

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4299924号
(P4299924)

(45) 発行日 平成21年7月22日(2009.7.22)

(24) 登録日 平成21年4月24日(2009.4.24)

(51) Int.Cl.		F I		
GO6T 13/00	(2006.01)	GO6T 13/00		A
HO4N 5/262	(2006.01)	HO4N 5/262		

請求項の数 4 (全 16 頁)

(21) 出願番号	特願平11-215587	(73) 特許権者	593081408
(22) 出願日	平成11年7月29日(1999.7.29)		ソニー・ユナイテッド・キングダム・リミテッド
(65) 公開番号	特開2000-149058(P2000-149058A)		Sony United Kingdom Limited
(43) 公開日	平成12年5月30日(2000.5.30)		イギリス国 サリー, ウェーブブリッジ, ブルックランズ, ザ ハイツ (番地なし)
審査請求日	平成18年3月7日(2006.3.7)	(74) 代理人	100122884
(31) 優先権主張番号	9816762:0		弁理士 角田 芳末
(32) 優先日	平成10年7月31日(1998.7.31)	(74) 代理人	100113516
(33) 優先権主張国	英国 (GB)		弁理士 磯山 弘信
		(74) 代理人	100080883
			弁理士 松隈 秀盛

最終頁に続く

(54) 【発明の名称】 ビデオ特殊効果装置

(57) 【特許請求の範囲】

【請求項 1】

複数のビデオ特殊効果処理モジュールと、
表示スクリーンと、

上記ビデオ特殊効果処理モジュールを表すモジュール・スクリーン・アイコンを表示する、上記表示スクリーン上のパレット領域と、

上記表示スクリーン上のグラフ領域と、

上記スクリーン・アイコンを上記パレット領域から上記グラフ領域に移動させ、該スクリーン・アイコンを互いに図形的に連結させて、複合ビデオ特殊効果処理の表示を作成するためのユーザ操作可能手段と、

上記グラフ領域に表示された2以上の連結されたスクリーン・アイコンのグループを選択し、そのグループを上記パレット領域に転送するためのユーザ操作可能手段と、

上記複数のビデオ特殊効果処理モジュールのプログラムコード、及び、ユーザの操作データが記憶されるメモリとを具備、

上記グループは、上記パレット領域では、1つの複合処理スクリーン・アイコンとして表示され、

上記メモリ内の上記プログラムコードが、上記複数のビデオ特殊効果処理モジュール間に共通した動作を制御するコア構造部と、上記複数のビデオ特殊効果処理モジュールのそれぞれの効果に関する動作を制御するプラグイン部とで構成され、

上記メモリが上記ビデオ特殊効果処理モジュールによる効果の出力を記憶するキャッシュ

10

20

ユを有し、上記ビデオ特殊効果処理モジュールによる効果の出力を上記キャッシュに記憶する動作を上記コア構造部が制御するビデオ特殊効果装置。

【請求項 2】

上記パレット領域から上記グラフ領域に転送された複合処理スクリーン・アイコンは、上記グラフ領域において単一のアイコンとして表示される請求項 1 の装置。

【請求項 3】

上記パレット領域から上記グラフ領域に転送された複合処理スクリーン・アイコンは、上記グラフ領域においてアイコンのグループとして表示される請求項 1 の装置。

【請求項 4】

複合処理スクリーンの表示を上記グラフ領域におけるアイコンのグループと該グラフ領域における単一のアイコンとのどちらかに変えるためのユーザ操作可能制御手段を有する請求項 2 又は 3 の装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ビデオ特殊効果装置に関するものである。

【0002】

【従来の技術】

ビデオ特殊効果装置においては、ユーザは一般に、多数の使用可能なモジュールの中から一連の効果モジュールを選択することにより、ビデオシーケンスに適用しようとする複合特殊効果を設定することができる。例えば、ユーザが設定する一連（「有向非線返しグラフ (directed acyclic graph)」) の効果には、

(i) イメージローダ (画像の格納)

(ii) モーショントラッカ (動きの追跡)

(iii) 動きの追跡及びイメージローダに連動した照明効果

(iv) 動きの追跡及びイメージローダに連動した画像再編成

が含まれるであろう。

【0003】

これらの装置の中には、コンピュータ・スクリーン又はウィンドウ上のグラフィカル・ユーザ・インタフェースを用いて複合特殊効果が設定されるものがあり、上記スクリーンにより、個々の効果モジュールを表すスクリーン・アイコンをコンピュータ・マウスでスクリーン上の「パレット」領域からスクリーン上の「グラフ」領域へ「引き入れ」ている。グラフ領域では、アイコンを図形的に (即ち、コンピュータ・スクリーン上に引かれた図形線により) 互いに連結し、複合効果のフローチャート表示を作成することができる。

【0004】

【発明が解決しようとする課題】

本発明は、ユーザが、標準的な連結された効果を作成しそれらを 1 つのグループとして後日使用するために記憶させ検索できるように望んでいることを認め、ユーザがこれをなすような便利な方法を提供しようとするものである。

【0005】

【課題を解決するための手段】

本発明によるビデオ特殊効果装置は、
複数のビデオ特殊効果処理モジュールと、
表示スクリーンと、

上記ビデオ特殊効果処理モジュールを表すモジュール・スクリーン・アイコンを表示する、上記表示スクリーン上のパレット領域と、

上記表示スクリーン上のグラフ領域と、

上記スクリーン・アイコンを上記パレット領域から上記グラフ領域に移動させ、該スクリーン・アイコンを互いに図形的に連結させて、複合ビデオ特殊効果処理の表示を作成するためのユーザ操作可能手段と、上記グラフ領域に表示された 2 以上の連結されたスクリ

10

20

30

40

50

ーン・アイコンのグループを選択し、そのグループを上記パレット領域に転送するためのユーザ操作可能手段と、

上記複数のビデオ特殊効果処理モジュールのプログラムコード、及び、ユーザの操作データが記憶されるメモリとを具え、

上記グループは、上記パレット領域では1つの複合処理スクリーン・アイコンとして表示され、上記メモリ内の上記プログラムコードが、上記複数のビデオ特殊効果処理モジュール間に共通した動作を制御するコア構造部と、上記複数のビデオ特殊効果処理モジュールのそれぞれの効果に関する動作を制御するプラグイン部とで構成され、上記メモリが上記ビデオ特殊効果処理モジュールによる効果の出力を記憶するキャッシュを有し、上記ビデオ特殊効果処理モジュールによる効果の出力を上記キャッシュに記憶する動作を上記コア構造部が制御する。

