

## 視覚刺激による次注視点決定の場の中心窩視野への拡張とその効用

喜多伸之

産業技術総合研究所 知能システム研究部門

あらまし：反射的および意図的な注視制御の統合的に実現するために、注意を引き付ける刺激すべてを視野上の視覚刺激として表現し、その中から注意をはらうべき度合いに応じて次に注視すべき3次元位置を決定する手法を先に提案した。ただし、システム的能力上、透視投影による狭い視野を用いたため、注意の及ぶ範囲が狭い範囲に限られていた。実用のためには、より広い範囲に注意を払えることが必須であるので視野を拡大する努力を行ってきたが、この度、中心窩視野を利用できるよう拡張できたのでそれについて報告する。特に視野の拡大に伴う計算量の増加に対処するための工夫に焦点をあてつつシステムの拡張部分の実装について説明し、さらに、次注視点決定の場が、中心窩視野の利用により一期的に広がったことの効用について実例を用いて紹介する。

### Next Gaze Point Decision in the Foveated View Field

Nobuyuki KITA

Intelligent Systems Research Institute

National Institute of Advanced Industrial Science and Technology

**Abstract** : We proposed a general framework of visual attention control for efficiently executing multiple tasks involving both reactive and premeditated types of attention. The system need wide field of view for the real utilization. In this paper, the extension of the system to treat wider field of view is introduced.

#### 1. はじめに

MIRU2000において、能動指標を用いることによる反射的および意図的な注視制御の統合的実現手法を提案した[1]。その手法は注意を引き付ける刺激すべてを視野上の視覚刺激として表現し、その中から注意をはらうべき度合いに応じて次に注視すべき3次元位置を決定するというものであった。したがって、どの程度広範な刺激を考慮できるかは刺激表現の場である視野の広さに依存している。我々は当初よりこのことを意識して中心窩画像を獲得するためのデバイス開発をおこなったが、MIRU2000の時点では、意図的な注視刺激の存在する仮想空間において中心窩視野画像を合成できるにいたっておらず、一般的な透視投影に基づく視野を用いて実験を行っていた。今回、仮想空間の中心窩視野画像を合成できるよう拡張し、中心窩視野を視覚刺激表現の場として利用できるようになった。本論では視野の拡大に伴う計算量の増加に対処するための工夫に焦点をあてつつシステムの拡張部分の実装について説明し、さらに、次注視点決定の場が、中心窩視野の利用により一期的に広がったことの効用について実例を用いて紹介する。

#### 2. 注視制御の概要

複雑な環境内を移動し作業するロボットの視覚に要求されるタスクには、障害物の発見などのように反射的注視制御を必要とする視覚タスクもあれば、既知の環境を点検するときのように、あらかじめスケジュールされた箇所への意図的注視制御を必要とする視覚タスクもある。我々は両者を含む複数の視覚タスクが与えられた場合に次の注視点を決める枠組みを次のように提案した。まず個々のタスクについてはタスクに応じた決定過程により並列に次注視点の候補を決定する。次にすべてのタスクの次注視点を候補としてさらに絞り込む。そのため、例えばタスクに与えられた優先度や、すべてのタスク遂行のトータルな効率など、共通に比較できる属性を用いて選択基準とする。ただし、反射的な注視点候補は入力画像上の刺激、つまり2次元情報として与えられ、一方、

