(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3630726号
(P3630726)
(45)発行日　平成17年3月23日(2005.3.23)　(24)登録日　平成16年12月24日(2004.12.24)

(51)Int.Cl.[7]　F I
G06T 11/60　G06T 11/60　100A
G06T 11/80　G06T 11/80　B

請求項の数 1　(全 25 頁)

(21)出願番号　特願平6-235297
(22)出願日　平成6年9月29日(1994.9.29)
(65)公開番号　特開平7-282284
(43)公開日　平成7年10月27日(1995.10.27)
審査請求日　平成13年8月1日(2001.8.1)
(31)優先権主張番号　133422
(32)優先日　平成5年10月8日(1993.10.8)
(33)優先権主張国　米国(US)

(73)特許権者　596170170
ゼロックス　コーポレイション
ＸＥＲＯＸ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国　コネチカット州　スタンフォード、ロング・リッジ・ロード　８００
(74)代理人　100079049
弁理士　中島　淳
(74)代理人　100084995
弁理士　加藤　和詳
(72)発明者　デニス　エル．ヴェナブル
アメリカ合衆国　１４６２３　ニューヨーク州　ロチェスター　アカデミー　ドライヴ　５５

最終頁に続く

(54)【発明の名称】電子的構造化画像ジェネレータ

(57)【特許請求の範囲】
【請求項１】
ディスプレイおよびプリントされることが可能な少なくとも一つの出力構造化画像を用いて少なくとも一つのラスタ画像の表示を発生する手段であって、少なくとも一つの前記出力構造化画像が少なくとも一つの構造化画像オブジェクト上で少なくとも一つの画像処理動作を実行した結果として発生される、少なくとも一つのラスタ画像の表示を発生する手段と、
少なくとも一つの構造化画像を構築およびディスプレイする手段であって、前記構造化画像が、少なくとも一つの前記構造化画像オブジェクトと、ペーストボードと、構造化画像定義と、を有し、該ペーストボードは、少なくとも一つの前記構造化画像オブジェクトが描画されるフレームを定義し、該ペーストボードは、該ペーストボードに対して、どのように少なくとも一つの該構造化画像オブジェクトが描画されるか、を定義する少なくとも一つの画像属性を有し、該構造化画像定義は、少なくとも一つの該構造化画像を発生するために使用される構造化画像形成処理を記述し、該構造化画像定義は、少なくとも一つの構造化画像オブジェクト上で実行され、少なくとも一つの前記出力構造化画像を形成するために併合される、少なくとも一つの相互接続された画像処理動作を含む、少なくとも一つの構造化画像を構築およびディスプレイする手段と、
前記ペーストボードの少なくとも一つの前記属性を有する前記構造化画像定義を形成および変更するため、前記発生する手段を制御する手段、および、前記構築する手段および前記ディスプレイする手段を制御する手段であって、前記構造化画像を構築およびディスプ

レイする手段によって発生される前記構造化画像定義に応答して、少なくとも一つの前記出力構造化画像を発生する手段が前記制御手段によって前記少なくとも一つの出力構造化画像を発生するように制御され、前記構造化画像定義は前記ペーストボードの少なくとも一つの属性を有する、前記発生する手段を制御する手段、および、前記構築する手段および前記ディスプレイする手段を制御する手段と、
コンピュータプログラム命令およびオペレータ入力コマンドの少なくとも一方に応答して、前記少なくとも一つの出力構造化画像のディスプレイ、および、プリントの少なくとも一方を行うための手段と、
を有する電子的構造化画像ジェネレータ。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】

この特許書類の開示の一部は著作権保護を受ける資料を含む。この著作権所有者は、米国の特許及び商標事務局の特許出願又は記録に現れるので、特許書類又は特許開示のいづれによってもファクシミリの複写（再生）に対して意義を申し立てないが、いずれにしろ全てのあらゆる所有権の権利を保留している。

【０００２】

本発明は階層的及びデバイス独立フォーマット内のオブジェクトの集合として複雑なカラーラスタ画像を表現するための画像処理装置及び／又はジェネレータに係り、特に、ソースデータだけでなく、それらをレンダリングするために必要とされる画像処理動作を含み且つ適切なユーザインターフェースを用いて、再編集の可能性、カスタマイゼーション（個別化）及びエンハンスメント、自動的画像アセンブリ、及び高性能の画像形成を支援するＳＩ（構造化画像）フォーマットに関する。

【０００３】

【従来の技術】

【０００４】

現在使用されている業務用で入手可能なパッケージは相当な量のメモリを操作する必要がある。例えば、Ａｄｏｂｅ　Ｐｈｏｔｏｓｈｏｐ　（アドーベ・フォトショップ）（登録商標）はインタラクティブな（対話型）編集中に、フルサイズ（全寸法）の三つの複写を記憶する。従って、一般的な２４－ビットのカラー画像（３００ｄｐｉで８．５×１１インチ）が画像コピー毎に２４メガバイトのメモリを必要とし得る。

【０００５】

さらに、業務用ソフトウェアは、大型画像を編集する時は（特に）反応が遅く貧弱な性能（パフォーマンス）を呈することが多い。また、上記に説明されているように、Ｐｈｏｔｏｓｈｏｐ　（登録商標）は（その大きさにかかわらず）画像の三つの複写を記憶しなければならない。全てのインタラクティブ動作が完全画像を処理するので、性能は極めてゆっくりである。

【０００６】

さらに、一般的な業務用ソフトウェアは完全なラスタとして出力画像を記憶する。従って、画像の新しいバージョンごとに、完全なラスタが書き出されなければならず且つこれらのバージョン内に含まれた動作をアンドゥ（取り消す）することはできない。上記のように１画像あたり２４バイトでは、ディスクの要求条件が広範囲にわたる。さらに、このような画像をアーカイブ（保存）することによって拡張記憶空間も必要とされる。

【０００７】

また、複数のカスタマイズされた画像を構築するためにジョブ・テンプレートを作成する能力を有する業務用印刷アリーナ（環境）における画像プロセッサに対する要求が決定的に存在している。このようなジョブ・アプリケーション（適用業務）・テンプレートの作成を可能とする業務用画像処理パッケージは殆ど存在していない。さらに、Ｐｈｏｔｏｓｈｏｐ　（登録商標）のような業務用画像処理ソフトウェアパッケージは出力デバイス用に最適化された出力画像を形成する能力が限られている。一般には、カラー補正及び簡単

な解像度の変換が、適用できる全ての能力といえる。
【０００８】
さらに、一般的な業務用画像処理ソフトウェアにおいて、オペレータによって実行される画像処理動作のリストを編集する能力も非常に限られる。例えば、Ｐｈｏｔｏｓｈｏｐ（登録商標）は一度だけのアンドゥ（取り消し）動作を可能にするにすぎない。また、一般的な業務用のソフトウェアは、画像処理の最適化を許容しない。要するに、画像を一旦形成すると、動作を結合し且つ高解像度画像でのパフォーマンスを高める能力を失う。
【０００９】
最後に、構造化ドキュメント理論とそのテクノロジが製品やサービスにどのように影響し得るかについて業界の関心が高まっている。現在の構造化ドキュメント理論は、ラスタ画像を、階層的性質又はそれに対応する画像処理動作を有さない単純な内容ブロックとみなす。ページ指向記述に関しては、１９９１年３月発行のアイトリプリイー（ＩＥＥＥ）の「コンピュータ・グラフィックス＆アプリケーション・ジャーナル」の７１～７８頁のＪ．　Ｓｃｈｏｎｈｕｔ　（ションハット）、Ｈ．　Ｗｉｅｄｌｉｎｇ　（ウィードリング）、及びＶ．　Ｓａｍａｒａ　（サマラ）著の”Ｃｏｎｓｔｒｕｃｔｉｖｅ　Ｐａｇｅ　Ｄｅｓｃｒｉｐｔｉｏｎ”　を参照されたい。構造化文書の一般的な説明に関しては、１９８９年のエレクトロニック出版のケンブリッジシリーズにおいて、Ｊ．　Ａｎｄｒｅ（アンドレ）、Ｒ．　Ｆｕｒｕｔａ　（古田）、及びＶ．　Ｑｕｉｎｔ（クイント）らによって発表された”Ｓｔｒｕｃｔｕｒｅｄ　Ｄｏｃｕｍｅｎｔｓ”を参照せよ。
【００１０】
【発明が解決しようとする課題】
従って、本発明の目的は、最小のメモリ要求によってディスプレイ解像度画像上で動作し、且つ高解像度、高品質のプリント画像形成が可能なＳＩフォーマットを提供することによって、上記の欠点を取り除くことである。ＳＩ（構造化画像）は、レンダリング（描画）を出力するためにオブジェクト、記憶、編集及び／又は準備並びに分散及び／又は移動の獲得の全域にアドレス可能なベーステクノロジである。
【００１１】
本発明の他の目的は、ディスプレイ解像度画像上でインタラクティブに動作し、これにより、良好なインタラクティブ・パフォーマンスを有することが可能なＳＩフォーマットを提供することである。さらに、高解像度出力は、単に、ＳＩが高解像度でレンダリングされるか又はラスタ化されることを要求するだけで発生し得る。レンダリング処理は、特定のデバイスに対してレンダリングするのに適したソース画像及びデバイス依存情報を自動的に使用する。他の画像エディタはディスプレイ解像度画像を使用することができるが、ＳＩはタスクによって画像の解像度を選ぶ。
【００１２】
本発明の他の目的はまた、構成要素オブジェクトへのポインタを含むテキストファイルとして画像を記憶することである。出力ラスタはディスプレイ又は印刷が要求されるまで発生しない。従って、ＳＩの何百ものバージョンが一般に一つのラスタ画像によって要求されるディスク空間の容量の中に存在し得る。ＳＩはＳＩファイル内に構成要素オブジェクトのラスタを記憶しないので、例えば、クリップ（クリッピング）技術のようなアーカイブな（保存的）画像の使用は、ソース（資源）の一つの複写だけが要求された時に一層効率的となる。
【００１３】
本発明の目的はさらに、構成要素オブジェクトと画像処理パラメータの両方に対して定義されない参照を有するように作成され得るＳＩを提供することである。このようなＳＩは、例えば、コマンドラインアーギュメント（引数）として定義されない情報を提供することによってレンダリングされ得る。ＳＩは種々の入力データによって何度もレンダリングされ、カスタム出力ラスタを発生する。定義されない参照を含む有効ＳＩを有する能力は、カスタマイゼーションと自動画像アセンブリを動作可能にする。
【００１４】

