グラフィクスとCAD　20－10
（1986. 2. 14）

ロボット用マルチメディアディスプレイ
― ハードウエアと基本ソフトウエア ―

塚本享治　　松井俊浩
（電子技術総合研究所）

1．まえがき

人の近づけない環境でロボットに作業させる場合には、あらかじめ予測できない事態にしばしば遭遇する。これに対処するには、ロボットとオペレータとが協力し、平常時はロボットがプログラム通り作業を行なうが、不測の事態が発生したらオペレータが介入して作業を助ける必要がある。このことを想定して、ＬＡＮに接続されたロボットの監視、プログラミング、および操作をワークステーションから行なうことのできるシステムの開発を進めている[1),2)]。このシステムのワークステーション用ディスプレイとして、作業現場の状況を映す実画像、計算機の理解する環境やロボットの状態を表わす3次元グラフィックス、および計算機と会話を行なうのに必要なテキスト、の3種類の情報をマルチウインドに高速に重ね合わせて表示することのできるマルチメディアディスプレイ（MMD）の開発を行なってきた[2)-12)]。現在までに、写真1のようなシステムを開発し、ハードウエアと基本ソフトウエアの動作を確認した。ほぼ満足のいく性能が得られたのでその結果について報告する。MMDの設計思想については、すでに報告しているため[2)]、本稿では述べない。

2．MMDの機能

MMDは画面を見ながら、ロボットのプログラムの開発、ロボットの作業状況の監視、作業環境や動作の教示、などを行なうことを想定しており、そのために以下の機能と性能を実現した。

(1) マルチメディア

遠隔におかれたロボットの作業状況を監視するための実画像、ロボットを制御し管理する計算機システムがその内部に作り上げた状況（モデル）やオペレータの教示内容をわかりやすくするための3次元グラフィックス、文字や記号による計算機システムとの対話のためのテキスト、以上3種類のメディアを同時に扱うことを可能にした。

(2) スーパインポーズ

実際のロボットの状況（環境）に計算機システムが理解したモデルを重ね合わせて、両者の食違いをわかりやすくしたり、環境に対する指示や質問をモデルへの指示や質問に置き変えるために、3種類のメディアを実画像、3次元グラフィックス、テキストの順に上にスーパインポーズできるようにした。

(3) マルチウインド

同じ環境やモデルを複数の方向から見たり、異なる環境やモデルが同時に同一画面で見られるように、６０個のマルチウインドを実現し、各ウインド毎に独立に3種類のメディアを並置したりスーパインポーズすることを可能とした。

(4) 立体視

実際のロボットの操縦やモデル世界でのロボットの操縦を容易にするために、実画像と3次元グラフィックスに対してＰＬＺＴ方式の眼鏡を使った立体視を可能とした。

(5) ピッキングとモデル計測

実画像の上に3次元グラフィックスを重ね合わせ、実際の環境に非接触で指示したり、距離の計測や問い合わせを行なうことを可能にした。

(6) 実時間処理

以上のことが、ホスト計算機と接続した状態で画面を見ながら実時間で行なえるように、ホスト計算機にバス接続し、現時点においてウインド数枚で毎秒５～１０コマ程度の描画速度を達成している。

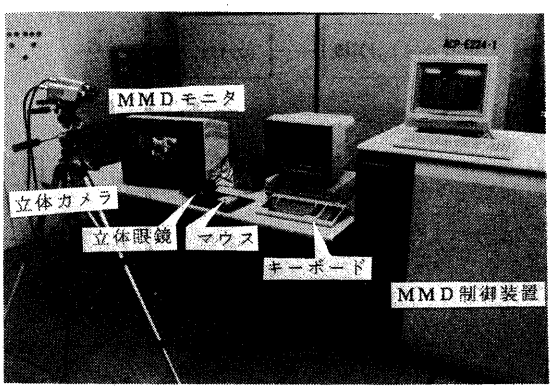

写真1　MMD全景

３．ハードウエア

MMDのハードウエア構成を図１に示す。システム全体はパイプライン構成とし、メディアごとの系統に分けて別々に画面を形成し、それらを合成して表示画面とする方式をとっている。平均 400石を実装する470㎜ X 380㎜ のボード23枚で実現している（写真２）。

３．１　システムの全体構成[2)]