10

【0006】

本発明によれば、幾つかの連結された効果モジュールを表すただ1つのアイコンとして上記グループをパレット領域に表示できるので、上述の課題を解決することができる。

【0007】

【発明の実施の形態】

以下、図面を参照して本発明を具体的に説明する。

図1は、本発明を適用できるデジタルビデオ特殊効果装置の概要図である。図1において、連続する画像を表すデジタルビデオ信号は、入力インタフェース(I/F)100を介して受信されディスクアレイ装置110に記憶される。ディスクアレイ装置110はまた、本効果装置により生成された処理済み画像をも記憶し、これらは出力インタフェース120を介し出力として供給される。

20

【0008】

中央処理ユニット(CPU)130は、ディスクアレイに記憶されたデータにアクセスし、ユーザの指令(コマンド)に従い種々のビデオ特殊効果を達成する。CPU130は、マウス及びキーボードのようなユーザ入力装置140から入力を受け、作業データ及びプログラムコードをメモリ150に記憶させ、出力データを生成し、表示駆動部170を介してスクリーン160に表示させる。

【0009】

本効果装置は、適当なソフトウェア、例えばマイクロソフト・ウィンドーズNT(登録商標)オペレーティングシステムにより動作する汎用コンピュータ(例えば、PC)として具現することができる。この例では、ディスクアレイは、ウルトラSCSIデータリンクを介してCPU130に接続される。

30

【0010】

図2は、図1の装置に使用する作動ソフトウェアの編成の概略を(極めて一般的な水準で)示すものである。

このソフトウェアは、2種類のプログラムコードより成り、1つは図2の左側に示すコア構造部であり、もう1つは図2の右側に示す種々のプラグイン部である。このソフトウェアが初めに格納されると、コア構造部は、常に存在して異なる特殊効果プロセス間で共通する本装置の動作の各部分を制御する。これに対し、プラグイン部は、個々の特殊効果(照明効果、動き追跡効果など)に関するもので必要な時にのみ格納される。

40

【0011】

この配置は、現在ユーザが要求する効果モジュールに関するプラグインのみをメモリに格納すればよいので、極めて効率のよいものである。こうすると、想定されるすべての特殊効果に対するプログラムコードを格納しなければならないシステムに比べて、メモリを少なくすることができる。これはまた、装置の初期設定をもつと迅速化し、システムを最初スタートさせる際にプログラムコードをすべてメモリに格納する必要をなくし、グラフィティ(後述参照)において最初にアイコンを選ぶときのコードの格納による遅延を少なくすることができる。更に、この配置は、グラフィティ及びプラグインを除くコア処理のみを含むので、供給又は組込むべき装置部品(特に、作動ソフトウェア)を減らすこと

50

ができる。このシステムはまた第三者又はユーザがプラグイン部及びコア構造部間の決められたインタフェース・プロトコルを守る限り、彼等自身のプラグインを生成することを可能にしている。よって、ユーザは、比較的小さなプラグイン・プログラムを書込むことにより、注文の効果を極めて簡単に生成することができよう。

【 0 0 1 2 】

コア構造部及びプラグイン部は、David Chappell著1996年Microsoft Press 発行の「Understanding Active X and OLE」という本に記載された、いわゆるOLE (Object Linking and Embedding) プロトコルを用いて互いに伝達を行う。

【 0 0 1 3 】

OLEシステムでは、ソフトウェア設計者は、コンピュータ・プログラムの種々異なるセクションをいわゆる「COM (Component Object Model) オブジェクト」として実現できる。各COMオブジェクトは、各々が幾つかの「メソッド」を含む1以上の「COMインタフェース」を支えている。メソッドは、特定の運動を実行する機能又は手順である。COMメソッドは、上記COMオブジェクトを使用するソフトウェアによって呼び出すことができる。このシステムでは、COMオブジェクトを使用するソフトウェアの他の部分は、所定のインタフェースを介してのみそうすることができるように制約されている。即ち、それらは、所定のCOMインタフェースを介さなければ、該オブジェクト内のプログラムコード又はデータに直接アクセスすることができない。

10

【 0 0 1 4 】

そこで、本システムでも、コア構造部はプラグイン部とこれらのCOMインタフェースを介して伝達を行う(実のところ、コア構造部は、COMインタフェースを提供できる幾つかの相互連絡オブジェクトを有するが、基本原理は同じである。)。

20

【 0 0 1 5 】

図3は、図2に示したものよりずっと詳細に作動ソフトウェアの編成を示すものである。図3は、4つの象限に分割されている。左上の象限はいわゆるビューワーウィンドー310を示し、右上の象限は効果ユーザインタフェース(UI)320を示し、右下の象限は、関連するパラメータデータ(PD)を有する「効果サーバ」330を示し、左下の象限は、オブジェクトマネージャ350、描画マネージャ352及び変更マネージャ358と共にグラフを含むコア・プロセッサ340を示す。左下と右下の象限の間のインタフェースには、ウィンドーズNTレジストリの一部をなすメタ・データベース354と、効果アイコンを含むパレット356と、省略パラメータ値とがある。メタ・データベース354については図11を参照してあとで説明する。パレット356については図6~9を参照して後述する。

30

【 0 0 1 6 】

コア・プロセッサ340内には、個々の特殊効果を続けて連結した「グラフ」、即ち「有向非繰返しグラフ」がある。各効果は、グラフ内でプロキシ(代理)効果(PE)として表示されている。各PEは、効果の出力が得られると直ちにこれを記憶するキャッシュ(C)を有するので、例えば一連の効果の中で重要な効果に関連するパラメータが変化した場合、一連の効果の中の効果からデータを再使用することができる。各プロキシ効果は、夫々の効果サーバ330と対応している。

40

【 0 0 1 7 】

オブジェクトマネージャ350は、システム内で用いるオブジェクトの存続期間やメモリ処理を制御する責任をもつ。描画マネージャ352は、描画タスクの開始、進行、優先順位の指定及び停止を制御する。変更マネージャ358は、情報の取消し/やり直し及びシステム各部への変更の連絡を制御する。

【 0 0 1 8 】

図3は、基本的に、左側の2つの象限(左上及び左下)がどの効果にも関係しないというコア構造部の機能的特徴を示すように分割されている。これらのオブジェクトは、ユーザがどの特殊効果を実施しようとして望んでいるかに関係なく、メモリに格納される。右側の2つの象限(右上及び右下)は、プラグイン部に関係する。各プラグインは、そのプラグイ

50

ンによって実施される効果に関連した処理を行うサーバ330と、その効果に関するユーザインタフェース制御（実際は、ビューワーウィンドー310における表示制御）を行うユーザインタフェース（UI）320とを有する。