また、本発明の目的は、ＳＩに「ネイティブ」形式で全ての構成要素データを記憶させることである（例：テキストはＡＳＣＩＩ （アスキー）として記憶され、グラフィックスはＣＧＭ（コンピュータ・グラフィックス・メタファイル）フォーマット又はいくつかの他の適切なフォーマットその他において記憶される）。適切なレンダリング手順は、レンダリングを最適化するためにデバイス依存特性を用いてネイティブ形式をラスタへ変換する。構成要素データの適切な基準化（スケーリング）はＳＩのレンダリングに特有である。

【００１５】

最後に、さらに、本発明の目的は、オリジナルの構成要素データへのポインタ、及び画像に用いられる全ての画像処理動作の記述を含むＳＩを提供することである。動作に先立つ編集は、テキストファイルを編集するのと同じくらい単純であり得る。ＳＩをベースとしたインタラクティブ・エディタは、動作の編集をテキスト編集又は構造化画像エディタのいずれかを介してインタラクティブに可能にすることができる。画像処理動作は適切なユーザコマンドによって追加され、削除され、且つ変更され得る。

【００１６】

【課題を解決するための手段】

上記及び他の目的を達成し、且つ上記の欠点を克服するため、一組の表現及びペースト（貼付け）ボードによって記述されるＳＩをその最も単純な形式において定義する。表現はＳＩの内容を指定し、ペーストボードはＳＩの「フレーム」を指定する。複数のラスタ表現の概念は技術上理解されており、現在いくつかの使用可能なアプリケーションに使用されている。しかしながら、ＳＩの場合、複数の表現のアイデアは、テキスト、グラフィックスのような非ラスタデータ並びに画像処理及び複数の構成要素オブジェクトのマージング（併合）を指定する画像複合（コンポジット）を含むために拡大される。テキスト及びグラフィックスのような構成要素オブジェクトは、編集可能性のためにネイティブ形式で記憶される。有効な表示の要求条件は、ラスタ形式へレンダリングすることができ且つ単一ＳＩの全ての表現がレンダリングされた時に同じ「外見」を有すること、即ち全ての表現は本質的に類似した外見を有するべきであることである。それらは異なる次元又はエンコーディング（符号化）を有するが、明確に同じ画像のレンダリングでなければならない。

【００１７】

本発明の態様は、ディスプレイおよびプリントされることが可能な少なくとも一つの出力構造化画像を用いて少なくとも一つのラスタ画像の表示を発生する手段であって、少なくとも一つの前記出力構造化画像が少なくとも一つの構造化画像オブジェクト上で少なくとも一つの画像処理動作を実行した結果として発生される、少なくとも一つのラスタ画像の表示を発生する手段と、少なくとも一つの構造化画像を構築およびディスプレイする手段であって、前記構造化画像が、少なくとも一つの前記構造化画像オブジェクトと、ペーストボードと、構造化画像定義と、を有し、該ペーストボードは、少なくとも一つの前記構造化画像オブジェクトが描画されるフレームを定義し、該ペーストボードは、該ペーストボードに対して、どのように少なくとも一つの該構造化画像オブジェクトが描画されるか、を定義する少なくとも一つの画像属性を有し、該構造化画像定義は、少なくとも一つの該構造化画像を発生するために使用される構造化画像形成処理を記述し、該構造化画像定義は、少なくとも一つの構造化画像オブジェクト上で実行され、少なくとも一つの前記出力構造化画像を形成するために併合される、少なくとも一つの相互接続された画像処理動作を含む、少なくとも一つの構造化画像を構築およびディスプレイする手段と、前記ペーストボードの少なくとも一つの前記属性を有する前記構造化画像定義を形成および変更するため、前記発生する手段を制御する手段、および、前記構築する手段および前記ディスプレイする手段を制御する手段であって、前記構造化画像を構築およびディスプレイする手段によって発生される前記構造化画像定義に応答して、少なくとも一つの前記出力構造化画像を発生する手段が前記制御手段によって前記少なくとも一つの出力構造化画像を発生するように制御され、前記構造化画像定義は前記ペーストボードの少なくとも一つの属

性を有する、前記発生する手段を制御する手段、および、前記構築する手段および前記ディスプレイする手段を制御する手段と、コンピュータプログラム命令およびオペレータ入力コマンドの少なくとも一方に応答して、前記少なくとも一つの出力構造化画像のディスプレイ、および、プリントの少なくとも一方を行うための手段と、を有する電子的構造化画像ジェネレータである。

【００１８】

【実施例】

構造化画像形成（ＳＩ）は構成要素画像（即ち子オブジェクト）と出力ラスタ画像をレンダリングするために必要な画像処理動作（ＩＰＯ）からなる複合ラスタ画像の階層的記述である。ラスタとしてレンダリングすることができるあらゆるデータタイプはＳＩの潜在的な構成要素である。ＳＩは、プリントされたドキュメント（文書）に組み込むための複雑な画像の簡単且つ迅速な形成と操作のためのイネーブラ（ｅｎａｂｌｅｒ）である。

【００１９】

ＳＩは従来のハードウェアシステム１００に実行され（図１参照）、この従来のハードウェアシステム１００は、マイクロプロセッサ１１０でラン（実行）する一つ又はそれより多くのソフトウェアプログラムによってシステム１００の種々の他の構成要素から信号を受け取ったり種々の他の構成要素へ信号を出力したりするためのマイクロプロセッサ１１０を含む。ユーザ入力デバイス１２０、例えば、マウス、キーボード、タッチスクリーン、及び／又はそれらの組み合わせは、（ＳＩに）例えば、制御信号を供給することによって、オペレータがＳＩとインターフェースするのを可能にするために提供される。ビデオディスプレイモニタ又はプリンタのような画像出力デバイス１３０がＳＩデータを出力するために提供され得る。スキャナのようなＳＩ入力デバイス１４０が次いで画像ソースとしてメモリに記憶される画像を走査するために提供される。画像ソースはまた、例えば、ディスクから供給され得る。

【００２０】

マイクロプロセッサ１１０は、画像を組み立て且つ編集するためにＳＩによって使用されるオペレーティングプログラム及び画像データを記憶するためのリードオンリーメモリ（ＲＯＭ）１５０を含む。ランダムアクセスメモリ（ＲＡＭ）１６０は種々のオペレーティングプログラムをランするために提供され且つ追加のファイル１７０はＲＡＭオーバーフローのために提供され得る。

【００２１】

しかしながら、重要なことは、ＳＩが、種々のハードウェアの構成要素がネットワーク化され且つそれらが同じ位置に留まったり留まらなかったりするネットワーク・システムだけでなく、組み込みのシステム（図示せず）で実行することが可能なことである。ハードウェアシステムが配線される必要はない。上記の従来のハードウェアシステムは例としてあげているにすぎず、組み込み又はネットワーク・システムほど好ましくはない。

【００２２】

ＳＩはラスタ画像へレンダリングできるデータの表現である「オブジェクト」と、レンダリングされるべきオブジェクトに対する「フレーム」を表す「ペースト（貼付け）ボード」によって定義される。ＳＩはＳＩ記述内で指定される絶対サイズ（寸法）を有さない。各サブオブジェクトのサイズはサブオブジェクトの親に相対するサイズとしてペーストボード内に記憶される。

【００２３】

ＳＩオブジェクトをラスタ画像へ変換するため、以下に示すＲｅａｄＯｂｊｅｃｔ（）（オブジェクトを読み取れ）レンダリング手順が、例えば、パラメータとしてオブジェクト、ペーストボード、及び親オブジェクトの絶対寸法を有するユーザ入力デバイス１２０を用いて適切なコマンド命令を介して呼び出される。ＳＩオブジェクトとあらゆる子オブジェクトは適切な出力サイズへ自動的にレンダリングされる。ＳＩは構造化画像定義（ＳＩＤ）ファイルによって記述される。

【００２４】

ＳＩオブジェクトが適切なＳＩ走査ソフトウェアを用いてそのＳＩＤへ自動的に走査されてもよいことが理解されよう。従って、インタラクティブな編集だけが、印刷用にＳＩＤを作成し又は準備するための決して唯一の方法ではない。

【００２５】

ＳＩオブジェクトはラスタ画像へレンダリングされ得るデータの表現である（図２に参照）。このデータは、ＴｉＦＦ（タグ・イメージ・ファイル形式）ファイルのような単純なラスタ画像又はサブオブジェクトの階層的集合のようなより複雑なデータ、及び関連するラスタ処理動作を表現することができる。使用可能なオブジェクトデータタイプは、

１）ラスタ画像－ＴｉＦＦ、ＲＦＳ　、又は他のディジタル・ディスプレイ、

２）トイ・テキスト－単純なテキスト標記、

３）ＣＧＭグラフィックス－単純な図形表記、

４）ＩＰＤ－ＳＩ画像処理記述、

５）ＭＲファイル－複数の画像表現の記述を含むファイル、

を有する。

【００２６】

ＳＩテクノロジの重要な特徴は、ＳＩＤ内に画像処理動作（ＩＰＯ）を記憶する能力である。オブジェクトデータタイプのＩＰＤ（画像処理記述）は以下により詳細に説明するように、ＩＰＯを記憶するためのメカニズムを提供する。

【００２７】

単一のＳＩオブジェクトは一つより多くのデータタイプ及び／又はデータファイルを表現し得る。例えば（図３を参照されたい）、単一ＳＩオブジェクトは様々な解像度で記憶されたいくつかのＴｉＦＦファイルを含むディレクトリを参照してもよい。各々のＴｉＦＦファイルは同じ画像を表現するが、種々の絶対画素寸法及び照度を有する。このコンセプトは複数表現（ＭＲ）として知られている。ＳＩはダイナミックに決定される複数表現を指定するが、ＭＲファイルフォーマットは（以下に記述されるように）複数表現の明示的仕様を必要とするアプリケーションに対して定義される。

【００２８】

ＳＩ内の最上位のオブジェクト、つまり親を全く持たないオブジェクトはルート（根）オブジェクトと呼ばれる。これは以下のオブジェクト・レンダリング処理の説明において、より詳細に記述される。

【００２９】

ペーストボードはＳＩオブジェクトがレンダリングされる「フレーム」である。いくつかの属性はペーストボード内で指定され得る（図３参照）。１）相対サイズ（ＲｅｌａｔｉｖｅＳｉｚｅ）、２）角度、３）あてはめモード（ＦｉｔＭｏｄｅ　）、４）位置調整、及び５）制御点（ＣｏｎｔｒｏｌＰｏｉｎｔ）。