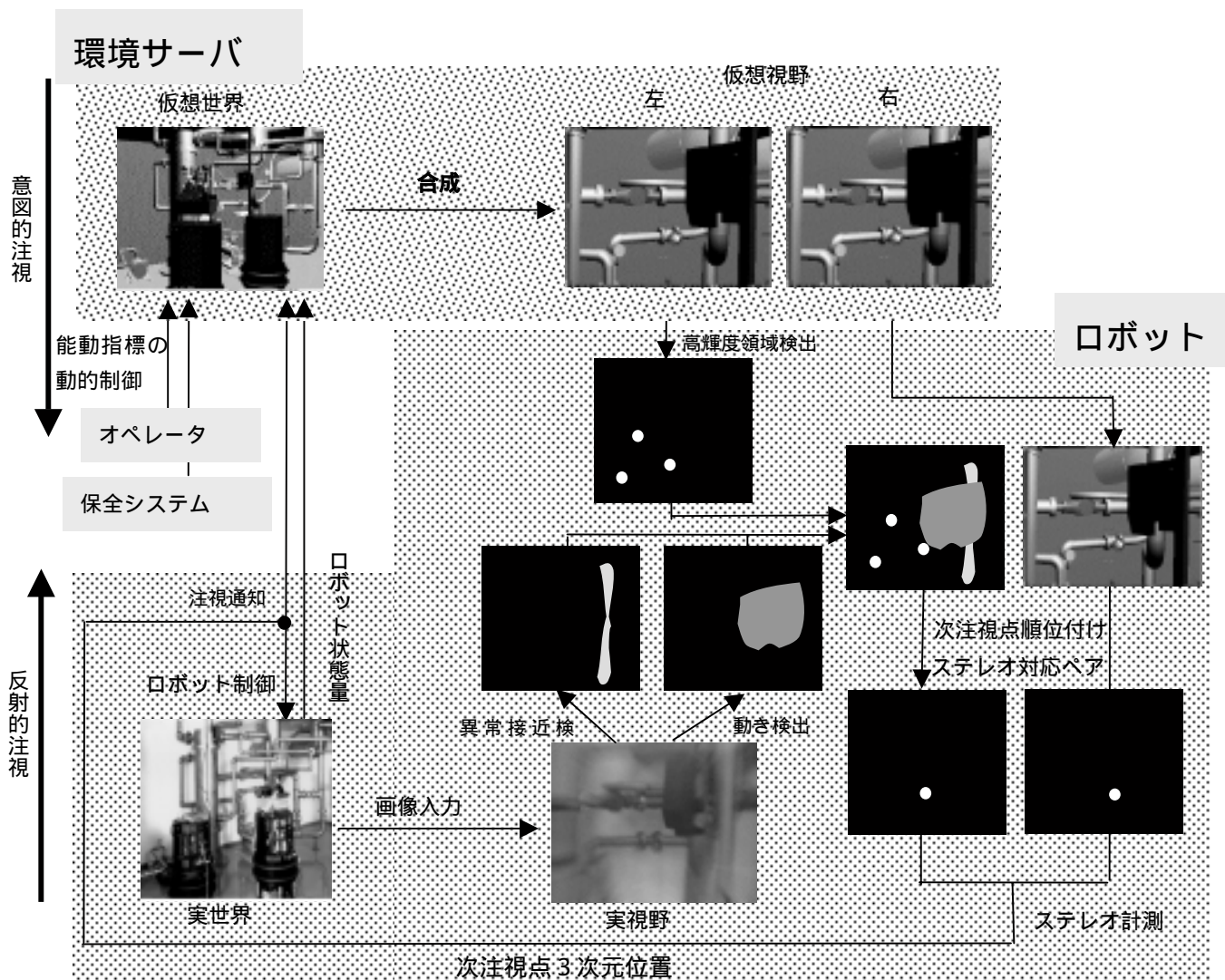


図 1 次注視点決定の仕組み

意図的な注視点候補は3次元情報として与えられるので、空間上の距離や大きさなど有用な属性が選択基準に考慮できない。そこで、視野画像が基準とする2次元空間を共通の場として採用し、3次元情報として与えられる意図的な注視の候補を視野空間に変換し、実際の視野から画像処理によって抽出した反射的な注視の候補と融合する。これによりすべてのタスクの次注視点候補が視野空間上の特徴として得られるため、それらから唯一の次注視点を選択する効果的な基準の設定が可能になる(図1)。

### 3. 環境サーバと能動指標

この仕組みを具体化するためには意図的な注視の候補を3次元空間中の可視の媒体により表現し、それらをロボット視覚のカメラパラメータにしたがって視野空間に変換できる機構が必要である。環境サーバと能動指標はこのためのツールとして提案した。

#### 3.1 環境サーバ

環境サーバは次のような機能を持つ。

- (1) ロボットが動き回る環境を忠実にモデル化した仮想環境(幾何モデル・反射モデル・カメラモデル・照明モデルなどをVRMLで記述)を維持する機能。

- (2) 仮想環境内でロボットや能動視覚装置の実際の動きをオンラインで再現する機能。
  - (3) 能動視覚をはじめとして、ロボットの持つセンサの入力をシミュレーションする機能。
  - (4) サーバとして多様なクライアントと情報の授受をおこなうための通信機能。
- である。

### 3.2 能動指標

環境サーバが管理する仮想環境内で、意図的な注視の候補を表す媒体である。具体的には能動指標は発光する仮想球であり、仮想環境内の任意の場所に設定でき、仮想環境内を移動する他のロボットにも貼り付けることができる。現在、この能動的指標の動作は、種々の時間パターンにしたがう点滅と、折れ線上の等速移動であり、これらのパラメータは先に述べた環境サーバの対話機能により初期設定でき、さらに、他の運転システムなどからの動的な変更や追加・削除も可能である。この能動指標はロボットの視野を合成した画像上に輝度の高い点として現れる。