【0019】

図3には同様の上下に分ける分割線があり、これにより、2つの上方の象限（左上及び右上）は特殊効果に関連するユーザインタフェース制御に関係し、2つの下方の象限（左下及び右下）はそれらの効果を実施するために行う処理や編成に関係していることを示す。

【0020】

ビューワーウィンドー310は、本装置によって実行するために作成された一連の効果のうち特定の効果の出力及びその制御パラメータを観察する機会をユーザに与えるものである。即ち、ユーザがビューワーウィンドーを開くと、或る効果の出力が生成されるか、或いは可能な場合キャッシュ・メモリから検索される。

10

【0021】

多数のCOMオブジェクトを使用するこのタイプのシステムでは、各オブジェクトがその作業又は結果のデータを別々のデータファイルに蓄えるべきだとするのは望ましくない、と一般に考えられている。実際、そのようにすると、OLEシステムを確立させた論理の多くと合わないようである。代わりに、このタイプのシステムのデータは、全部のオブジェクトによりただ1つのファイル又は「複合文書」に記憶される。ただし、順序付けされた構造であれば、複合文書内に記憶される。

【0022】

元来、複合文書内にはファイル及びディレクトリ構造に似た或る構造が設けられている。ディレクトリの均等物はいわゆる「ストレージ」であり、ファイルの類似物はいわゆる「ストリーム」である。各複合文書はルート（根）ストレージを有し、その下方にはストレージ及びストリームのよく知られた木構造がある。COMストリーム・インタフェースはCOMストレージ・インタフェースより簡単であるが、勿論ストレージ・インタフェースの方がもっと柔軟性がある。

20

【0023】

一般に、各COMオブジェクトを自分のストレージ又は自分のストリームに割り当て、その中に作業データを記憶させることができる。しかし、コア・プロセッサ340が幾つかのプラグインの動作を調整する本ビデオ特殊効果プロセッサのような階層システムにおいては、これまで1つのストリームを各効果プラグインに割り当てるようにしていた。

30

【0024】

これに対し、本実施態様では、単にストレージ又はストリームのいずれか一方をプラグインに割り当てることは、プラグイン・オブジェクトの設計及び動作に余りに多くの制約を課すことになると認識されていて、代わりに、コア構造部とプラグイン部との間のインタフェース定義において、各プラグインは、プラグイン設計者が望む場合、その作業データを容易に記憶できるように1つのストリームを選択でき、同じく望む場合、複数のストリームとストレージの自己の「ディレクトリ」配置の下でその作業データを記憶できるように1つのストレージを選択できるようになっている。この選択は、プラグイン・プログラムコードを作成するプラグイン設計者によって予め行われる。

40

【0025】

以下の説明において、種々のオブジェクト間の通信プロトコルについては図4及び図5に関連して説明し、コア・プログラム340のグラフィエディタ部を用いて個々の効果を連結し、複合効果を作る方法については図6～図9を参照して説明し、ビューワーウィンドー及びそれと効果ユーザインタフェース（UI）320との対話については、図10を参照して説明する。

【0026】

図4は、図3の配置における更新された効果パラメータの伝播状態を模式的に示す。この伝播状態は、ユーザが「照明」特殊効果を設定し、光源の効果が画像に加えられる場合の例である。光源は、ユーザが定めるソースと画像に関する行先位置とを有する。ユー

50

ザがこれらの位置の1つを変えたいと望む場合、現在該効果から描画されている出力を無効とし、その変化を図3の異なる象限に示す種々のオブジェクトに伝達させる必要がある。この処理過程を図4に示す。

【0027】

図4において、ユーザは、変更されたパラメータをビューワーウィンドー310を介して実際に書入れる(図10参照)。ビューワーウィンドーは特定の効果ユーザインタフェース320と対応しており、このインタフェースはまた個々の効果に関する各プラグインの一部をなす。(実のところ、1つのビューワーウィンドーを1より多い効果と対応させることができ、任意の特定時間にかかれるビューワーウィンドーをすべての効果がもつには及ばないが、都合上図3の簡略化された配置に基いて説明する。)プラグインは、「これより編集に入る(about to edit)」との通知をコア側に発する。

10

【0028】

上記の変更が行われた後、プラグインは「変更発生」メッセージを発する。これにより、一連の動作が発生及び/又は開始される。最初のステップ401で、ビューワーウィンドーは更新されたパラメータを効果UI320に伝達する。効果UI320は、「セット」コマンドに対応する効果サーバ330に発し、修正された値を効果サーバ330内のパラメータ・メモリにセットさせる。このステップを図4の402で示す。

【0029】

ステップ403で、効果サーバ330は、「取消し/やり直し」オブジェクトをコア・プロセッサ340の中に書込んで、変更の前後におけるパラメータの(効果サーバ内の)記録を指し示すハンドル又はポインタの詳細を供給する。この「取消し/やり直し」オブジェクトを受取ると、ステップ404で、コア側は、パラメータが変更され幾つかのキャッシュされたデータ出力が無効になるかも知れないという通知を全ビューワーウィンドーに流す。この通知は、パラメータの変更が行われたビューワーウィンドーに限定されない。

20

【0030】

ステップ405で、各ビューワーウィンドーは、対応する効果UI320にその処理パラメータの更新を要求するメッセージを発する。ステップ406で、効果UI320は、「ゲット」コマンドに対応する効果サーバに発して新しいパラメータを得させる。新しいパラメータは、ステップ407で効果UI320に返送される。効果UIはそれから、ステップ408でビューワーウィンドーにその変化を伝えて表示させる。

30

【0031】

一般に、処理パラメータが変わると、1つ以上の効果出力の描画をやり直す必要が生じる。図5は、描画し直しコマンドが装置内を伝播する様子を示し、これは図4の処理から引き続いて起こる。

【0032】

ステップ502ではビューワーウィンドーは描画し直しコマンドを描画マネージャ352に発する。すると、描画マネージャは描画し直しコマンドに対応する効果サーバ330に発する。効果サーバは、画像の作り直しを終えると、「終了」メッセージをコア・プロセッサ340に発する。コア側は、ステップ505でこれをビューワーウィンドーに知らせ、ステップ506及び507で、ビューワーウィンドーは効果UI320と対話し描画し直した効果出力を表示する。

40

【0033】

数個のビューワーウィンドーが開き、数個のフレームが関心の域(これは、テスト目的でユーザがビデオクリップ全体の部分集合として決めることができる。)内にある場合、処理資源をとっておくため次の優先順に従い多数の同時発生タスクとして画像が描かれる。