【００３０】

ペーストボードは、親オブジェクトに相対してレンダリングされたオブジェクトのサイズを指定する。例えば、サブオブジェクトに用いられたペーストボードは親オブジェクトに相対して（１．０、０．５）のようなサイズを指定してもよい。ここで、（１．０、０．５）＝幅の比率と高さの比率である。レンダリングの時、親が（１０００×２０００）画素でレンダリングするように指定された場合、サブオブジェクトは、ペーストボードの寸法がそのサブオブジェクトに全幅であるがその半分の高さでレンダリングするように指定したので、（１０００×１０００）画素でレンダリングされる。デフォルト（省略時）相対サイズは（１．０、１．０）である。

【００３１】

ＳＩをレンダリングする時、ペーストボードの相対サイズは、親オブジェクトの絶対サイズに基づいてペーストボードの絶対サイズを決定するために使用される（図４参照）。しかしながら、最適な品質のレンダリングの理由のためには、ペーストボードと親サイズによって決定される寸法とは異なる絶対寸法へ子オブジェクトをレンダリングさせるのが好ましいこともある。例えば、子オブジェクトを、１０対１（幅対高さ）のアスペクト比を

有する走査されたテキストのＴｉＦＦファイルであるとみなす。対応するペーストボードが（０．６７、０．３３）の相対サイズを有するとみなす。親オブジェクトを（４５０×３００）画素の絶対サイズへレンダリングさせる。それゆえ、子ペーストボードは（３００×１００）の絶対サイズを有する。ＴｉＦＦファイルをこのウィンドウへあてはめるようにスケーリングされた場合、このファイルは、ｙ方向に３倍、引き延ばされ、これによりテキストが大きく歪むことになる。この場合、オリジナルのアスペクト比でテキストをレンダリングして、所望のペーストボードへあてはめるのが好ましく、従って、このＴｉＦＦファイルは寸法（３００×３０）にスケーリングされ、１０対１のアスペクト比（幅対高さ）を維持する。

【００３２】

このタイプのレンダリングを支援するため、ＳＩは、ペーストボードによって定義される寸法において、ラスタがいかにしてレンダリングされるかを定義するために６個のＦｉｔＭｏｄｅ（あてはめモード）属性（図５）を定義する。これらのＦｉｔＭｏｄｅは図５において定義される。ＦｉｔＡｌｌ（フィットオール）を除く全てのモードがオブジェクトのアスペクト比を保持する。しかしながら、実際にレンダリングされたラスタ画像はペーストボードのサイズによって常に制限される。従って、レンダリングされるラスタの実際の寸法は、ペーストボードのサイズと、指定されたＦｉｔＭｏｄｅによりレンダリングされたオブジェクトのサイズの交点によって定義される。図５において、ＦｉｔＯｕｔｓｉｄｅ（フィットアウトサイド）モードに対してレンダリングされたオブジェクトは、ペーストモードよりかなり大きいが、実際のレンダリングされたラスタはペーストボードによって制限される。ＦｉｔＩｎｓｉｄｅ（フィットインサイド）モードに対して、オブジェクトはアスペクト比を維持しながら、ペーストボードにあてはまるようにレンダリングされる。従って、オブジェクトのアスペクト比がペーストボードのアスペクト比に一致しない場合、実際のレンダリングされたラスタはペースト寸法より小さいこともある。デフォルトＦｉｔＭｏｄｅ（あてはめモード）はＦｉｔＩｎｓｉｄｅ（フィットインサイド）である。

【００３３】

ＳＩオブジェクトは、いくつかのアプリケーションに対して、上記に定義されたＦｉｔＭｏｄｅ（あてはめモード）を用いてレンダリングされ得るが、しかしながら、ペーストボードのサイズに相対するレンダリングされたラスタの位置に対して制御を提供することが望ましい。位置調整属性は、図７に示す例を用いて図６に示すように、９個の使用可能な位置の内の一つにおいて、レンダリングされたラスタがペーストボード内の何処に位置するかを指定するのに使用される。デフォルト位置調整は中心（ｃｅｎｔｅｒ）である。位置調整属性は、ＦｉｔＭｏｄｅ（あてはめモード）がＦｉｔＡｌｌ（フィットオール）の場合は無視される。

【００３４】

各オブジェクトはレンダリングされたオブジェクトを回転するために使用される角度（Ａｎｇｌｅ）属性を有してもよい。その効果は、オブジェクトを指定されたペーストボードへレンダリングし、次いで指定された角度の分、ペーストボードを制御点（以下を参照されたい）の回りに回転させることである（図８）。指定された角度でオブジェクトのレンダリングを適正に処理することが、使用されているオブジェクト・レンダリング・プログラムの役割である。あるレンダリングプログラムは適切な角度でオブジェクトを最適にレンダリングすることが可能であるが、他のレンダラーは、最初のオブジェクト・レンダリングの後でラスタ回転プログラムを呼び出してもよい。この選択は処理系依存型である。レンダリングされたラスタの実際の寸法は、図８の（ｂ）に示すように、回転され且つレンダリングされたオブジェクトの境界ボックスとなる。デフォルト回転角度は０．０度である。

【００３５】

一組の子オブジェクトから成る親オブジェクトの場合、各々の子はレンダリングされ且つＭｅｒｇｅＰｏｉｎｔ（マージポイント）（併合点）として知られる位置で親のペースト

ボードへ併合される。併合点は親オブジェクトのペーストボードに相対して指定される。ＣｏｎｔｒｏｌＰｏｉｎｔ（コントロールポイント）（制御点）は子のペーストボードに相対して対応する点である。子の制御点は親の併合点と整合しようとする。例えば、（０．５、０．５）の制御点は、結果的に子を併合点の中心へ位置させる。このデフォルト制御点値は（０．５、０．５）である。

【００３６】

（併合点と共に）制御点はレンダリングされた子オブジェクトが親オブジェクトのペーストボードの何処へ併合されるかを制御する。ＳＩＤ内では、制御点が指定され且つペーストボードが子のペーストボードに相対する座標内の子オブジェクトと対応する。しかしながら、レンダリングされた子オブジェクトは、必ずしもＦｉｔＭｏｄｅ　（あてはめモード）とＡｎｇｌｅＭｏｄｅ　（角度モード）に依存して指定されたペーストボードを塗りつぶさない。従って、各オブジェクトに対するレンダリングプログラムはレンダリングされた子に対して同一の相対位置を維持するために制御点を自動的に調整しなければならない。

【００３７】

図９に示した実施例を考察されたい。制御点は最初は子の中心（０．５、０．５）にある。ＦｉｔＡｓｉｓ　モードで、子をＴｏｐＲｉｇｈｔ（右上）モードへ調整してレンダリングした後、制御点は絶対値（－１５、３０）へ自動的に調整される。これによって、併合動作はレンダリングされたラスタを予想の位置へ位置させる。この実施例を理解するため、子はサイズ（０．３、０．６）を有するようにＳＩＤ内で指定される。レンダリングは親が（２００×１００）であることを示し、それゆえ、必要とされる子のサイズは６０×６０である。しかしながら、ＦｉｔＭｏｄｅ　（あてはめモード）はＦｉｔＡｓｉｓ（そのままあてはめる）を示す。実際の子は１５×４５で記憶され、そこでレンダリングされたラスタは１５×４５となる。調整はＴｏｐＲｉｇｈｔ（右上）を示し、そこで１５×４５のラスタは要求されたサイズのＴｏｐＲｉｇｈｔ（右上）６０×６０に位置し、これによってレンダリングされたラスタの原点は要求されたラスタに相対して（４５、０）となる。経路上をフローするラスタサイズがレンダリングされた子のサイズであり且つ要求された子の６０×６０のサイズではないので、制御点は実際のレンダリングされた子に相対する移動（シフト）を反映するために調整されなければならない。制御点は要求された子のサイズに相対して（３０、３０）へマッピングされる。レンダリングされた子の原点が（４５、０）であるので、制御点値がレンダリングされた子に相対して（３０－４５、３０－０）＝（－１５、３０）であることが容易に理解できる（図９参照）。

【００３８】

制御点の情報はレンダリングされたラスタ画像のヘッダの中へ記憶される。このように、制御点情報はその知識を要求するあらゆるＩＰＯ（画像処理動作）にとって使用可能である。特に、ＳｐａｔｉａｌＴｒａｎｓｆｏｒｍ（空間的変換）ＩＰＯは下記の例において示すように、指定された変換に従って制御点を適切に調整しなければならない。

【００３９】

ＲｅａｄＯｂｊｅｃｔ（）（読み取りオブジェクト）はＳＩオブジェクトを他の処理に適するラスタ画像へ変換する手順を表現する。ＲｅａｄＯｂｊｅｃｔ（）は、パラメータとしてペーストボード、ＳＩオブジェクト（作成されたもの又は走査されたもの）、及び親オブジェクトの絶対サイズを用いる。ＲｏｏｔＯｂｊｅｃｔ（）（ルートオブジェクト）の場合、指定されたサイズは所望される出力画像の寸法であり且つペーストボードの相対サイズのフィールドは一般的に空であり、且つデフォルト値（１．０、１．０）を示唆する。他の情報がオブジェクト・レンダリングの品質改良のために使用され得る場合、そのような情報がＲｅａｄＯｂｊｅｃｔ（）（読み取りオブジェクト）へ渡されてもよい。

【００４０】

ＲｅａｄＯｂｊｅｃｔ（）は適切なラスタ画像をレンダリングするために下記のいくつかの動作を実行する。

１）絶対ペーストボードサイズを決定し、２）オブジェクトによって参照される全ての表

現を決定し、３）使用可能な情報に基づいて、レンダリングのために適切な表現を選択し、４）適切なレンダリング手順を呼び出すことによってＦｉｔＭｏｄｅ　、位置調整、及び（可能ならばＡｎｇｌｅ　（角度））によってデータをラスタへレンダリングし、５）（ステップ４）で実行されない場合、）レンダリングされたラスタをＡｎｇｌｅ　属性によって指定された角度まで回転し、且つ６）制御点位置を調整する。ＲｅａｄＯｂｊｅｃｔ（）手順の出力はレンダリングされたオブジェクトのラスタ画像である。制御点はラスタ画像の座標系におけるラスタ画像のヘッダ内で指定される。

【００４１】

ＲｅａｄＯｂｊｅｃｔ（）は、ペーストボード内で指定された相対サイズと親寸法と乗算して絶対ペーストサイズを決定する。ペーストボードが指定されない場合、（１、１）の相対サイズが割り当てられる。例えば、指定されたサイズが（１０００×２０００）、相対サイズが（１．０、０．５）だった場合、ペーストボードの絶対サイズは（１０００×１０００）である。実際にレンダリングされたオブジェクトは、上記に説明されているように、ＦｉｔＭｏｄｅ　（あてはめモード）とＡｎｇｌｅ　（角度）属性によりこのサイズと違うこともある。