システムは、MMD管理部、MMD制御部、表示計算部、重畳制御部、表示部の５ステージに分けられる。

(1) MMD管理部

MMD管理部は、モデルを作成したり、オペレータやプログラムとの会話を管理する部分であり、ホスト計算機に搭載される。

(2) MMD制御部

MMD制御部は、MMD装置全体を統括制御する制御プロセッサ（ＤＰＵ）、表示計算部と重畳制御部が画面の形成に使用するウインドマスクメモリ（WMM）、WMMを使って可視なウインドを求める[4)]ウインド管理プロセッサ（WMP）、およびMMD管理部とインタフェースをとるためのＦＩＦＯと制御レジスタで構成されている。ＤＰＵは、ホストからのコマンドを受けて４MBのセグメントバッファにモデルを構築し、ウインドポートの変更がある場合にはWMMを更新し、そののち、画面を書換えるために表示計算部の必要なパイプラインを起動する。

(3) 表示計算部

表示計算部は、左右実画像、左右グラフィックス、テキストの５本のパイプラインで構成しており、いずれもＤＰの指示を受けてメディアに応じた処理を行ないそれぞれのフレームバッファに２次元画面を作成する。

(4) 重畳制御部

重畳制御部では、各パイプラインでフレームバッファに作成した画面をモニタからの読み出し時にミキサによって合成を行なって表示画面を形成する。右表示画面は、右実画像、右グラフィックス、テキストから合成し、左表示画面は、左実画像、左グラフィックス、テキスト

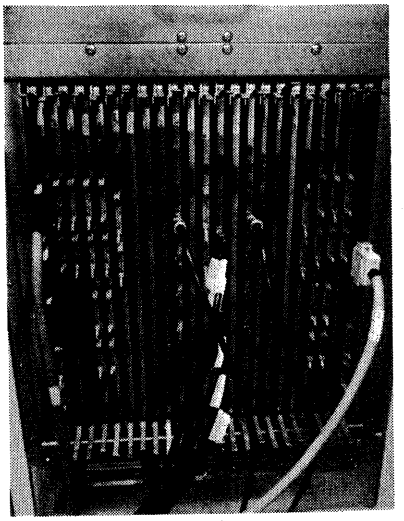

写真２　ＭＭＤ制御装置

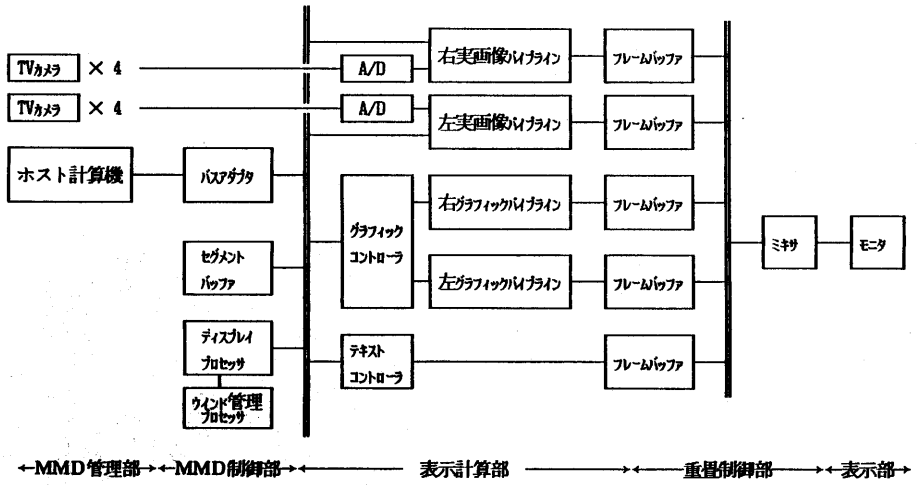

図１　ハードウエア構成

から合成する。左右の表示画面は５７ヘルツで切り換わり、立体眼鏡の電子シャッタもこれに同期して切り換わる。

３．２　マルチメディア・マルチウインド[3)]

MMDのハードウエア上の特徴は、表示計算部をメディアごとに別のパイプラインで処理し、その結果を重畳制御部で合成する方式をとった点にある。

MMDのウインドは、不透明な背景を有する矩形領域であり、ウインド名やウインドの状態などを表示する表題領域とウインドの中味を表示する作業領域からなり、周辺および領域の境界は枠で囲んでいる。ウインド内のメディア情報を表示する場所をレイヤポートと呼び、表題領域にはテキスト用のタイトルポート１つを、作業領域には実画像、グラフィックス、テキストを表示する実画像ポート、グラフィックスポート（枠を持つ）、テキストポートの３つのポートをそれぞれ任意の大きさで任意の位置に配置することができる。