(i) ユーザが観察するために現在表示されている単数又は複数の画像、

(ii) 関心のある出力シーケンスの最初と最後の画像、

(iii) 関心のある出力ビデオシーケンスの残りの画像

【0034】

更に細かい段階として、描画マネージャは、効果サーバに描画し直しコマンドを発する前

50

に、シーケンス内のどの画像を描画すべきかを指定する「描画準備」メッセージを発する。効果サーバは、これに応じてその「依存性」を、即ち描画マネージャから要求される前に必要となる描画像を作成しなければならないことを通知する。これらの画像は、他の効果（例えば、非繰返し有向グラフにおける直ぐ前の効果）により描画されるものか、或いはその効果自体によって描画されるものかも知れない。この後者の例は、モーショントラッカの場合に起りうる。即ち、例えば、画像5を描くためにモーショントラッカが画像4に対する自分の描画出力を必要とする場合である。

【0035】

効果サーバから返送された該メッセージを受取ると、描画マネージャは、それらの画像を求める「描画準備」メッセージを送る。以下、従属ツリーが終わるまで同様のことが行われる。

10

各階段において、プロキシ効果は、要求された画像又は描画出力がキャッシュに記憶されたかどうかをチェックし、描画マネージャに知らせる。

【0036】

そこで、例えば、画像5を指定する描画準備メッセージがモーショントラッカに送られると、モーショントラッカは、これに回答して画像4に対する自己の描画出力を要求していることを告げる。すると、描画マネージャは、画像4に対する描画準備メッセージをモーショントラッカに送る。モーショントラッカは、これに回答して画像3を要求していることを示す。以下同様である。こうして、所要画像（画像5）が描画される前に必要な描画ジョブのリストが作成される。キャッシュに記憶されている描画出力は、描画マネージャのジョブのリストには含まれない。

20

【0037】

或る効果が前の効果の描画出力を必要とする場合には同じことが起こり、同様に一連の効果を前にさかのぼる。

この過程が終わると、描画マネージャは、現在要求されている画像が、その依存する画像がすべて描かれるまでは描かないように逆の順序で、要求されたすべてのジョブを進めてゆく。

【0038】

最適化の条件として、描画マネージャは、各効果への入力は何であるかを前記グラフから検出するのがよい。そして、効果サーバは、所定のコードを送り（例えば、ゼロ応答）簡単に「この効果に対する入力はすべて要求済み」と告げるとよい。

30

【0039】

更に、各効果サーバから、その出力が隣り合う画像と同じであるかどうかを描画マネージャに知らせるように、同じプロトコルを使用するとよい。これの簡単な例は、その出力が不変である場合の（一定）パラメータ・プラグインである。更なる例は、どれか他の効果の出力が既に準備されキャッシュに記憶されていて、連続する出力が同一かどうかを容易に検出できる場合である。かような通知に回答して、描画マネージャは、有向非繰返しグラフで後の他の効果サーバにその情報を送る。すると、その効果サーバは、（適当ならば）一連の画像の1つだけを描画し、入力が変わらない他の画像に対してその出力を繰返す。

40

【0040】

図6は、グラフ・エディタ・ウィンドー610とパレット・ウィンドー600を示す概略図である。これらは、コア・プロセッサ340の制御の下に表示スクリーン160に表示される。

パレット・ウィンドー600は幾つかのアイコン620を含み、これらの各々は、考えられる種々の効果の1つにマッピングされ、それらの効果を表している。本システムには、各効果に対するプラグインが存在する。ユーザは、マウスを用いてこれらのアイコンをスクロール可能なグラフ・ウィンドー610に「引き込む」ことができる。ユーザは、グラフ・ウィンドー内にこれらのアイコンを配置し、該ウィンドー内で図形線として示す論理リンク630で互いに連結することができる。

50

【 0 0 4 1 】

論理リンク 6 3 0 は、或る効果の出力を次の効果の入力へ送ることを表し、(この具体例では)常にグラフ・ウィンドーの下から上の方への向きをもつ。即ち、図 6 の例は、イメージロード (I / L) アイコン 6 4 0 の出力を照明効果 6 5 0 に送ることを示す。

【 0 0 4 2 】

ユーザがグラフ・ウィンドー内に上記の図形線を描くと、コア・プロセッサ 3 4 0 は、論理リンクを設定して描画出力が或る効果プラグインから他のプラグインに送られる経路を決定する。

これより、図 7 を参照して図形線を作成する方法を説明する。それから、図 8 を参照して論理リンクを説明する。

10

【 0 0 4 3 】

図 7 において、ユーザが照明効果を(例えば、マウスのクリックで)選択すると、アイコン 6 5 0 からマウスポインタ 7 3 0 に向かう動かせる図形線 7 2 0 が得られる。マウスポインタが混合効果アイコン 7 0 0 に近づくと、その混合効果アイコンが拡大され、即ち拡大線 7 1 0 に囲まれ、その最下部に 2 つの入力ポート 7 4 0 が現れる。マウスポインタが入力ポートの一方に近づくと、図形線 7 2 0 はその入力ポートに飛びつき、マウスクリックで所定場所に固定される。これにより、効果 6 5 0 と効果 7 0 0 の間に論理的及び図形的な連結が確立される。

【 0 0 4 4 】

これらの論理的及び図形的な連結が確定されると、ユーザは、グラフ・ウィンドー内で効果アイコンが連結されたグループ 8 0 0 を「箱に入れる」ことができる(図 8)。ここで、「箱に入れる」とは、コンピュータ・マウスを使って標準的な方法で上記グループの周りに箱を描くことである。(これを実現する 1 つの方法は、箱の左上隅でクリックして保持し、マウスを右下隅に引き、これからマウスボタンを放すことである。これは、複数の画面のオブジェクトを選択する標準的な方法である。)

20

【 0 0 4 5 】

ユーザはそれから、連結された効果のグループをパレット領域に引き込むことができる。これにより、該グループの入力から形成される一連の入力と、該グループの出力から形成される一連の出力とを有する新しい複合効果アイコン 8 1 0 が作成される。論理的にいえば、効果アイコン 8 1 0 は、特定のプラグインにマッピングされないで、特定の方法で相互連結されたプラグインの連結グループにマッピングされる。