【００４２】

ＳＩオブジェクトはラスタ画像へレンダリングすることができるデータファイル又は表示への基準として動作する。この表現はラスタ、テキスト及び／又はベクトル図形のようないくつかのタイプの内の一つであってもよい。さらに、あるオブジェクトはいくつかの表現を参照してもよいが、これらの表現の各々は同一画像を表現するが、記憶フォーマット、サイズ及び／又は解像度の点で異なる。ＲｅａｄＯｂｊｅｃｔ（）が呼び出された時、このＲｅａｄＯｂｊｅｃｔ（）は、Ｆｉｎｄｅｒ（）手順を呼び出すことによってレンダリングするためにどの表現が「最良（ベスト）」表現であるかを決定しなければならない。Ｆｉｎｄｅｒ（）はオブジェクトによって参照された全ての表現を位置づけ且つどの表現がレンダリングに最適かを選択する。

【００４３】

一旦表現が選択されると、この表現はラスタ画像へレンダリングされなければならない。各々の支援されているデータタイプはこの表現をラスタ画像へ変換するためにそれと対応するレンダリング手順を有さなくてはならない。これらのレンダリング手順はＦｉｔＭｏｄｅ　とＡｎｇｌｅ　属性を支援しなければならない。好ましくは以下のデータタイプが支援される。

データタイプ	レンダラー	例
ラスタ	レンダ＿ラスタ () (render＿raster())	TiFF,RES,PCX
シンプルテキスト	レンダ＿テキスト() (render　text())	テキストの１行、 １フォント
シンプルグラフィックス	レンダ＿グラフィックス() (render＿graphics())	ＣＧＭ
パターン	レンダ＿パターン() (render＿pattern())	一定のカラー
ＩＰＤ	レンダ＿ｉｐｄ () (render＿ipd())	

【００４４】

Ｒｅｎｄｅｒ（）（レンダリング）手順は回転角度をオブジェクト・レンダリングへ直接組み込むか又はラスタ回転動作をラスタ化されたオブジェクトへ用いてもよい。例えば、

使用可能なテキスト・レンダリング・プログラムは指定された角度でテキストを直接レンダリングする能力を有し、これによりラスタ回転が続くレンダリングより良好な品質が得られる。

【００４５】

上記に説明したように、制御点は、ＦｉｔＭｏｄｅ（あてはめモード）や位置調整やＡｎｇｌｅ（角度）に依存してレンダリングされたオブジェクトと併合点の同一相対位置を保持するために調整されなければならない。ＲｅａｄＯｂｊｅｃｔ（）の主な役割は、この調整を行い、且つ調整された制御点をラスタ画像ヘッダへ挿入し、それをさらに（空間的変換などのように）調整することができる、あらゆるＩＰＯもまた、この調整された制御点を利用できるようにすることである。

【００４６】

ＲｅａｄＯｂｊｅｃｔ（）手順はＦｉｎｄｅｒ（）関数（機能）を呼び出して出力ラスタ画像へレンダリングするためにオブジェクトデータを検出し且つリターンする。Ｆｉｎｄｅｒ（）手順は、ＳＩのレンダリングに応じて６個の動作を実行する：１）オブジェクトの全ての表現を検出し、２）各表現（ＴｉＦＦ、テキスト、グラフィックス）のタイプを識別し、３）最良の表現を選択するために必要な各表現の属性を決定し、４）各々の利用可能な表現へメリットとなる機能を適用し、且つ最高のメリット値を有する表現を選択し、５）ポインタを表現データへリターンし、且つ６）表現のデータタイプをリターンする。Ｆｉｎｄｅｒ（）メカニズムは特定の処理系に必然的に依存する傾向がある。しかしながら、機能性についてのいくつかのステートメントが与えられ得る。

【００４７】

構造化画像定義言語（ＳＩＤＬ）の現在シンタックスにおいて、ＳＩオブジェクトは、三つの方法の内の一つにおいて記述され得る（以下のソースコードは、最初に述べたように、著作権保護を受けており且つ前述のハードウェアシステムで実行され得る）。

```
(1) Object (オブジェクト) = {
          External (外部) = {
            System (システム) ="UNIX (ユニックス) (登録商標) "
               Name (ネーム) ="bird.TiF";
               } ;
                  } ;
(2) Object (オブジェクト) = {
       internal (内部) = {
          Bytecount (バイト計数) =１０２４
          Data (データ) = {
               ...
               } ;
            } ;
               } ;
(3) Object (オブジェクト) = {
       IPD = {
          AspectRatio (アスペクト比) =1．5；
          DefaultWidth (省略値幅)  =4インチ
               ...
               } ;
               } ;
```

【００４８】
Ｅｘｔｅｒｎａｌ（外部）として定義されたオブジェクトは、オブジェクトによって参照されたデータが現在ＳＩＤ内に記憶されていないオブジェクトである。一般に、実際のデータは別個のファイル内に記憶される。システムフィールドは、Ｆｉｎｄｅｒ（）がデータをどのように検出するかを理解するために使用する記述である。ネームエントリはＦｉｎｄｅｒ（）が所望されるデータを実際に検出するために使用する一組のキーワードである。上記の例においては、Ｆｉｎｄｅｒ（）は、「ＵＮＩＸ（ユニックス）（登録商標）」のシステムエントリが実際のデータが”ｂｉｒｄ．ＴｉＦ”のネームエントリを有するファイル用の適切なユニックス（登録商標）ファイルシステム経路を捜すことによって見つかることを示すことを理解するために実行される。システムエントリは”ＳＹＢＡＳＥ（登録商標）” であった場合、Ｆｉｎｄｅｒ（）はデータ用の”ＳＹＢＡＳＥ（登録商標）” データベースを検索するためのキーワードとしてネームフィールド内のキーワードを取り扱う。

【００４９】
インターナル（内部）として定義されたオブジェクトはＳＩＤ内に直接記憶されたデータを含む。これは、ラスタアイコン又はシンプルテキストファイルのような小さなオブジェクトをＳＩＤ内に直接記憶するのに有用であり、それ故外部記憶装置を必要としない。

【００５０】
ＩＰＤは、構造化画像（ＳＩ）を他のＳＩオブジェクトとＩＰＯから構成されるのを可能

にするＳＩオブジェクトデータタイプである。ＳＩオブジェクト内に記憶される情報に基づいて、Ｆｉｎｄｅｒ（）は、全ての利用可能な表現を検出し、且つデータタイプと、レンダリングのために「最良」の表現を選択するために必要とされるあらゆる属性を決定しなければならない。これらの機能を実行するために使用されるメカニズム及び／又はプロトコルはＳＩ内では定義されないが、このようなメカニズムが存在し且つ本明細書中に使用可能であることが理解される。

【００５１】

全ての表現が一旦決定されると、Ｆｉｎｄｅｒ（）は、出力ラスタ画像へレンダリングするために最適な表現を選択しなければならない。このメカニズムはメリット（優先的）機能計算を各表現に適用することによって実行される。正確なメリット機能は処理系依存であり且つＳＩ内で定義されない。しかしながら、これらの表現の属性、所望される出力画像属性、目標の出力デバイス（例：白黒のプリンタ、カラーディスプレイ）、及び処理の複雑性に基づくべきである。ある表現が一旦選択されると、Ｆｉｎｄｅｒ（）は、処理を開始するために、ＲｅａｄＯｂｊｅｃｔ（）への実際データとデータタイプへポインタをリターンする。

【００５２】

上記のように、ＩＰＤは、構造化画像（ＳＩ）を他のＳＩオブジェクトとＩＰＯから構成されるのを可能にするＳＩオブジェクトデータタイプである。ＩＰＤは、構成要素オブジェクトとこれらの構成要素オブジェクトに用いられるＩＰＯに対する参照の構造化記述を表現する。図１０はＩＰＤのグラフィックな表現である。

【００５３】

ＩＰＤは下記の記述子のための基準を備える。１）デフォルトサイズ、２）ＲｅａｄＯｂｊｅｃｔ、３）経路、４）ＩＰＯ、５）Ｍｅｒｇｅ　（併合）、６）ペーストボード、７）イニシャライザ、及び８）構成要素ペーストボード。図１０において、経路と呼ばれる各垂直なラインがＩＰＯの集合である。ＲｅａｄＯｂｊｅｃｔは、ＳＩオブジェクトを、経路内のＩＰＯによって処理するのに好適なラスタ画像へ変換する手順を表現する。レンダリングされたオブジェクトは経路を介してフローし且つ各ＩＰＯによってシーケンシャルに処理される。各経路は、ＳＩオブジェクトがレンダリングされる「フレーム」を定義するため、その経路の内部に、ＲｅａｄＯｂｊｅｃｔによって使用されるペーストボードを記憶する。この経路ボードは構成要素ペーストボードと呼ばれる。水平ラインはＩＰＤオブジェクトのペーストボードであり且つ「ブランク」のラスタ画像を表現し、その画像上に構成要素のペーストボードが併合される。黒の三角形は併合動作を表現し、この併合動作において、構成要素のペーストボードがオブジェクトのペーストボードに組み合わせられる。構成要素の併合順序は明示的であり、従って「上部」及び「下部」の画像のセンスが支援される。図１１はＳＩのレンダリングの間のラスタデータのフロー（即ちＲｅａｄＯｂｊｅｃｔからＭｅｒｇｅ　までの制御フロー）を描写する。

【００５４】

各ＩＰＤはＩＰＤ用のデフォルトサイズを指定する情報を含まなければならない。この情報は、画像がレンダリングされるように意図されたサイズについてのレンダリング・プログラムに対するヒントとして使用される。現在のＳＩＤＬ（構造化画像定義言語）シンタックスにおいて、この情報はアスペクト比及びデフォルトの幅ストリングとして好ましくは記憶される。例えば、

```
ＩＰＤ＝｛
    AspectRatio （アスペクト比）＝１．５
    DefaultWidth（デフォルト幅）＝４インチ
    ...
    ｝；
    ｝；
```