このようなウインドがスクリーン上で重なると、上に重なったウインドは下のウインドの重なった部分を隠す必要があり、一部が隠されたウインドのレイヤポートに描画する際には隠された部分への描画を抑制しなければならない。同一ウインドのレイヤポートが重なる場合にはスーパインポーズを行なうため、描画の抑制はウインドが重なる場合に限られる。

1280×1024の画面を 128×128 の領域に分け、各領域ごとに画面上に見えるべきウインド番号を記憶しておくマスクメモリがWMMであり、ウインドやレイヤの配置場所や重なりの順序が変わるたびにWMMを作り直す。WMMはグラフィックパイプラインとテキストパイプラインから参照できるようになっており、これらにおいてはフレームバッファのアドレス(x,y) に色ｃを書き込む際、(x,y) に見えるウインド番号をWMMから求め、それが描画中のウインド番号と一致すればフレームバッファの (x,y)をｃで更新するが、一致しなければ更新を行なわないよう制御する。こうしてフレームバッファに表示画面が作成される。

重畳制御部では、実画像、グラフィックス、テキストをこの順に上に重ね合わせる。重ね合わせのために、テキストとグラフィックスには『透明色』なる色を導入している。すなわち、①テキスト文字においては、字画部は不透明色、背景部は透明色、②グラフィックスにおいては、辺と面は不透明色、背景と隠線処理モードの面は透明色、を割り当てる。不透明色は下のレイヤを隠すが透明色は下のレイヤを隠さないという制御を画素単位で行なっている。

３．３　実画像パイプライン

実画像パイプラインは左右２本のパイプラインで構成しており、その制御はＤＰＵが行なっている（図２）。

ＤＰＵが入力カメラをマルチプレクサ（ＭＰＸ）に設定し入力を指示すると、次のフレームの先頭からＴＶ信号を入力しＡＤ変換器（ＡＤＣ）でＡＤ変換して６ビットデータにしたのち、 512×384 のイメージレイヤメモリに書き込む。次に、ＤＰＵがイメージレイヤメモリの切出し矩形とフレームバッファの書込み矩形を設定してスケーラ（ＳＣ）にニューフレームを指示すると、ＳＣは投影法で拡大縮小変換を行なってフレームバッファに書き込む。終了はいずれも割込みでＤＰＵに通知する。

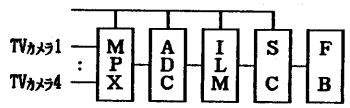

図２　実画像パイプライン

３．４　テキストパイプライン

テキストパイプラインは図３の構成となっている。テキストコントローラ（ＴＬＣ）は、ＤＰＵから表示を更新する文字またはテキストレイヤが与えられると、ＷＭＭを使って更新の必要な部分を求め行単位で文字コード列に展開してラインバッファ（ＬＢＦ）に送出する。キャラクタゼネレータ（ＣＧ）で文字をピクセルに展開し、スケーラ（ＳＣ）によってＳＰＣ法で拡大縮小変換してフレームバッファに書き込む。

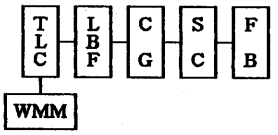

図３　テキストパイプライン

３．５　グラフィックパイプライン[9)]

グラフィックパイプラインは、グラフィックコントローラ（ＧＣＵ）以下が左右２本のパイプラインに分かれた構成となっている（図４）。

グラフィックコントローラ（ＧＣＵ）は、ＤＰＵから表示を更新するウインドの集合を得ると、各ウインドのグラフィックレイヤに結合されたモデルのデータ構造をトラバースして表示データとして送出する。座標変換プロセッサ（ＴＰ）は、座標データに対しカレントマトリックスとの乗算を行ない、座標変換処理を施こす。カレントマトリックスは、ウインドビューポート変換、透視変換、ネストしたモデリング変換などを表わすマトリックスを１つに結合したもので、このためＴＰはマトリックスの結合演算も行なう。これらのマトリックスはモデリング変換のようにモデル毎に変化するものと、ウインドビューポート変換、透視変換のようにウインド単位でしか変化しないものがある。これらのマトリックスは結合される毎にＴＰ内にスタックして効率的に使用する。１つのマトリックスは６４バイトからなり、スタックレベルは２５６レベルである。クリッププロセッサ（ＣＰ）は、線分や面に対しビューポートによるクリッピングを行なう。ビューポートとの交点計算は同次座標空間において比例計算により直接求めている。エリアフィルプロセッサ（ＦＰ）は線分のまま、面はラスタ毎の線分に分解して、直線発生器（ＤＤＡ）へ出力する。ＤＤＡは線分をピクセルに展開してフレームバッファに書き込む。その際、デプスバッファ（ＤＢＦ）を用いて隠面処理を実行する。隠線処理は、面の色が透明色、線分の色が不透明色である隠面処理として扱うことができる。