30

【 0 0 4 6 】

複合効果アイコン 8 1 0 は、それからパレット・ウィンドーの一部となり、ユーザによりグラフの設計に使用される。ユーザがあとで複合アイコン 8 1 0 を利用したい場合、単にそれをグラフ・ウィンドーの適所に引き込むだけでよい。効果アイコン 8 1 0 は、グラフ・ウィンドー上でも 1 つのアイコンのままであるのがよいが、他の具体例では元のグループ 8 0 0 に引き伸ばすこともある。或いはまた、1 つのアイコンとして圧縮した形で表示してもよい。ただし、ユーザが引き伸ばしボタンをクリックして元のアイコン・グループ 8 0 0 を観察できるように、引き伸ばしボタン付きの表示とするのがよい。いずれにしても、アイコン 8 1 0 によって与えられる複合効果は、元のグループ 8 0 0 のコピーである。

40

【 0 0 4 7 】

図 9 は、この方法の根底にあるデータメモリを示す。図 9 において、アイコン 8 5 0 は、パレット領域 6 0 0 におけるパレットアイコン 6 2 0 からグラフ・エディタ領域 6 1 0 に引き込まれたものである。

パレット領域 6 0 0 について述べると、図 3 に示したパレット 3 5 6 には、ルート(根) 8 6 0 及び該ルートからぶら下がる個々のデータ項目 8 7 0 を有するツリー(木)のように配置されたデータ構造が記憶される。各データ項目 8 7 0 は、1 つの効果アイコン 6 2 0 を表すが、効果 8 7 5 のような複合効果の場合は別である。この場合、その効果を形成する効果アイコン(3 a , 3 b)は、そのデータ項目 8 7 5 からぶら下がる下部構造とし

50

て配置される。

【 0 0 4 8 】

グラフ・エディタ領域には、効果を記憶する類似のデータ構造が存在する。ここでは、ただ1つの効果 8 8 5 がぶら下がったルート 8 8 0 が示されている。グラフ・エディタ領域において幾つかの効果をグループとして一緒に集めパレットに引き込む場合、それらは更に、構造 8 7 5 に似た複合効果構造となる。

【 0 0 4 9 】

図 1 0 は、ビューワーウィンドーの概略図である。ビューワーウィンドーは、画像表示領域 9 0 0、種々の「属性ページ」 9 1 0、効果制御部 9 2 0 (ここでは、照明効果の例における位置決め十字線を示す。)及び「バーボタン」 9 3 0 を具えている。

ビューワーウィンドーの基本配置は、コア構造部によって設定され、効果ごとに基準となるものである。ただし、属性ページ 9 1 0 を用いて調整できる個々の項目は、特定の効果に対応する効果ユーザインタフェース (UI) 3 2 0 によって設定される。効果 UI 3 2 0 はまた、制御部 9 2 0 に対する表示の細目を与える。

【 0 0 5 0 】

図示の例では、十字線 9 2 0 は、照明効果における光のソース又は目標位置を決定する。ユーザは、コンピュータ・マウスを用いて十字線を移動させることができる。十字線を移動させると、当該制御に関連するパラメータ (x, y) 値が変わり、図 4 の手順 (パラメータ値の更新) が開始される。その手順の最後のステップ 4 0 8 (図 4) で、効果 UI は訂正されたパラメータ値をビューワーウィンドーに発する。その段階で、十字線はその新しい位置に再表示される。したがって、ユーザには移動操作が十字線を最終の位置に動かしたように見えるが、実のところは、移動操作がパラメータを図 4 に示す経路によって更新させ、その結果十字線の移動が生じたのである。

【 0 0 5 1 】

図 1 1 は、作動ソフトウェアの初期の配置を模式的に示すものである。これは、本装置の特定の動作期間において何等かの描画を行う前の状態を表す。

プラグインは、「ダイナミック・ロード・ライブラリー (DLL)」のようなウィンドーズ・オペレーティング・システムによって実現できる。DLL は一般に、プログラムコード、データ及びサブルーチン・ライブラリを含む大きなファイルである。従来、記憶を保存し、システムの性能を改善するために、DLL によって取扱われる特定の処理の実行又は開始のため最初に必要なときに DLL をメモリに格納していた。この具体例では、記憶保存及びシステム性能改善のこのアイデアを、更に一歩進めて用いる。

【 0 0 5 2 】

即ち、パレット領域から1つの効果アイコンを最初に取出すとき、従来は、その効果に対応する DLL をメモリに格納して、コア・プロセッサ 3 4 0 にグラフを作成するための十分な情報 (例えば、他の効果アイコンとの相互連絡性) を提供していた。

【 0 0 5 3 】

この具体例では、当該効果に対する DLL をその段階では格納しない。その代わりに、該効果を表すいわゆる「メタデータ」 1 0 0 0 を格納する。このメタデータは、コア・プロセッサに、該効果と他の効果との相互連絡性を定める情報 (例えば、入力と出力の数) を提供する。これにより、コア・プロセッサは何も DLL を格納することなくグラフを作成することができ、それらが絶対に必要となるまで大きなファイルを格納しないので、メモリの節約になる。

【 0 0 5 4 】

ビューワーウィンドーが或る効果に関連して開かれているか、又は複合効果が他の手段によって実行されている場合は、DLL が格納され、メタデータは破棄されるか又は無視される。

図 1 2 ~ 1 4 は、特に本システムの自動化を容易にする効果プラグインの機能的特徴を模式的に示すものである。

【 0 0 5 5 】

図 1 2 は、以前に提案されたプラグインの模式図である。この効果プラグインは、画像情報（「クリップ」1300として示す。）を受けて、3つの処理パラメータ P 1 , P 2 及び P 3（照明位置など）に基いて動作する。図 1 2 のプラグインでは、パラメータ値は、プラグイン内に、即ちプラグインの一部として書かれた注文プログラムコードにより設定される。このため、例えば、パラメータが時間と共に変化する動画システムや、照明位置のようなパラメータが、モーショントラッカのような他の効果によって変化する装置では、効果プラグイン内に余分なコードを加えたものや、多数の効果プラグインを複合したものがしばしば必要となり、パラメータの総合的な制御が極めて難しくなる。

【 0 0 5 6 】

図 1 3 は、本発明の実施形態による他の方法を示す模式図である。ここで、各パラメータは、「主」効果プラグイン 1330 に連結された別々のプラグイン 1320 により、効果間の連結が上述のグラフ・エディタにおいて定められたのと同様にして定められる。実のところ、上述の説明はプロセス全体を簡略化して行ったものであるが、簡略化は説明を助けるためである。

【 0 0 5 7 】

パラメータ・プラグインは、例えば、グラフ・エディタ及びパレットにおける「ページから離れた」スクリーン位置にそれらを表示することにより、通常はユーザから隠されている。