## 4. 中心窩画像合成

能動視覚装置 ESCHeR の中心窩レンズによる投影の像高曲線は図 2 のようである[2]。これを 1/3 インチ CCD により撮像すると、水平方向の視野角度が約 120 度となり、さらにこれを、320x240 画素の画像として読み込むと、水平方向の解像度変化は図 3 となる。この中心窩投影を模擬した画像を視野中心での解像度を補償するよう透視投影画像から合成するためには、中心解像度が約 20 画素/度以上の解像度で視野角度約 120 度の透視投影画像が必要である。これは少なくとも水平方向約 4000 画素の画像を合成しなければならないことを意味している。我々は以下のような工夫により効率よく中心窩画像を合成する手法を提案し実装した。

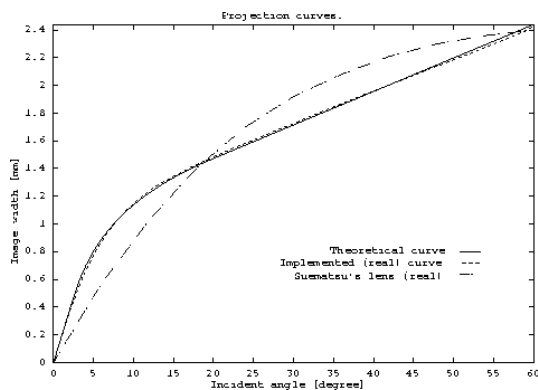


図 2 中心窩投影の像高曲線

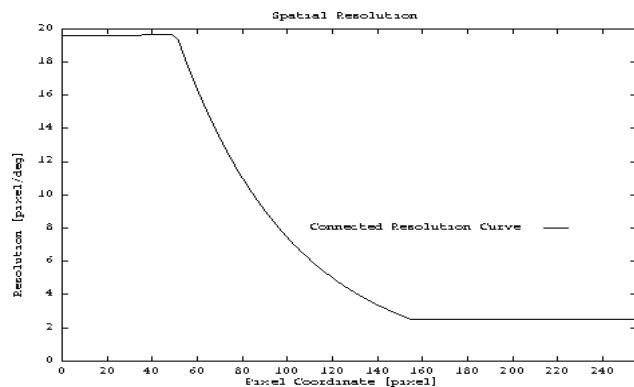


図 3 中心窩投影の解像度曲線

## 4.1 手法

### 4.1.1 視野分割合成

先に示したように 1 枚の透視投影画像で大きな視野角をカバーしようとする必要画素数が膨大になる。したがって、視野角を分割して合成することを考える。水平方向の視野 120 度を 1 枚の透視投影画像でカバーしようとする、4000 画素以上必要であったが、2 等分して水平方向の視野 60 度とすると、それをカバーする透視投影画像は約 1300 画素となる。このように、分割による効果は非常に大きく、分割数を増やすにしたがって、全視野を合成するために必要なトータルな画素数は減少していく。ただし、画像合成に要する処理量は画素数に比例する部分のほかに、合成回数に比例するオーバーヘッド部分などがあり、これらは合成に使用するソフトやハードに依存するため、最適な分割数は一般的には決まらない。

#### 4.1.2 解像度分担合成

中心窩画像を分割した視野は、解像度の高い部分と低い部分が混在する。これをほぼ解像度が一樣な1枚の透視投影画像を元画像として合成するのは無駄が多い。ここでは、解像度の高い部分をカバーする高解像度の透視投影画像と、分割視野全体をカバーする低解像度の透視投影画像を用いた。図4(a)(b)は(c)に示した左右2分割された左側視野を合成するための、高解像度透視投影画像(382x705画素)と、低解像度透視投影画像(430x674画素)である。左側視野の半円で囲まれた内側の画素は高解像度画像から、外側の画素は低解像度画像から計算している。半円で示した分割のための視野角の閾値は、合成に必要な2枚の透視投影画像の合計面積を最小にする値を求め用いている。