面処理が中心となるため、ＣＰとＦＰのパイプラインをさらに次のように細分化して高速化を図った。ＣＰを座標軸に対応して３組のプロセッサ(CPX,CPY,CPZ) に分けてパイプライン化している。CPX,CPY,CPZ は同一のハドウエア構成で、CPZ のみが物体のｚ座標クリップによる切断面処理のために特別な処理を行なっている。ＭＭＤでは物体をストラクチャと称するセグメントの上位概念で表わしており、その単位で切断面の処理を行なっている。

ＦＰも３組のプロセッサ(FP1,FP2,FP3) によりパイプライン化を行なった。FP1 はＹソートされたエッジリストの作成、FP2 は１ラスタ毎のｘ接片を求める処理、FP3 はｘ接片をソートしてＤＤＡに出力する。

FP3 の出力は水平線分であるためＤＤＡは水平方向に適した構成をとっている。また、隠面処理では隠される面も処理する必要がありＤＤＡの描画する画素の量が増加するため、ＤＤＡはＥＣＬ素子を使用して高速化を図っている。

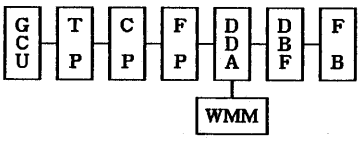

図４　グラフィックパイプライン

４．制御ソフトウエア

ＭＭＤ制御ソフトウエアは、ホスト計算機とのインタフェースを管理しＭＭＤ装置全体を統括制御するソフトウエアであり、ＤＰＵに搭載されている。アセンブリ言語で記述され、サイズは約８０Ｋバイトである。

４．１　制御ソフトウエアの構成[10)]

制御ソフトウエアは、コマンド解析部とニューフレーム部に分けられる。コマンド解析部では、コマンドを入力・解析してデータ構造に変更を加え、ニューフレームに必要な情報を生成する。一方、ニューフレーム部はその情報をもとに必要なパイプラインに対してニューフレームを起動する。

コマンドの入力とパイプラインの制御は、本質的に非同期動作するため、それを制御するソフトウエアはそれぞれに専用タスクを配するマルチタスク構成とし、制御ソフトウエアは次の６つのタスクで構成した（図５）。

(1) コマンドバッファ解析タスク
　コマンドバッファからの入力で起動され、コマンドを解析する。
(2) コマンドＦＩＦＯ解析タスク
　コマンドＦＩＦＯからの入力で起動され、コマンドを解析する。
(3) グラフィック部ニューフレームタスク
　コマンド解析部から起動され、WMMの更新、ウインドの表示・消去・移動、およびグラフィックパイプラインに対するニューフレームの起動を行なう。

(4) テキスト部ニューフレームタスク

一括処理されたタイトルレイヤとテキストレイヤの表示・消去・移動のために、テキストパイプラインに対してニューフレームを起動する。

(5) イメージ部ニューフレームタスク

イメージレイヤの表示・消去・移動のためにイメージパイプラインに対しニューフレームを起動する。

(6) イメージ部サンプリングタスク

イメージの取込み準備を行なったのち、イメージパイプラインに対して取込みを指示する。

4．2　ホストインタフェース

ホスト計算機のメモリに直接マッピングされたセグメントバッファとイベント通知用の３２ビットの制御レジスタを使ってコマンドＦＩＦＯを補強するインタフェースをセグメントバッファ上に実現しており、MMDは次のインタフェースを持つ装置として見える。

(1) コマンドＦＩＦＯ

即時処理するコマンドをコマンドＦＩＦＯ解析タスクに伝えるためのハードウエアＦＩＦＯ。

(1) コマンドバッファ

一括処理可能な大量のコマンドをコマンドバッファ解析タスクに伝えるためのリングバッファ。

(2) レスポンスバッファ

コマンドに対する応答データとエラーレポートのためのリングバッファ。

(3) コマンドＦＩＦＯマーク

コマンドＦＩＦＯ経由で伝えられたコマンドの終了状況を伝えるためのソフトウエアレジスタ

(4) コマンドバッファマーク

コマンドバッファ経由で伝えられたコマンドの終了状況を伝えるためのソフトウエアレジスタ

(5) エラーステータスとリカバリステータス

ハードウエア障害をホストに伝え、そのリカバリハンドシェイクを行なうためのソフトウエアレジスタ

コマンドＦＩＦＯ、コマンドバッファ、および対応するマークの組み合わせ方はホストのMMD管理部に任せ、MMD制御部は処理の仕方だけを定めている。

コマンドＦＩＦＯとバッファを通してコマンド解析部が受けつけるコマンドとしては、①MMD／スクリーン制御関係１９個、②ウインド制御関係１３個、③レイヤ制御関係１２個、④グラフィックモデリング関係６８個、⑤ビューイング関係１８個、⑥イメージ関係９個、⑦テキスト関係３３個、⑧ピッキング関係3個、計１７５個のコマンドを実現した。