そこで、或る効果を自己充足の非動画式に（即ち、他の効果からパラメータ値を移入することなく）作動させようとする場合、主効果のビューワーウィンドーを用いて各パラメータ・プラグイン 1320 に対するパラメータを設定する。

【 0 0 5 8 】

パラメータを他の効果の出力、例えば動き追跡効果によって与えられる位置の値により決めようとする場合、主効果プラグイン 1330 と適切なパラメータ・プラグイン 1320 との間の論理リンクを切り離し、動き追跡効果とのリンクを開始させればよい。

この方式が動画化にいかに関与するかを理解するために、図 1 4 を参照する。

【 0 0 5 9 】

図 1 4 は、最初に図 3 に示したコア側とプラグイン側の間を左右に分けた各部分を示す。図 1 4 の左側には、1つのプロキシ効果（PE）1340 が「主」効果サーバ 1350 に対して設けられている。また、複数のプロキシ効果 1360 がパラメータ・プラグイン 1320 の各々に対して設けられる。これらのプロキシ効果 1360 は、プロキシ効果 1340 よりずっと簡単なものであり、プロキシ効果 1360 とパラメータ・プラグイン 1320 との間の通信は、プロキシ効果 1340 と効果サーバ 1350 との間の通信プロトコルの簡略化された部分集合を使用して行う。

【 0 0 6 0 】

実際のところ、プロキシ効果 1360 は単一のデータ値（非動画システムの場合）でよく、或いは動画システムではそれらの値のリストでもよい。動画システムでは、データ値のリストは「キーフレーム」値、即ち連続する個々の画像のために設定されたデータ値として表現でき、間に介在する値は、線形もしくはユーザが決めた非線形補間に従ってコア側により補間される。よって、動画を、各プラグイン内に注文動画ソフトウェアを書込むことなく、特に簡単で便利な方法で作成することができる。

【 0 0 6 1 】

この説明を先に効果間の依存性について述べた説明と結びつけると、描画マネージャから「描画準備」メッセージを効果サーバ 1350 が受取ったとき、それに応答して、出力を供給する前にその入力が必要である、と告げることができる。効果の入力には勿論パラメータ・プラグインが含まれるので、次の段階で描画マネージャは描画準備メッセージを各パラメータ・プラグインに送ることになるであろう。パラメータ・プラグインが単一の値しか含まないか、或いは現在の画像がキーフレームであれば、パラメータ・プラグインは直ちに適切なパラメータを描画時に供給することができる。しかし、パラメータ・プラグインが動画データを含み、現在の画像がキーフレームでなければ、パラメータはまず

10

20

30

40

50

補間してから効果に使用しなければならない。

【 0 0 6 2 】

図 1 5 は、システム・キャッシュ 1 1 0 0 の模式図である。これは全キャッシュ領域の概略図であり、実のところ、先に述べたように、キャッシュはまた、夫々のプロキシ効果に対応した複数の個々のキャッシュと見ることもできる。しかし、かような個々のキャッシュにメモリ資源が動的に割当てられるので、図 1 5 の表示も正しいものである。

【 0 0 6 3 】

キャッシュは、システム・メモリ 1 5 0 の中に設けられ、画像 1 1 1 0 及び効果からの非画像描画出力 1 1 3 0 (例えば、動き追跡効果の場合の動きベクトル)を記憶することができる。

10

【 0 0 6 4 】

キャッシュの意図は、有向非繰返しグラフ内の各効果の描画出力(これが画像であるかどうかにかかわらず)を記憶することである。こうすると、或る効果の有向非繰返しグラフにおける特定の位置が変更された場合、その位置よりあとの効果を描画し直して新しい出力を供給する必要がなくなる。代わりに、キャッシュされている出力を再使用できる。更なる利点は、特定の効果(開かれたビューワー・ウィンドーに表示されたもの)の出力をパラメータ変更の前後に記憶することにより、取消し/やり直し動作を助け動作の速度を早めることである。対応するパラメータ変更も記憶されるので、キャッシュメモリ 1 1 0 0 から適切な素材を格納することにより、パラメータの変更を簡単に取消したり、やり直したりすることができる。これは、変更が行われたときに効果サーバによって書込まれた取消し/やり直しオブジェクトの制御の下に行われる。

20

【 0 0 6 5 】

画像は、動きベクトルのような簡単なデータ値よりはるかに大きなメモリ空間、恐らく百万倍のメモリ空間を占める。したがって、この具体例では、キャッシュメモリが容量の限界に近づいている状態で他の画像を記憶しようとするときは、最も前にアクセスしたキャッシュ内の画像を削除して、新しく記憶する画像のための空間を作る。ただし、キャッシュ内の他のデータ、パラメータ値や非画像描画出力などは、画像に比べるとほんの僅かのメモリ空間しか使わないので、本装置の動作期間中は削除しない。よって、この情報は、該情報が動作期間を通じて有効である限り、再使用又は取消し/やり直し動作に使用できる。

30

【 0 0 6 6 】

実際には、データ項目を洗い流せるかどうかをプラグインに指定させるようにすることができる。即ち、画像データ項目は取決めにより洗い流せるものと設定し、非画像データ項目は洗い流せないものと設定することができる。

【 0 0 6 7 】

図 1 6 は、コア・プロセッサ 3 4 0 及び効果サーバ 3 3 0 間の非同期/同期コンバータ(プラグイン・インタフェース) 1 2 0 0 の模式図である。

コンバータ 1 2 0 0 は、描画マネージャから非同期再描画コマンドを、「行うべき」列、即ち行うべき描画ジョブのリストの形で受取る。1つのジョブが終わると、「終了」メッセージがコンバータ 1 2 0 0 から描画マネージャに返送される。

40

コンバータ 1 2 0 0 は、非同期のジョブ要求を受け、同期要求を適切なソフトウェア・プラグインに発する。これは、インタフェース 1 2 0 0 が制御「スレッド」(ウィンドーズ用語)をソフトウェア・プラグインに回送することを意味し、これが、該ジョブが完了するまで該スレッドの制御を保持することになる。その時のみ、ソフトウェア・プラグインが該スレッドをインタフェースに返送し、該インタフェースが「終了」メッセージをコア側に発して応答する。

【 0 0 6 8 】

初期設定の際、コア・プロセッサは、各プラグイン(又はむしろ当該プラグインに関連するメタデータ)に問合せ、該プラグインが同期又は非同期通信を処理できるかどうかを決定する。ハードウェア・プラグイン(例えば、特定の 방법으로描画するための周辺カード