【００５５】
ＡｓｐｅｃｔＲａｔｉｏ （アスペクト比）フィールドは幅に対する高さの比率によって定義される。ＤｅｆａｕｌｔＷｉｄｔｈ（デフォルト幅）は幅とユニット（単位）を定義するストリングである。使用可能な単位はｉｎ（インチ）、ｃｍ、ｍｍ、ｍ、ｐｔ（点）、及びｐｉｘ（画素）である。
【００５６】
ＩＰＯは画像処理コマンドのデバイス独立記述である。一つだけのＩＰＯが現在ＳＩ内で指定されている。それは下記の例においてより詳細に説明される空間的変換ＩＰＯである。しかしながら、追加のＩＰＯが処理系依存であることが理解されよう。これらの中で、いくつかの基本的な機能が不可欠と考えられ且つ以下のようなＩＰＯが好ましいとされる。
１）トランスフォーマ（変換器）：画像のサイズ又は形状を空間的に変換する（例：アフィン変換、ワープ変換）。
２）フォーマッタ：一つのフォーマットから他のフォーマットへ画素を変換する（例：コンバート、インターリーブ、文字のｃ間隔）。
３）フィルタ：種々のフィルタリング動作を適用する（例：コンボルブ（重畳）、中央値、形態学）。
４）カラーライザ：画像のカラー値を調整する（例：相対、絶対、コントラスト、白の点）。
５）マスク：マスキング動作を適用する（例：加算、減算、変換、設定）。
６）その他：他の雑動作を適用する。
【００５７】
ＩＰＯはその動作の選択領域を指定できる。選択は画像内のどの画素が特定の動作において処理されるべきかについての記述である。例えば、選択はビットマップ又は領域の数学的記述を含んでいてもよい。通常、選択はその性質はバイナリである。画素が処理されたり又は処理されなかったりする。しかしながら、「グレー（中間色）」の選択値によって、出力画素値はオリジナルの画素値と処理される画素値の間でスケーリングされる。
【００５８】
図１２に示されているように、ＩＰＤ内の全ての子オブジェクトは（オブジェクトペーストボードに相対する）併合点と（子ペーストボードに相対する）制御点によって指定される位置で親ペーストボードへ併合される。マージ（併合）順はＩＰＤにおいて明示的である。
【００５９】
通常、ＩＰＤオブジェクトのペーストボードは、構成要素ペーストボードが併合される「ブランク（空白）」ラスタを表現する。イニシャライザを用いて、カラー、パターン、又はラスタ画像によってもペーストボードを初期化するとが可能である（図１３参照）。イニシャライザは実際にはＲｅａｄＯｂｊｅｃｔ（）とＩＰＯを含む経路である。制限は相対サイズが（１．０、１．０）であり且つ空間的変換ＩＰＯがこの経路内に存在していないことである。ペーストボードはレンダリングされたイニシャライザのサイズとなる。一般的に、イニシャライザは、ペーストボードを着色するか又は背景のラスタ画像によってペーストボードを初期化するために使用される。

【００６０】
さらに、本発明は、定義されていないと識別されたＳＩＤ内でフィールドを有する構造化された画像が定義されるのを可能にする。このようなフィールドは、このＳＩに対してダイナミックに結合されるデータに対するプレイス（位置）ホルダであり且つランタイム（レンダリングタイム）指定である。定義されないフィールドに対して二つの分類がある。１）結合されないＳＩ、及び２）不完全ＳＩ。
【００６１】
概して、ＳＩの構成要素オブジェクトは、ＳＩＤ自体の内部には含まれないが、（Ｅｘｔｅｒｎａｌ（外部）オブジェクトのＳｙｓｔｅｍ（システム）及びＮａｍｅ（ネーム）フィールドなどの）ＳＩオブジェクトのフィールドによって参照される。これはデータの構造化画像へのダイナミックな結合を可能にする。
【００６２】
例えば、ユーザが印刷用にＳＩを作成しているインタラクティブ編集セッションについて考察されたい。このＳＩは一回だけ使用され、その後は廃棄されるべきである。ユーザはインタラクティブＳＩエディタを用いてＳＩを作成する（これはＳＩを作成するための例示的な方法にすぎず、適切な走査ソフトウェアを用いたテンプレート又は自動作成による作成より必ずしも好ましいというわけではない）。ＳＩの全ての構成要素は公知であり且つ定義されている。一旦完了すると、ユーザはＳＩをプリントし且つジョブが完了する。このＳＩを記述するために生成されたＳＩＤが完全に結合しているのは知られており、且つこれは全ての構成要素がレンダリングの前に定義付けられたことを意味する。要するに、「完全結合」とは、出力ＳＩラスタのレンダリングの前に、ＩＰＤ内の全てのオブジェクトが、定義されたエンティテイを指し示すことを意味する。
【００６３】
しかしながら、多くのアプリケーションにおいて、完全に結合した画像はジョブに対しては適切ではない。例えば、画像の内の一つの構成要素が地理的販売立地条件と共に変わるようなブローシャを出す広告主を考察されたい。この構成要素の全ての可能性のある画像はデータベース内に記憶される。この広告主は例えばＳＹＢＡＳＥ（登録商標）のようなデータベースシステムを識別するためにＥｘｔｅｒｎａｌ（外部）オブジェクトのＳｙｓｔｅｍ（システム）エントリを定義することは可能だが、Ｎａｍｅ（ネーム）フィールドは定義されていないとマークされる。ランタイムにおいて、正確なキーワードはそれぞれの要求された出力画像の生成のためにそのオブジェクトとダイナミックに結合され得る。この方法を用いて、単一ＳＩは多くの種々の出力画像を生成するのに使用され得る。結合されていない（非結合）画像は有効なＳＩであるが、これらの画像はレンダリングされるランタイムにおいて追加の情報を必要とする。
【００６４】
理解されるように、ＳＩは結合又は非結合画像に対して好みはないが、選択はアプリケーション依存である（非結合構造化画像を用いた代表的なユーザディスプレイに関しては図１４を参照されたい）。
【００６５】
非結合画像の概念と同様に、本発明はＩＰＯ（画像処理動作）のパラメータが定義されないのを可能にする。また、ＩＰＯ自体は定義されなくてもよい。例えば、車の色がプリント同士の間で変化するような自動車の画像を含むブローシャを作成する広告主を考察されたい。広告主は全ての可能性のあるカラーごとに車の個々の画像を有し且つＳＩＤのダイナミック結合を使用することができる。広告主はまた、単一の自動車の画像を使用し且つダイナミックに結合されるべきレンダリング手順に対する引数として所望されるカラーをＩＰＯへ渡すことを選択してもよい。カラー値がランタイムに提供されていない場合、出力ラスタのレンダリングが実行されることは不可能となる。これは「不完全」ＳＩとして知られる（図１５参照）。不完全なＳＩは有効ＳＩであるが、これらの画像は、追加の情報がなければレンダリングできない。
【００６６】

非結合ＳＩをレンダリングする時に明確な二つの重要な考察が得られる。１）結合メカニズム、及び２）レンダリング・モデル。

【００６７】

非結合ＳＩがダイナミックに結合されるメカニズムは処理系依存である。四つの可能なメカニズムは本明細書中で識別されるが、しかしながら、以下の他のメカニズムも可能であり得る。

１）ＳＩが「完全画像」を常に表現することを要求する。ＳＩが非結合及び／又は不完全である場合、ＳＩＤを変更するためにプリ（事前）フィルタリング動作を実行する。

２）現在参照（基準）結合メカニズムを使用する。ＳＩＤファイルは、ＳＩＤＬ内の定義されない参照に対応する「タグ」定義として他のファイル内に記述されるように、定義されない参照及び結合情報を含むことができる。結合は、例えば、ＳＩＤファイルと結合情報を連結することによって実現し、これにより完全なＳＩＤを作成する。

３）結合情報をＳＩレンダラーへのパラメータとして指定する。

４）ＳＩＤ内のあらゆる非結合及び／又は不完全なフィールドを満たすため、自動的に呼び出されるＦｉｎｄｅｒ（）手順に対するダイナミック結合方法を含む。

【００６８】

本発明の目的のために上記の結合メカニズムの全てが好ましいとされるが、しかしながら、種々のメカニズムにおける選択が処理系依存であることが理解される。

【００６９】

いくつかのアプリケーションに関しては、ＳＩはレンダリングタイムに完全結合し得ないフィールドを有することもある。例えば、各ページが、レンダリングタイムに結合されようとする６個の同一子オブジェクトを有するＳＩである文書、例えば、各子オブジェクトが売出し中の家に対応している不動産会社のリストなどについて考察されたい。文書が合計９件の家をディスプレイしようとする場合を考察されたい。これは二つの文書ページを必要とするが、子オブジェクトの内の三つのオブジェクトだけが第２のページ上に必要とされる。ＳＩレンダラーは、他のモデルにおいても動作可能であるけれど、以下の三つのモデルにおいて動作することができる。１）非結合エラーによる故障、２）非結合オブジェクトを無視して、不完全ＩＰＯを非動作として取り扱う、及び３）非結合オブジェクトのペーストボードのみをレンダリングして、不完全ＩＰＯを非動作として取り扱う。

【００７０】

全ての三つのモデルが本発明に使用するのに好ましいとされるが、しかしながら、この選択はアプリケーション依存であることが評価されよう。

【００７１】

レンダリングはＳＩＤＬ（構造化画像定義言語）において特定的に定義されないけれど、明確にするために簡単な説明が本明細書中において提供される。実際のレンダリング処理において、いくつかのレンダリング方法が目的、優先順位、資源、及びその他のような条件によって選択され得る。レンダリングは実行の特定的な問題であるが、しかしながら、ＳＩの全ての実行が所与のＳＩ及び出力デバイスに等価な出力を生成しなければならないことが理解されよう。画像処理プログラムにおける希少な差異が異なる結果を生じるので、これらの結果は正確には一致しないと予想される。しかしながら、出力はリーズナブルであるべきである。このステートメントは、種々のＣコンパイラは同一マシンコードを発生する必要はないが、すべてのコンパイラは作動マシンコードを発生しなければならないことに例えられる。

【００７２】

ＳＩのレンダリングに対して二つのアプローチが検査された。１）オブジェクト指向レンダリングと２）処理ツリー・レンダリング。

【００７３】

オブジェクト指向レンダリング（図１６参照）はオブジェクトツリーを直接上下解析し且つユーザが進む各オブジェクト（ＳＩ）をレンダリングする方法である。ＳＩがメッセージ「レンダリングせよ」を受け取る時、ＳＩは所与のサイズ及び解像度特性又は寸法属性

によってレンダリングされる。ＳＩの構築のために他の構成要素が必要とされた場合、オブジェクトは構成要素オブジェクトへメッセージ「レンダリングせよ」を送り且つ出力画像を取る。オブジェクト指向レンダリングは、オブジェクト・ツリーの「変換処理」として最高と見なされ得るオブジェクト・ツリーの直接的実行である。オブジェクト指向レンダリングはインタラクティブ編集状況におけるレンダリングに特に適している。