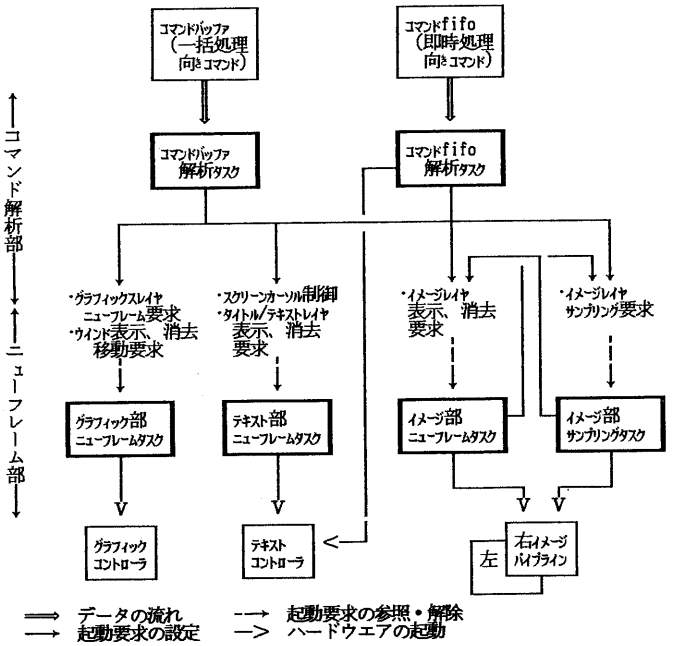

図5　制御ソフトエウアの構成

4．3　データ構造[5),6)]

コマンド解析部がセグメントバッファ中に管理するデータ構造の主な部分の構成を図６に示す。構造要素ごとの説明を以下に加える。

(1) ウインド

表示する情報をまとめ上げ、かつそれに表示場所を提供するものであり、可視・強調属性、優先度、形状、ウインドポート、色、などの属性と、左右の実画像レイヤ、最大８個のグラフィックスレイヤ、タイトルレイヤ、左右のテキストレイヤ、左右のビューイング、およびそれらに関連する情報から成る。

(2) タイトルレイヤとテキストレイヤ

文字列を格納するものであり、複数の行バッファから成る。

(3) イメージレイヤ

TVカメラからの映像を記憶するメモリであり、カメラ識別子、動作状態、可視属性、イメージメモリのアドレス、キューから成る。

(4) グラフィックレイヤ

モデルを応用上の意味に合わせてグループ化する単位であり、可視・強調・検出属性，描画法とストラクチャのリストから成る。

(5) ストラクチャ

セグメントとストラクチャを複数含んだ構造物を定義するものであり、それらのリスト、可視・強調・検出属性、ストラクチャカラー表、ストラクチャモデリング変換マトリックスから成る。

(6) スタティックセグメント

形状を定義するものであり、COREの出力プリミティブに、面、面縁、穴、などを追加したプリミティブ列から成る。

(7) ダイナミックセグメント

スタティックセグメントに動的属性を与えるものであり、可視・強調・検出属性、セグメントモデリング変換マトリックス、および他のダイナミックセグメントへのポインタから成る。

(8) GDP

一連のプリミティブ列をサブルーチン化するものであり、プリミティブ列から成る。

(9) ビューイング

ワールド空間から投影空間への視野変換マトリックスと投影変換マトリックス、およびCORE型の個別パラメータから成る。

これらのうち、ウインド、タイトルレイヤ、テキストレイヤ、グラフィックレイヤ、イメージレイヤ、ビューイング、ストラクチャ、スタティックセグメント、GDPは、それぞれの一覧表に登録されており、表中の番号で識別し参照される。

識別番号で指定して属性を変更することのできる構造要素のうち特に、グラフィックスレイヤ、イメージレイヤ、ビューイング、ストラクチャ、には次に述べるウインド参照ベクトルを設けている。

4．4　ウインド参照ベクトル [3)]