50

）又は恐らく異なる機械で作動している非同期ソフトウェア・プラグインをソフトウェア・プラグインの代わりに組込めば、ハードウェア・アクセラレータはこのような動作に極めてよく適しているので、該プラグインはコア側（実のところ、非同期的に描画タスクを開始する描画マネージャ）と非同期インタフェースを介して対話を行う。したがって、この場合、コンバータ1200は迂回される。

【0069】

コンバータは、コア側の一部として又は各関連プラグインの一部として実施してもよい。よって、コンバータ1200を2つのソフトウェアの間に設けるという反直観的な方法により、将来専用ハードウェアに改良される効率的なハードウェア・インタフェースが提供される。

10

【0070】

【発明の効果】

以上説明したとおり、本発明によれば、連結した効果をユーザが作成したとき、それらを後日使用するため1つのグループとして記憶し検索できる便利な方法を得ることができる。

【図面の簡単な説明】

【図1】デジタルビデオ特殊効果装置の概略図である。

【図2】図1の装置の作動ソフトウェアの編成を示す概略図である。

【図3】図1の装置の作動ソフトウェアの編成の詳細を示す模式図である。

【図4】図1の装置における更新された効果パラメータの伝播状態を示す模式図である。

20

【図5】図1の装置における再描画コマンドの伝播状態を示す模式図である。

【図6】グラフ・エディタ・ウィンドー及びパレット・ウィンドーを示す概略図である。

【図7】グラフ編集動作の例を示す説明図である。

【図8】複合効果アイコンの作成例を示す説明図である。

【図9】複合効果のファイル構造を示す模式図である。

【図10】ビューワーウィンドーの概略図である。

【図11】作動ソフトウェアの初期配置を示す模式図である。

【図12】以前に提案された効果プラグインの模式図である。

【図13】新しい形式の効果プラグインの模式図である。

【図14】図13の効果プラグインにおける効果サーバとプロキシ効果との関係を示す模式図である。

30

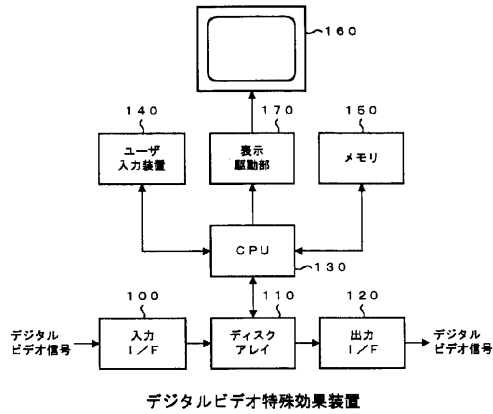
【図15】システム・キャッシュを示す概略図である。

【図16】プラグイン・インタフェースを示す概略図である。

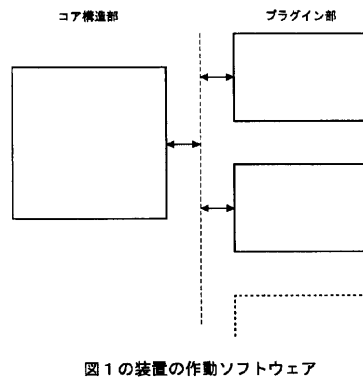
【符号の説明】

160 表示スクリーン、600 パレット領域、610 グラフ領域、620
モジュール・スクリーン・アイコン、140 ユーザ操作可能手段、800 連結
されたスクリーン・アイコンのグループ、810 複合処理スクリーン・アイコン