【００７４】

処理ツリーは、デバイス依存であり且つ解像度依存である画像処理コマンドのＤＡＧ（閉路を含まない有向グラフ）である（図１７参照）。プロセシング・ツリーは全ての入れ子構造のＳＩに対応するＳＩＤを単一のＳＩへ拡大し、次いで全ての座標及び寸法を絶対デバイス依存値へ変換するためにデバイス依存情報を用いることによって発生する。表現エントリが指し示すデータ内容はラスタへレンダリングされる。処理ツリーがデバイス依存であり且つ解像度依存であるので、ＩＰＯは当業者によって理解される様々な最適化技術によってその処理速度が最適化され得る。例えば、ユーザは二つのスケーリングと一つの回転動作を結合して一つのアフィン動作にしてもよく、これによって適合化連鎖動作をアフィン動作に結合する。要するに、三つの分離した動作を一つに結合するのは処理速度を改良するためである。

【００７５】

出力画像は、得られる処理ツリーの単純横断によってレンダリングされる。これはオブジェクトツリーの「コンパイレーション（編集）」であると考えられる。処理ツリー・レンダリングは文書印刷においては一般的な非常に大きな出力画像のレンダリングに特に適している。この画像処理最適化はレンダリングに必要とされる時間を相当減少することができる。

【００７６】

図１８に示すＳＩについて以下に簡単に説明する。この記述的な例は上記のようにＳＩＤＬを用いて書き込まれる。

この複合画像におけるトップレベルのＳＩは四つの構成要素を備える。

１）マウンテン画像－ラスタ画像オブジェクト（マウント１）

２）「サンプル．．．」－テキストオブジェクト（テキスト１）

３）木と家－ＩＰＤ（ホーム１）、及び

４）バード（鳥）画像－ビットマップラスタオブジェクト（バード１）。

第２レベルのＳＩは四つの構成要素を備える。

１）ツリーグラフィックス－グラフィックスオブジェクト（ツリー２）、及び

２）ハウス画像－ラスタオブジェクト（ホーム２）

【００７７】

トップレベルＳＩに対応するＳＩＤファイルは、図１９乃至図２１に示されている。ＳＩはＳＩＤＬにおける一組の「タグ」として記述され（ここでまた、以下のソースコードは著作権保護を受けており且つ前述のハードウェアシステム上で実行され得る）、これらの「タグ」は、タグタイプのネームと、”｛”及び”｝”によって囲まれたタグ本体によって指定される。タグは”：ネーム”シンタックスを用いてネーム付けされる。タグネームは”＄”を接頭辞として置いて他のタグに対する参照として用いられ得る。オブジェクトタグはＳＩのトップレベルタグである。この例において、オブジェクトタグはＩＰＤオブジェクトを表現し且つその記述は参照されたタグ”ｉｐｄ１”である。ＩＰＤタグ”ｉｐｄ１”は、ＡｓｐｅｃｔＲａｔｉｏ　（アスペクト比）とＤｅｆａｕｌｔＷｉｄｔｈ（デフォルト幅）、ペーストボード属性ＦｉｔＭｏｄｅ　（あてはめモード）、位置調整と角度、及びシーケンスがｍｅｒｇｅ１（マージ１）からｍｅｒｇｅ４（マージ４）までの併合順序を指定する四つのｍｅｒｇｅ　（併合）タグを定義する。各ｍｅｒｇｅ　タグは、併合点タグによってペーストポイントに相対すう併合点を指定し、且つ経路タグによって画像処理シーケンスを指定する。経路タグは、他のオブジェクトタグによって子オブジェクトを、サイズタグによって相対サイズを、制御点タグによって（子に相対する）制御点の位置を、且つＩＰＯタグのリストによって画像処理動作を、表記する。経路１、経路２

、及び経路４が全て、ラスタ、テキスト、又はグラフィックファイルを指示することに注目されたい。経路３は、グラフィカル・ツリーと家のラスタ画像を結合してラスタ画像を生成する他のＳＩＤファイルを指示する。

【００７８】

いくつかのアプリケーションに対しては使用可能な表現の明示的な記述が好まれる。このために、ＭＲ（複数表現）ファイルフォーマットが定義される。このファイルフォーマットはＳＩＤＬに類似したシンタックスを現在使用している。図２２及び図２３は、ＭＲファイルフォーマットの例である。各表現は、ファイルタイプを識別するフォーマットフィールド、及びデータを明示的に定義するデータフィールドと共にリストされる。ＴｉＦＦファイルのようなファイルベースのデータに対しては、データフィールドはファイルへのＵＮＩＸ（登録商標）経路を有する。単純なテキストに関しては、データフィールドはテキストストリングを有する。ＭＲフォーマットは、各表現が、データを記述する明示的に定義された一組の属性を有するのを可能にする。ラスタデータに関して、属性は、寸法、測光、又は他の属性を含んでいてもよい。テキストオブジェクトに関して、属性は、フォント、スタイルその他を含んでいてもよい。しかしながら、ＭＲファイルフォーマットの使用は本発明においては好ましいとされない。

【００７９】

ＳＩフォーマット内で指定された唯一のＩＰＯは空間的変換ＩＰＯであり、このＳｐａｔｉａｌＴｒａｎｓｆｏｒｍ（空間的変換）ＩＰＯはラスタ画像コンテントで線形の空間的変換を実行するために作用する。これらの例は回転、スケーリング、及び傾斜を有する。概して、空間的変換ＩＰＯは、（併合点と制御点属性を介して制御される）変換を除く全てのアフィン変換を実行する。

【００８０】

空間的変換のパラメータは以下の通りである。

パラメータ	値
スケール	ＸＹ－スケールに対して１値（保存、アスペクト比（a.r.）又は 任意のスケールに対して２値 （補間はアプリケーション依存、注記：多分引数）
回転	角度（度）
剪断	剪断角（度）
鏡像	"X"又は"Y"
変換点	３つの対のコーナー点
アフィン	アフィン係数
固定点	変換される子に相対する（x、y）－ デフォルト値は制御点位置。

【００８１】

ＦｉｘｅｄＰｏｉｎｔ（固定点）パラメータは参照（基準）点の回りで空間的変換の適用を可能にする。例えば、固定点を指定することによって、ユーザが、オブジェクトの中心の回りを回転する代わりに、固定点の回りにオブジェクトを回転するのを可能にする。概して、あらゆる支援された空間的変換が固定点の回りを動作するように定義され得る。固定点は空間的変換に対するパラメータであり且つペーストボードの部分ではないので、ＩＰＯは制御点を自動的に調整して、子オブジェクトが正確な位置で親ペーストボードへ併合されるのを確実とする。制御点がどのように調整されるかの例が以下に示されている。

【００８２】
例えば（図２４を参照されたい）、オブジェクトは（０．５、０．５）の制御点Ｃと子の上部左コーナＰに相対する（２．０、０．０）の固定点を有する。レンダリングが子がサイズ５０×５０であることを示すと想定されたい。固定点Ｆは子に相対して（－１００、０）である。制御点はＰに相対して最初は（２５、２５）であり且つ固定点に相対して（１２５、２５）である。子を固定点のまわりで４５度回転させると、固定点に相対する（７１、１０７）において原点を有する７１×７１（境界ボックス）の新サイズを有する子を発生する。オリジナルの制御点が固定点に相対して（１２５、２５）だったことを思い出されたい。従って、制御点は画素サイズにおいて（１２５、２５）－（７１、－１０７）＝（５４、１３２）即ち回転した子の原点Ｑに相対して（０．７６、１．８６）となるように調整される。従って、制御点は固定点のまわりでの回転によって適切な位置で子がペーストボードへ併合されるのを確実とするために調整される。
【００８３】
或いは、インタラクティブ・エディタは、ＳＩの形成の間、回転の中心その他を処理し且つＳＩＤを書き込む時に併合点を適切に単純に調整することができる。即ち、ユーザは、例えば、所望される位置で、回転の中心を設定することができ、ＳＩＤはもはや関係がなくなる。
【００８４】
重要なのは、ＳＩが単なるインタラクディブ・エディタではないことである。例えば、高校の卒業アルバム用プリンタは仕上げられたアルバムとしてプリント用に電子的に準備するために多数の個別の写真を有する。特に、写真は、（驚異的に時間がかかる）一枚一枚、又は一度に１２枚までスキャンできる技術を用いて、データベースへスキャン（走査）されなければならない。しかしながら、残念ながら、走査用に個々の写真を適切に並べるためにユーザがいかに懸命に努力しても、走査の結果として僅かなずれが必ず生じるのである。通常、各写真は（写真の）下へ印刷されるネームと結び付けるバーコード又はいくつかの他の識別機能を有している。
【００８５】
上記の走査状況下で、ＳＩを自動的に発生することが可能なソフトウェアが存在する。即ち、各個別の子オブジェクト（即ち個々の写真）を識別し且つあらゆる位置合わせ誤りを補正するために画像処理動作を含むＳＩが形成され、下方に適切なテキスト（参照バーコード）を有するページ上に各写真を適切に構成することができる。このようなソフトウェアは一度に多くのオブジェクトを走査することが可能であり且つ各子オブジェクトは印刷時の適切なレンダリングのために回転したり回転しなかったりする。これによって、ユーザの意のまま、卒業アルバム用プリンタはその特定のページへの配置が望まれる子オブジェクトのみを選び、所望通りにそれらを配列し、適切なテキストを付し、且つプリントすることができる。上記に説明したように、子オブジェクトは呼出し用バーコードによって参照される。
【００８６】
さらに、卒業アルバル用プリンタは、望まれれば、写真及びテキストのために「非結合」ホールを有するテンプレートを作成することにより他のＳＩを形成しようとする場合がある。つまり、卒業アルバル用プリンタは適当な数の子オブジェクトを印刷用のテンプレートに結合する。しかしながら、重要なのは、テンプレートもまたその中（背景など）に組み込まれた知的処理を有することができ、この知的処理はカラー、クロッピング、特殊効果、その他を自動的に変更する。
【００８７】
【発明の効果】
上記の発明の概念は、階層的及びデバイス独立フォーマット内のオブジェクトの集合としての複雑なカラーラスタ画像の表現を可能にする。出力ラスタへ貢献するオブジェクトは、テキスト、グラフィックス、他のラスタ、又は他のＳＩを源とし、且つ選択性及び変更のためにそれらのヘリテージ（譲り受ける権利）を維持する。ＳＩは、ソースデータのみ