マルチウインドシステムでは、効率を上げるためにニューフレームの量を最低限度に抑えることが重要である。参照しているウインドの集合を記憶する64ビット

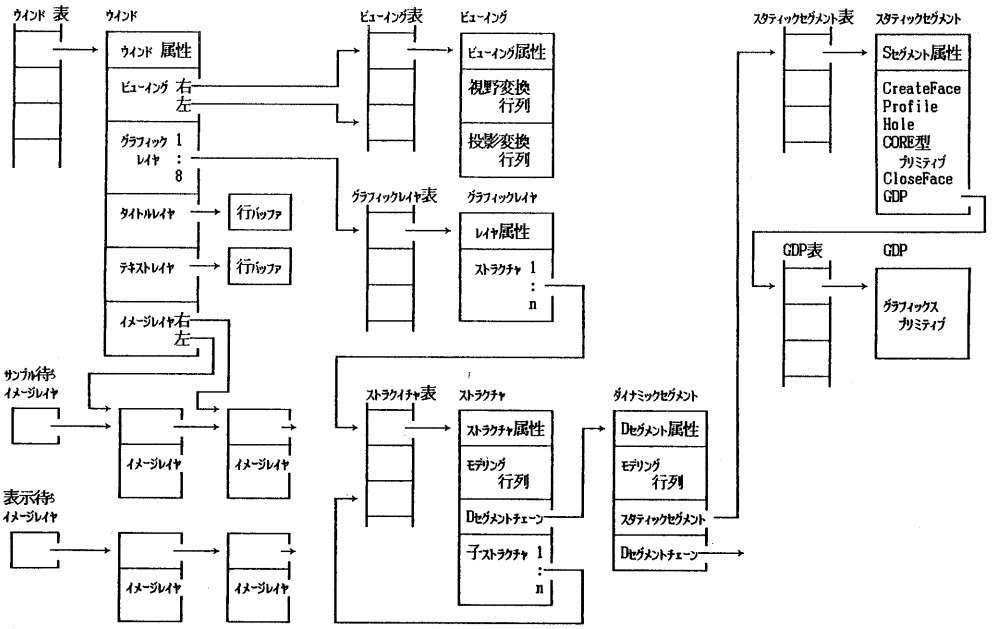

図6　制御ソフトウエアデータ構造

のビットベクトルをウインド参照ベクトルと呼ぶ。ウインド参照ベクトルにおいては、ビットｉ＝１はウインドｉが参照していることを、ｉ＝０はウインドｉが参照していないことを意味している。

コマンド解析部のタスクで構造要素に変更を施す際には、それによって起動が必要となるニューフレームタスクに対応するウインド参照ベクトルの記憶場所（ニューフレームベクトルと呼ぶ）に論理和をたし込む。求まったニューフレームベクトルはニューフレームの必要なウインドを表わしており、これをコマンド解析終了時にニューフレーム部のタスクに渡す。

ニューフレーム部のタスクはニューフレームベクトルをそれぞれのニューフレーム保留ベクトルに記憶したのち対応するパイプラインに対してニューフレームを起動する。パイプラインは、ウインドのニューフレームが終了するたびに、ニューフレーム保留ベクトルから対応するウインドのビットをクリアする。

ニューフレーム部のタスクに起動をかけると、コマンド解析部のタスクは新たなコマンドの入力・解析が可能となる。しかしながら、ニューフレーム保留中のウインドが参照するデータ構造に対して変更を加えると画面上に矛盾が生じる可能性がある。これを避けるため、コマンド解析部のタスクは、変更の対象となっている構造要素のウインド参照ベクトルとニューフレーム保留ベクトルとの論理積をとり、０でなければ０になるまでコマンド解析を保留する。

５．管理ソフトウエ [11)]

MMD管理ソフトウエアは、MMD装置とユーザのインタフェースを管理するソフトウエアである。現在、ホスト計算機としてはUNIXsystemVを搭載するU－station を使用しており、管理ソフトウエアはC言語で記述している。

５．１　管理ソフトウエアの構成

管理ソフトウエアは、ドライバ部と常駐管理プロセス部から構成される。ドライバ部では、MMDの各種識別番号の資源管理とMMDへのインタフェースを実現する。一方、常駐管理プロセス部では、ローカルデバイス、それを使ったMMDとユーザプロセスの管理、およびユーザプロセスからMMDへのアクセスの管理を行なう。

UNIXでは、システムコールは同時には１つしか発行できないので、複数のポートからのイベントを待つプログラミングがむずかしい。そこで、常駐管理プロセス部はマルチプロセス構成とし、図７のような５つの常駐プロセスで構成した。

