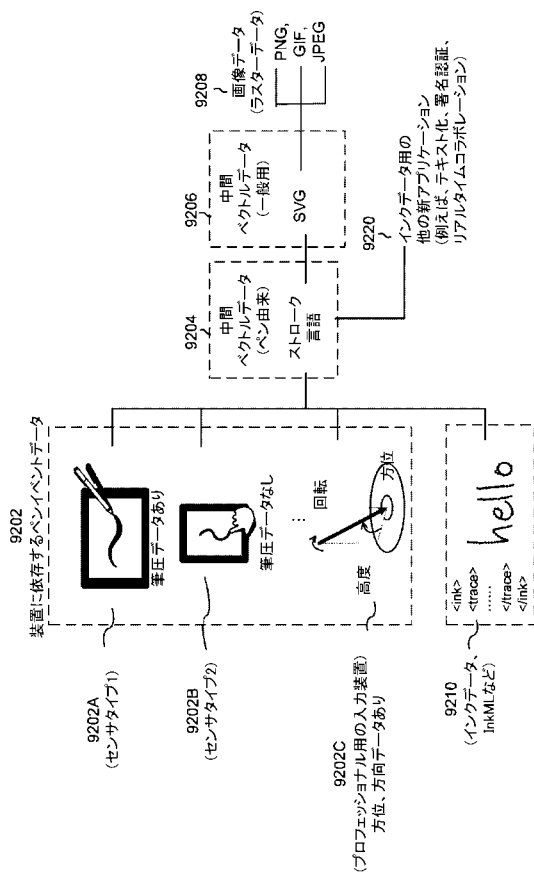


Fig. 92

【図 93】

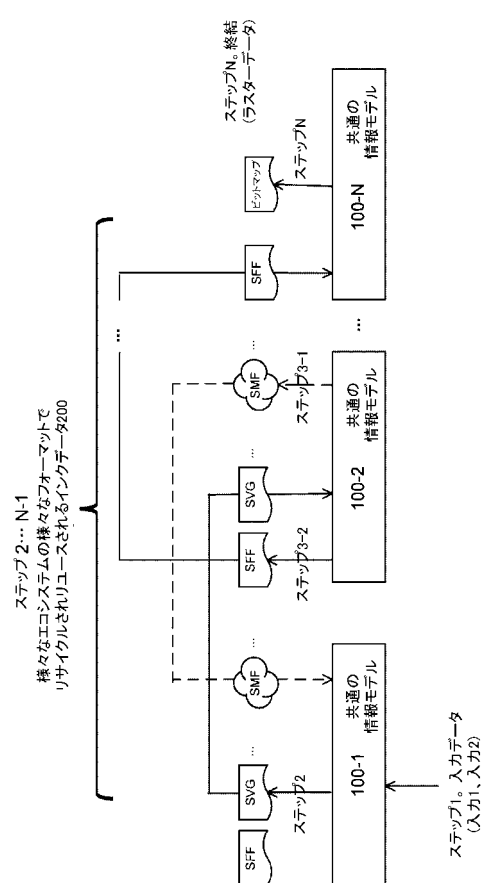


Fig. 93

フロントページの続き

(31)優先権主張番号 62/042,747

(32)優先日 平成26年8月27日(2014.8.27)

(33)優先権主張国 米国(US)

(72)発明者 アンゲロフ ブラニミール

ブルガリア共和国 1 3 0 3 ソフィア オソゴボ通り 3 8 ワコム ヨーロッパ ゲーエムベ
ーハー内

(72)発明者 ヨトフ ステファン

ブルガリア共和国 1 3 0 3 ソフィア オソゴボ通り 3 8 ワコム ヨーロッパ ゲーエムベ
ーハー内

(72)発明者 ワン ハイジ

ドイツ連邦共和国 4 0 2 1 9 デュッセルドルフ フェルクリンゲン通り 1 ワコム ヨーロ
ッパ ゲーエムベーハー内

(72)発明者 ムラディノファ ボリアナ

ブルガリア共和国 1 3 0 3 ソフィア オソゴボ通り 3 8 ワコム ヨーロッパ ゲーエムベ
ーハー内

Fターム(参考) 5E555 AA18 BA08 BB08 BC09 CA12 CB10 FA00