ならず、それらをレンダリングするために必要とされる画像処理動作も含む。ＳＩ技術は、再編集可能性、カスタマイゼーション（個別化）及びエンハンスメント、自動画像アセンブリ、及び高性能画像形成（即ち平行、分散、キャッシング、その他）を支援する。これは、以降の編集や再編集のために画像を記述するために非常に強力なフレームワークを提供する。従って、ＳＩは、オブジェクト、記憶、編集及び／又は準備、並びに分散及び／又は移動の獲得の全域にアドレス可能なベーステクノロジである。

【００８８】

本発明は、最小のメモリ要求によってディスプレイ解像度画像上で動作することができ且つ高解像度、高品質の印刷された画像形成が可能なＳＩフォーマットを提供する。

【図面の簡単な説明】

【図１】本発明によって使用され得る汎用構成要素を示すブロック図である。

【図２】ＳＩオブジェクトを作成する種々のデータタイプを示すグラフィック表現である。

【図３】ＳＩオブジェクトを作成する複数表現（ＭＲ）の例を示すグラフィック表現である。

【図４】アスペクト比を維持するためのレンダリングを示すグラフィック表現である。

【図５】本発明によるペーストボードによって定義される次元においてラスタがいかにしてレンダリングされるかを定義する利用可能なＦｉｔＭｏｄｅ　モデルをグラフィカルに示す図である。

【図６】本発明の位置調整属性をグラフィカルに示し、位置調整属性が９個の使用可能な位置の内の一つにおけるペーストボード内に位置するようにレンダリングされたラスタを指定するのにいかにして使用されるかを示す図である。

【図７】本発明の位置調整属性をグラフィカルに示し、位置調整属性の三つの例を示す図である。

【図８】図８は本発明の角度属性をグラフィカルに示し、（ａ）側は回転しないペーストボードを示し、（ｂ）側は回転するペーストボードを示す（オブジェクトは制御点の回りを回転する。）

【図９】本発明による制御点のレンダリング調整を示すグラフィック表現である。

【図１０】構成要素オブジェクト及びこれらの構成要素オブジェクトに適用される画像処理動作（ＩＰＯ）に対する参照の構造化記述を示す出力ＳＩを記述するときの画像処理定義（ＩＰＤ）のグラフィック表現である。

【図１１】ＳＩのレンダリングの間のラスタデータの流れを示す図である。

【図１２】全ての子オブジェクトが（オブジェクトペーストボードに相対する）併合点と（子ペーストボードに相対する）制御点によって指定された位置で親ペーストボードへ併合されるのを示すグラフィック表現である。

【図１３】ペーストボードイニシャライザ機能を示すグラフィック表現であり、この場合、走査されたテキスト画像によってペーストボードを初期化する。

【図１４】本発明による（入力オブジェクトに関する）非結合ＳＩを示すグラフィック表現である。

【図１５】本発明による（画像処理動作及び／又は属性に関する）不完全ＳＩを示すグラフィック表現である。

【図１６】本発明によるオブジェクト指向レンダリングを示すグラフィック表現である。

【図１７】本発明による処理ツリーレンダリングを示すグラフィック表現である。

【図１８】本発明によってグラフィカルにディスプレイされた対応するＳＩＤファイルを有するサンプルＳＩを示す図である。

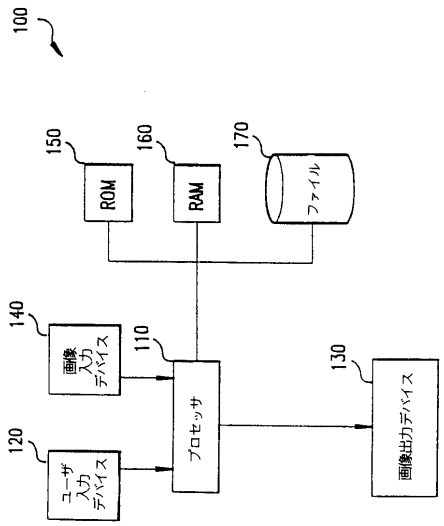
【図１９】図１８に示されているトップレベルＳＩと対応するＳＩＤファイルを示す図である。

【図２０】図１８に示されているトップレベルＳＩと対応するＳＩＤファイルを示す図である。

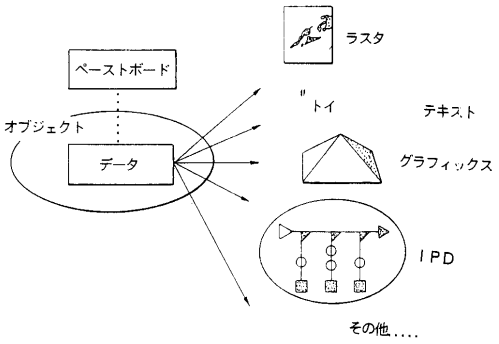
【図２１】図１８に示されているトップレベルＳＩと対応するＳＩＤファイルを示す図で

ある。
【図２２】適例の複数表現（ＭＲ）ファイルとそのグラフィック表示を示す図である。
【図２３】適例の複数表現（ＭＲ）ファイルとそのグラフィック表示を示す図である。
【図２４】本発明による固定点と制御点の相互作用をグラフィカルに示す図である。
【符号の説明】
１００　ハードウェアシステム
１１０　マイクロプロセッサ
１２０　ユーザ入力デバイス
１３０　画像出力デバイス
１４０　ＳＩ入力デバイス
１５０　ＲＯＭ
１６０　ＲＡＭ
１７０　ファイル

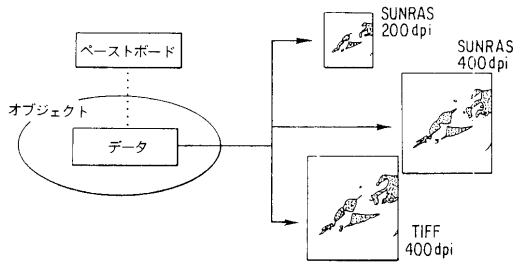
【図１】



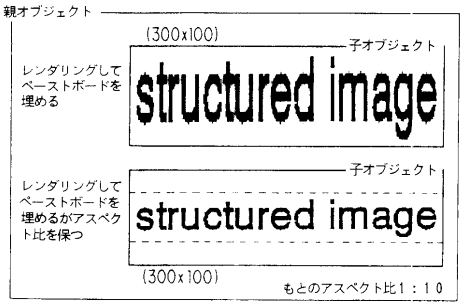
【図２】



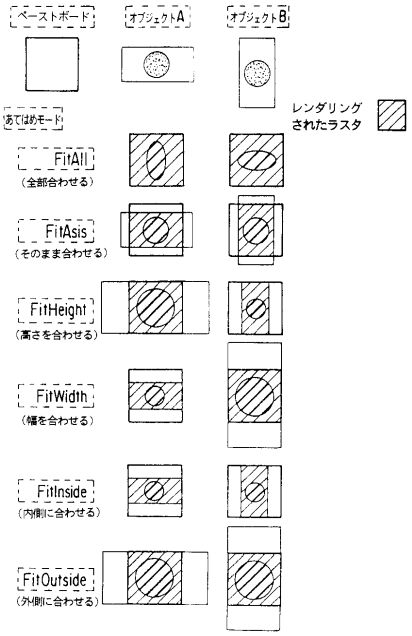
【図３】



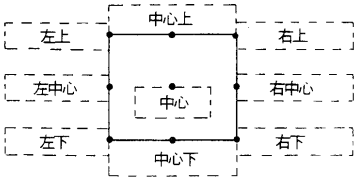
【図４】


親オブジェクト
(300x100)
子オブジェクト
レンダリングしてペーストボードを埋める
structured image
子オブジェクト
レンダリングしてペーストボードを埋めるがアスペクト比を保つ
structured image
(300x100)
もとのアスペクト比1：10

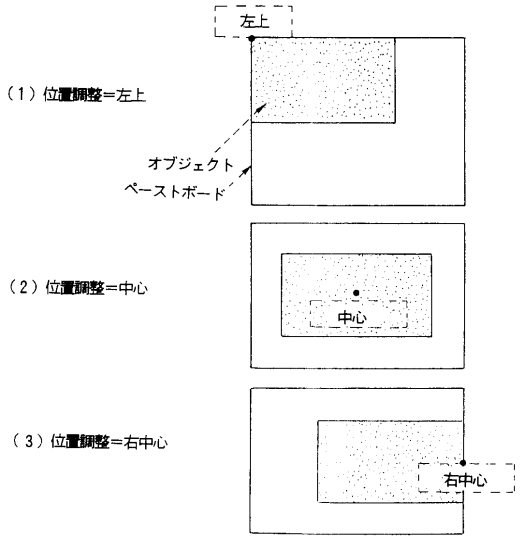
【図５】


ペーストボード
オブジェクトA
オブジェクトB
あてはめモード
レンダリングされたラスタ
FitAll
(全部合わせる)
FitAsis
(そのまま合わせる)
FitHeight
(高さを合わせる)
FitWidth
(幅を合わせる)
FitInside
(内側に合わせる)
FitOutside
(外側に合わせる)