５．２　MMDドライバ

MMDドライバ(mmddrv)は、特権を持たないプロセスに代わってMMDにアクセスする。MMDドライバは、マイナデバイス番号を違えて、mmdcmnd,mmdfifo,mmdmem の３つのデバイスをサポートしている。mmdcmdはコマンドバッファを通してコマンドを送出し、mmdfifoはコマンドFIFOを通してコマンドを送出する。mmdmemはMMDのアドレスを指定して内部メモリにアクセスする。

複数プロセスからの同時使用を可能としており、通常のデバイス同様に、openとclose で囲んだ内部において、コマンド出力用write,レスポンス入力用read，コマンドとレスポンス以外の制御用ioctl の各システムコールを発行することができる。 ioctlでは、MMD制御部が管理しているテーブルから、ウインド、レイヤ、ストラクチャ、コマンドマーク、レスポンスマークなどの未使用識別番号を確保し開放する機能、エラー、コマンドマーク、レスポンスマークなどの到着を待たせる機能、複数プロセス間におけるmmdcmnd,mmdfifo,mmdmemの相互排除機能、などを実現している。

５．３　常駐管理プロセス

常駐管理プロセスのそれぞれに関して次に説明する。

(1) MMDマネジャ(mmdman)

mmdmanは、管理ソフトウエアの中核となるプロセスであり、キューからのコマンドを受けて、ウインド作成、プロセス作成、ウインドとプロセスの結合、ウインドの呼出し、カラーパレットの設定などを行なう。

共有メモリを使って、ウインド、プロセス、FIFO型パイプ、キューなどの識別番号、タイトルなどのデータベースを管理している。

(2) ＴＴＹマネジャ(mmdtty)

mmdttyはキーボードの入力を目的のプロセスにマルチプレクスする。プロセスのstdin をFIFO型パイプに接続しておけば、mmdttyはキーボードtty からの入力を選択されたFIFO型パイプに出力する。

(3) マウストラッカ(tracker)

現在、３ボタン式のマウスを使用している。trackerは、マウスの動きを読んでスクリーンカーソルの移動コマンドをmmddrvに送る。押下されたボタンの位置によってローカルなウインド操作（ウインドのポップアップ、移動、拡大縮小、MENUウインドの呼出し）を行ない、それ以外の場合には、mmdmanにマウスイベントを通知する。

(4) エラーレポータ(mmderror,stderr)

mmderrorは、レスポンスバッファを通してMMDから通知されてくるエラーレポートを読み取ってウインドMMDERRに表示する。stderrは、UNIXが検出したエラーをウインドSTDERRに表示する。

５．４　MMDフィルタ

UNIXの既存プロセスからの出力をMMDに表示するためのフィルタとして、端末シミュレータmmdvt を設けている。mmdvtは、stdinから入力した８ビットコードやVT100 のエスケープシーケンスをMMDのコマンドに変換して特定のウインドに表示・制御する。

５．５　ユーザプロセスとの接続

MMDを利用するユーザプロセスには、次の３つのものを一部または全て付加することができる。第１はウインドに表示するために出力端に接続するmmdvt 、第２はキーボードから入力するために入力端に接続するmmdtty、第３はマウスからのイベントを入力するためのキューである。キューを付加すれば、ユーザが独自なメニューウインドを定義したり、マウスを使ったピッキングを行なうことがことが可能となる。

現在のところ、グラフィックス機能を利用するプログラミングはＬｉｓｐ上に実現した対象指向型プログラミングシステムで行なっている[12]。

最後に、MMDの画面の例を写真３に示す。３次元グラフィックスのマニピュレータ２台がスムーズに動いており、まだ定量的な性能評価を済ませていないが、当初の目標性能（毎秒１０ニューフレーム）[2]に近い値が出ているようである。