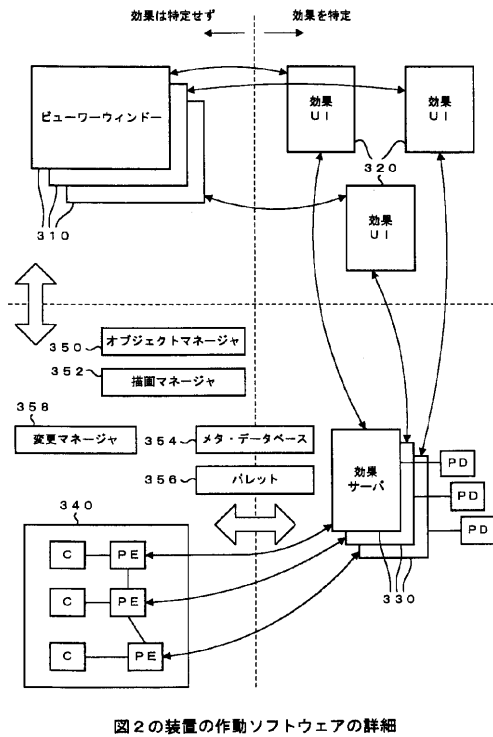
【図1】



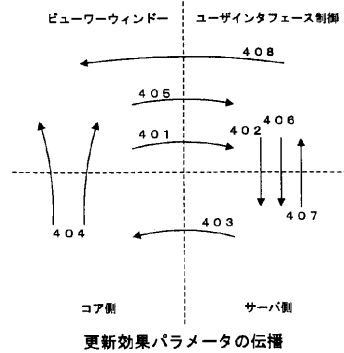
【図2】



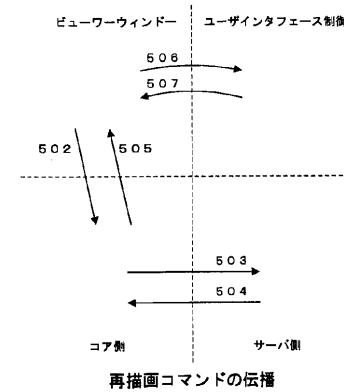
【図3】



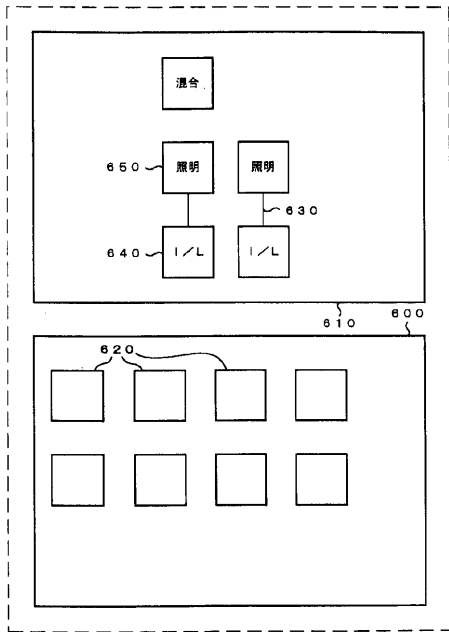
【図4】



【図5】

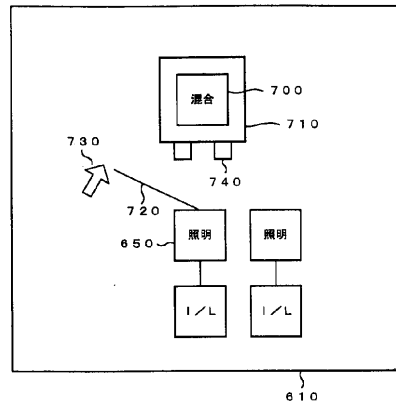


【図6】



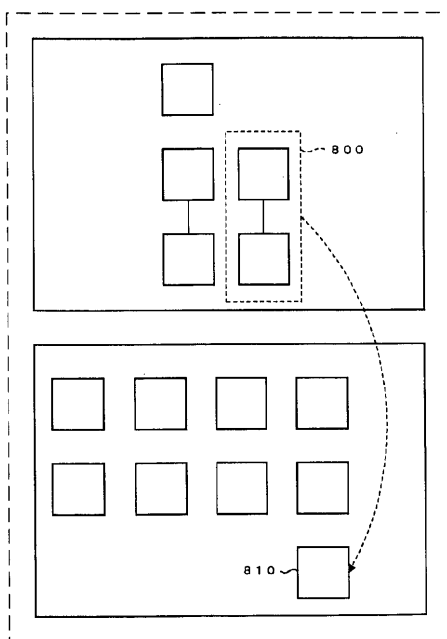
グラフ・エディタ・ウィンドー及びパレット・ウィンドー

【図7】



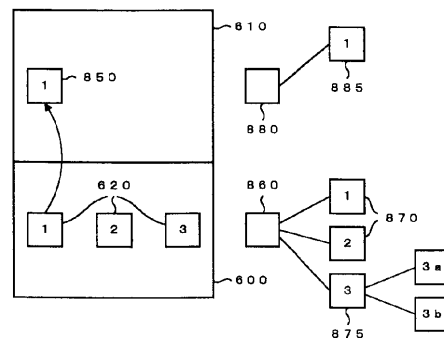
グラフ編集動作の例

【図8】



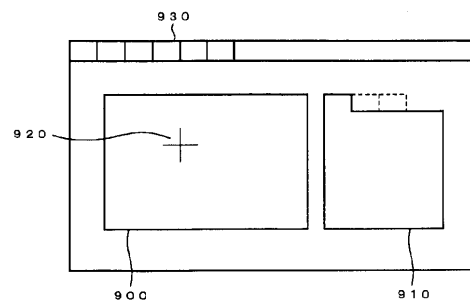
複合効果アイコンの作成例

【図9】



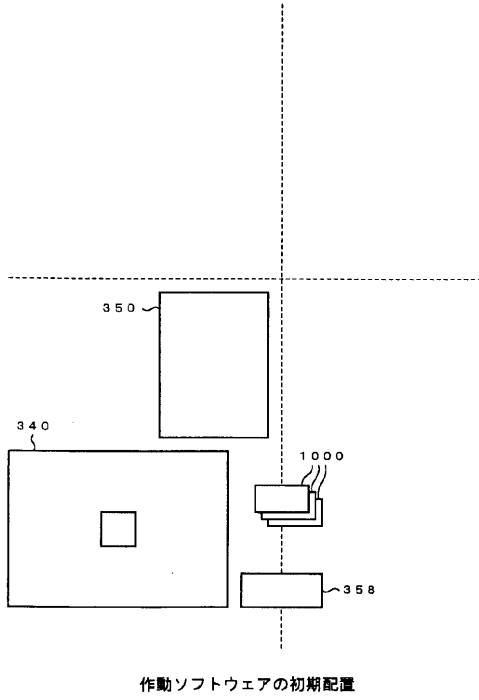
複合効果のファイル構造

【図10】

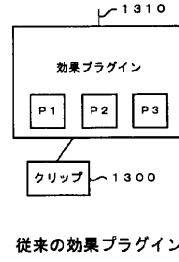


ビューワーウィンドー

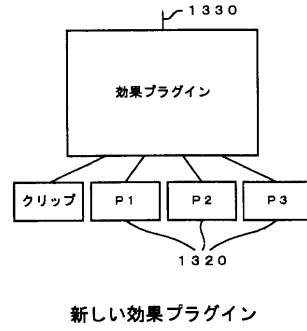
【図11】



【図12】



【図13】



【図14】

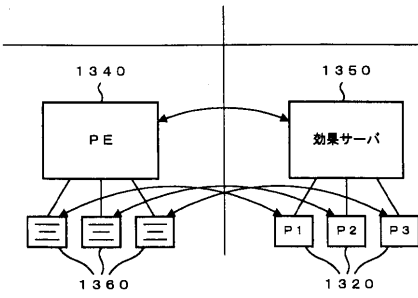
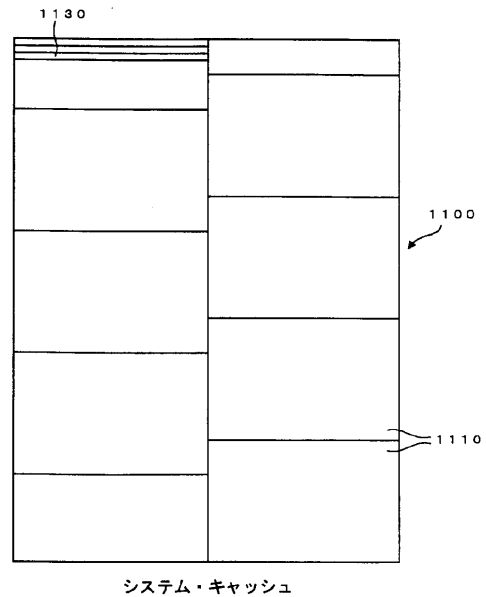
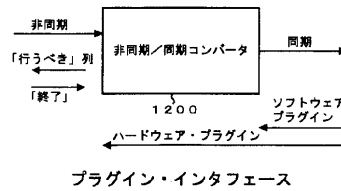


図13の効果プラグインにおける効果サーバとプロキシ効果との関係

【図15】



【図16】



フロントページの続き

- (72)発明者 グールド, アントニー ジェームズ
イギリス国 ハンプシャー, ベーキングストーク, ジョージ ストリート 64
- (72)発明者 フォスタ, リチャード ダニエル
イギリス国 ハンプシャー, フォーディングブリッジ, マーティン, マーティン クロス コテージ

審査官 村松 貴士

- (56)参考文献 特表平06-503695(JP, A)
特開平07-098651(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06T 13/00
H04N 5/262 - 5/28
G06F 9/06