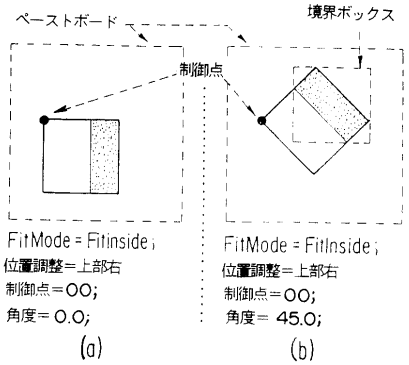
【図６】


左上
中心上
右上
左中心
中心
右中心
左下
中心下
右下

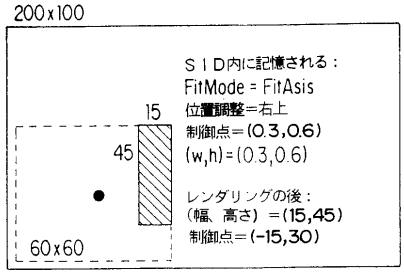
【図７】


左上
(1) 位置調整=左上
オブジェクト
ペーストボード
(2) 位置調整=中心
中心
(3) 位置調整=右中心
右中心

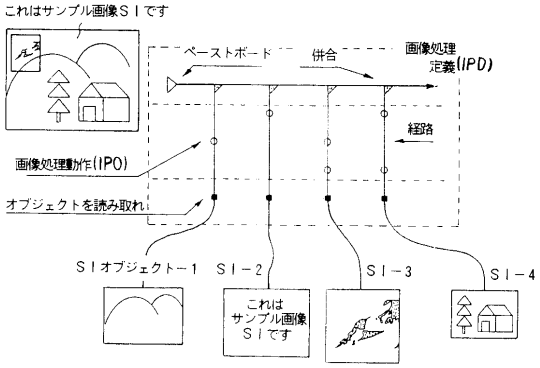
【図８】



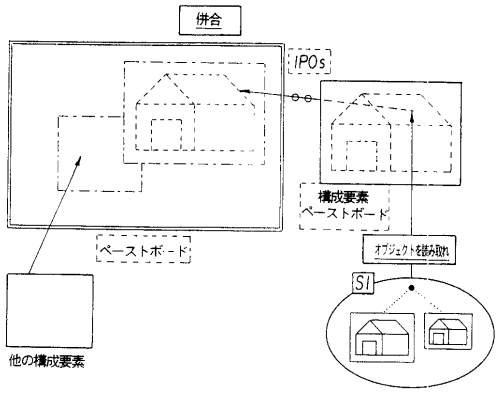
【図９】



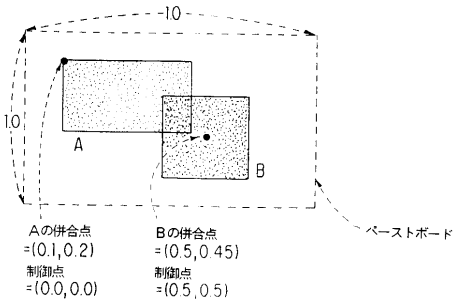
【図１０】



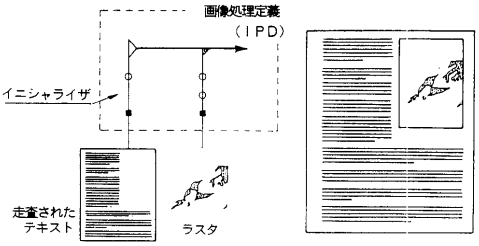
【図１１】



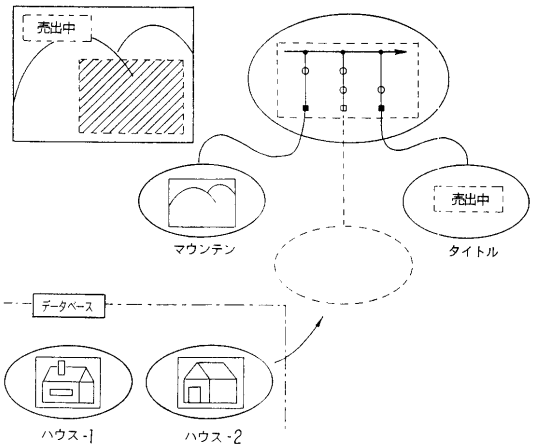
【図１２】



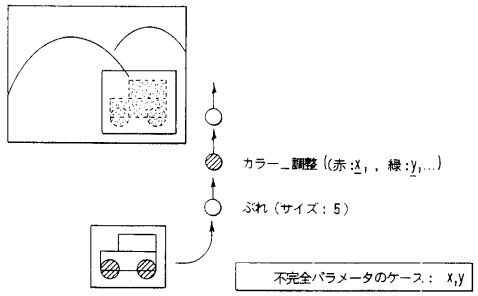
【図１３】



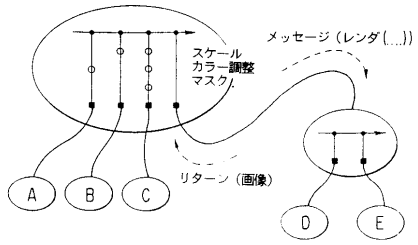
【図１４】



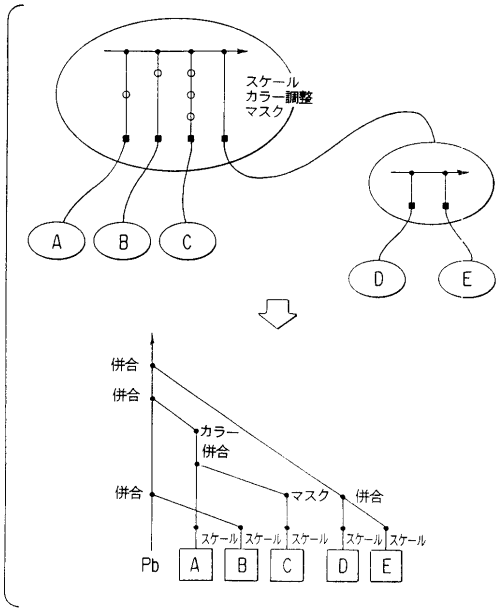
【図１５】



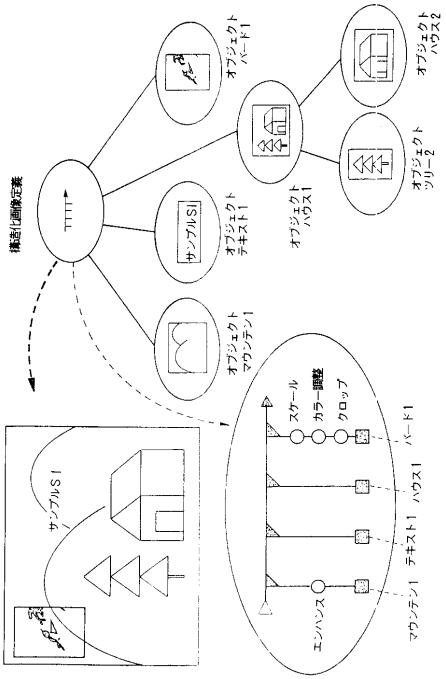
【図１６】



【図１７】



【図１８】



【図１９】

```
#!SIDL-1.1
#0000
#Copyright:(c) 1993 Fuji
Xerox Co. Ltd. and Xerox
Corp.
#Creator: Dennis L. Venable
#Created: 27-05-93 21:00:00
EDT
#Revision:1.0.0
#Root:SI-Example
#EndHeader

Object: SI-Example = {
     IPD = $ipd1 ;
     } ;

IPD: ipd1 = {
     AspectRatio = 0.7727 ;
     DefaultWidth = 11in ;
     FitMode = FitInside ;
     Justifcation = Center ;
     Angle = 0.0 ;
     Merge = $merge1;
     Merge = $merge2;
     Merge = $merge3;
     Merge = $merge4;
     };

Merge: merge1 = {
     MergePoint = 0.0 0.0;
     Path = $path1;
     };

Merge: merge2 = {
     MergePoint = 0.4 0.2;
     Path = $path2;
     };

Merge: merge3 = {
     MergePoint = 0.3 0.3;
     Path = $path3;
     };

Merge: merge4 = {
     MergePoint = 0.1 0.1;
     Path = $path4;
```

【図２０】

```
Path: path1 = {
    Object = $mount1;
    Size = 1.0 1.0;
    ControlPoint = 0.0 0.0;
    IPO: 1 = {
      IPOName = Enhance;
      };
    };

Object: mount1 = {
    External = {
      System = "unix";
      Name = "mount.tif";
      };
    };

Path: path2 = {
    Object = $text1;
    Size = 0.6 0.3;
    ControlPoint = 0.0 0.0;
    };

Object: text1 = {
    External = {
      System = "unix";
      Name = "sample.toytext";
      ;
      };
    };

Path: = path3 = {
    Object = $home;
    Size = 0.7 0.6;
    ControlPoint = 0.0 0.0;
    };

Object: home = {
    External = {
      System = "unix";
      Name =
      "house_tree.sid";
```

【図２１】

```
Path: path4 = {
    Object = $cat1;
    Size = 0.2 0.3;
    ControlPoint = 0.0 0.0

    ;
    IPO: 1 = {
        IPOName = Crop;
        Window = 0.2 0.0
        0.8 0.7;
        };
    IPO: 2 = {
        IPOName =
        ColorAdjust;
        Hue = 0.9;
        Chroma = 0.5;
        };
    IPO: 3 = {
        IPOName =
            SpatialTransform
            .
        Operation = Scale;
        Param = 1.2;
        };
    };

Object: cat1 = {
    External = {
        System = "unix";
        Name = "cat.ras";
        };
    };
```

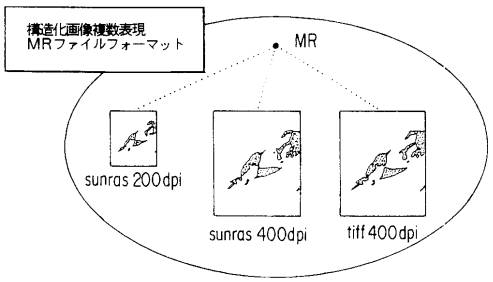
【図２２】

```
#!MR-1.0
#

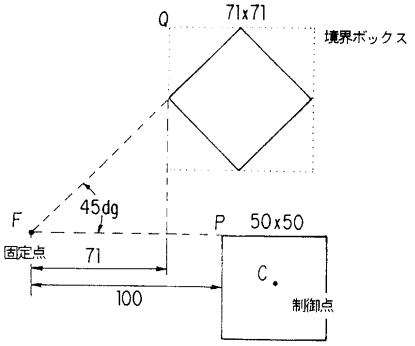
representation = {
    format = "sunras";
    attribute = {
        dpi = 200;
        };
    data                =
    "/image/cat.small.ra
    s";
    };

representation = {
    format = "sunras";
    attribute = {
        dpi = 400;
        };
    d    a    t    a
    = "/image/cat.larger.r
    as";
    };
representation = {
    format = "TiFF";
    attribute = {
```

【図２３】



【図２４】



フロントページの続き

(72)発明者　マイケル　アール．カンパネーリ
アメリカ合衆国　１４５８０　ニューヨーク州　ウェブスター　マリゴールド　ドライヴ　１１０５
(72)発明者　ウィリアム　エイ．ファス
アメリカ合衆国　１４６１２　ニューヨーク州　ロチェスター　ラッタ　ロード　７７７
(72)発明者　ジェイムズ　イー．ボルマン
アメリカ合衆国　１４５８９　ニューヨーク州　ウィリアムソン　イートン　ロード　３３２３
(72)発明者　長尾　隆
神奈川県海老名市中新田７９８番地高宮ハイツＢ－２０６
(72)発明者　山田　敏哉
神奈川県海老名市柏ケ谷９７５－2かしわ台ドミール２１，２０３号室
(72)発明者　山田　和也
神奈川県海老名市河原口６５８－２０１

審査官　伊知地　和之

(56)参考文献　特開平０５－２１０７２２（ＪＰ，Ａ）
特開平５－２５０３５８（ＪＰ，Ａ）
特開平４－９００６６（ＪＰ，Ａ）
特開昭６３－２２０３７０（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.^{7}，ＤＢ名)
G06T 11/60
G06T 11/80