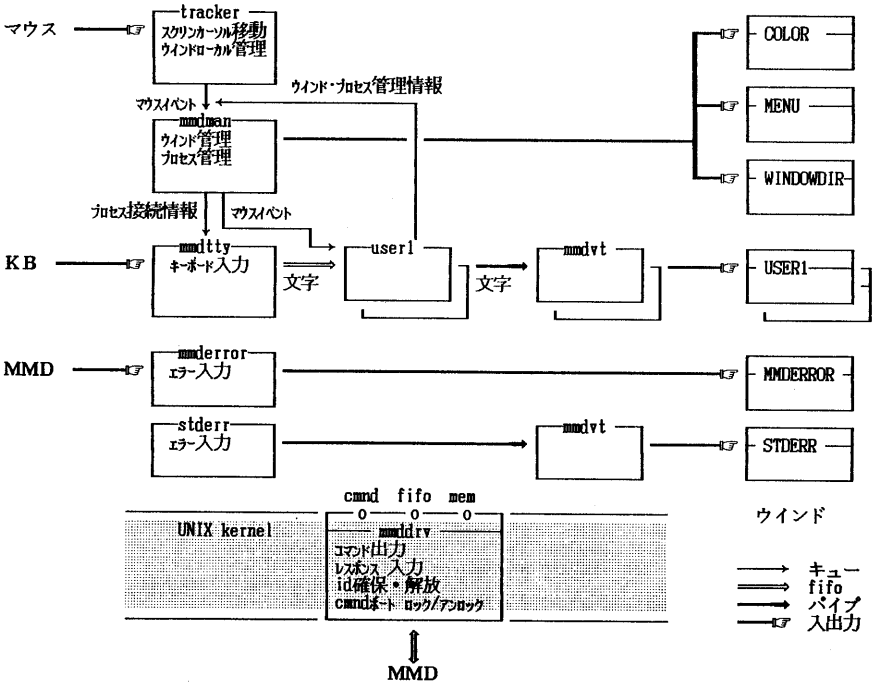

図7　管理ソフトウエアの構成

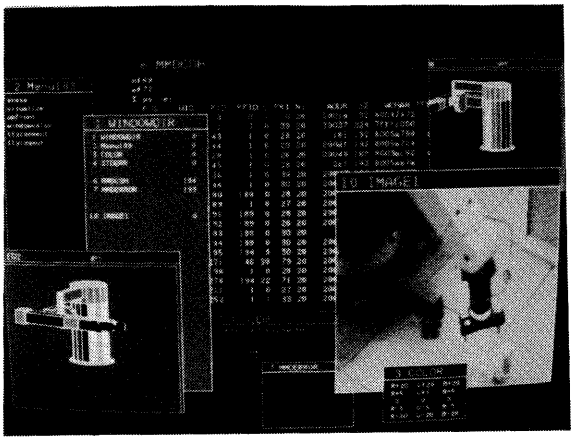

写真３　ＭＭＤの表示例

６．むすび

実画像、３次元グラフィックス、テキストの３種類のメディアを重ね合わせて表示することのできるマルチメディアディスプレイのハードウエアと基本ソフトウエアを開発し、その動作を確認したので報告した。今後、基本ソフトウエアの整備を進め、別に開発を進めている分散計算システムへ統合し、分散型のロボットシステムとして仕上げる計画である。

謝辞　本研究は通産省大型プロジェクト『極限作業ロボット』において行なわれたものであり、その機会を与えられた、当所、白井良明制御部長、若松清司前制御部長、赤堀寛システム制御研究室長に感謝致します。また、ハードウエアの製作を担当された三菱電機西出政司氏に感謝致します。

参考文献

1) Tsukamoto, M. etal. : "Conceptual Design of a Distributed Operating System for Intelligent Robots", 83ICAR (1983,9)
2) 塚本、松井：『ロボット用マルチメディアディスプレイー設計思想ー』、情報処理、オペレーティングシステム研究会資料24 (1984,9)
3) 塚本、松井：『マルチビューポート制御方式』、情報処理第29回全国大会 (1984,9)
4) 塚本、松井：『ロボット用マルチメディアディスプレイーニューフレームアルゴリズムー』、情報処理第30回全国大会 (1985,3)
5) 塚本、松井、西出：『ロボット用マルチメディアディスプレイーグラフィック機能ー』、情報処理第31回全国大会 (1985,9)
6) 松井、塚本、西出：『ロボット用マルチメディアディスプレイーモデリング機能ー』、情報処理第31回全国大会 (1985,9)
7) 松井、塚本：『ロボット用マルチメディアディスプレイーテキスト機能ー』情報処理第30回全国大会 (1985,3)
8) 松井、塚本：『ロボット用マルチメディアディスプレイー実画像機能ー』、第３回日本ロボット学会学術講演会(1985,11)
9) 西出、塚本、松井：『ロボット用マルチメディアディスプレイーグラフィックハードウェアー』、情報処理第31回全国大会 (1985,9)
10) 塚本、松井、大野、佐々木：『ロボット用マルチメディアディスプレイー制御ソフトウエアー』、情報処理第32回全国大会 (1986,3予定)
11) 松井、塚本：『ロボット用マルチメディアディスプレイー管理ソフトウエアー』、情報処理第32回全国大会 (1986,3予定)
12) 松井、塚本、小笠原：『対象指向型記述を用いたモデリングシステム構成法』、第２回日本ロボット学会学術講演会(1984,12)