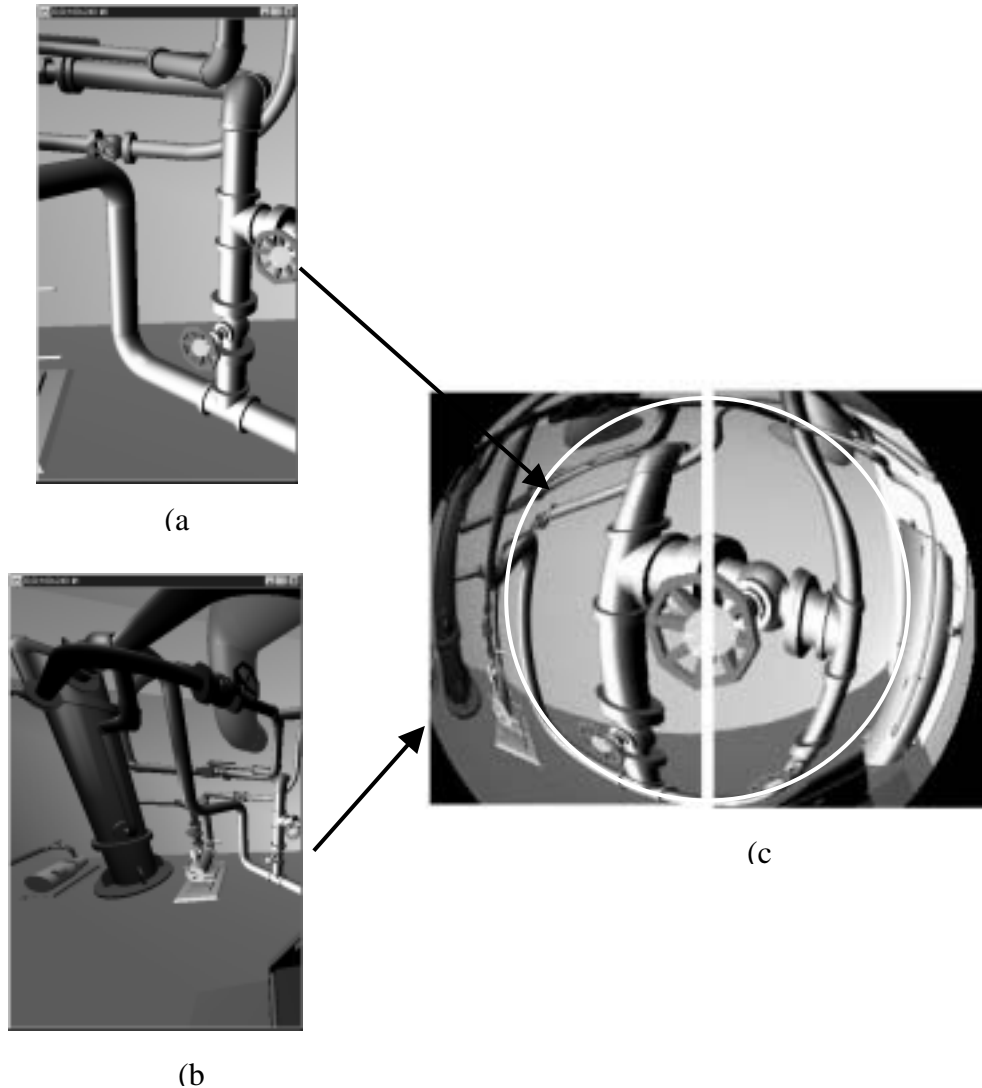


図4 解像度分担による効率化

に座標変換すると、例えば図5のようになる。ここでは、中心窩画像の1画素の領域を透視投影画像上に変換したものを、この4頂点で囲まれる四角形領域で近似するものとして、合成画像上の1画素の輝度値を求める。具体的には、この領域に含まれる各画素の輝度値を面積比で重み付け平均した値を計算する。ただし、対象となるピクセルインデックスおよび、重みは予め算出し、実行時にはテーブルからよみ

#### 4.1.3 画像変換

先に中心窩レンズによる投影の像高曲線を図2に示したが、これは設計時の理論的なものであり、実際のレンズでは加工精度などにより理論曲線からはずれる。そこで、個々のレンズに対してキャリブレーションにより像高曲線を求めるが、簡単のため5度おきに像高を求め、その間をラグランジェ補間して像高曲線を近似した。中心窩画像の1画素の4頂点をこの像高曲線にしたがって透視投影画像上

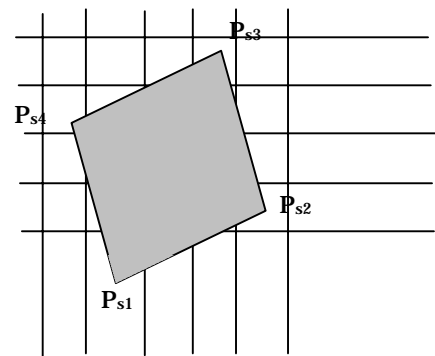
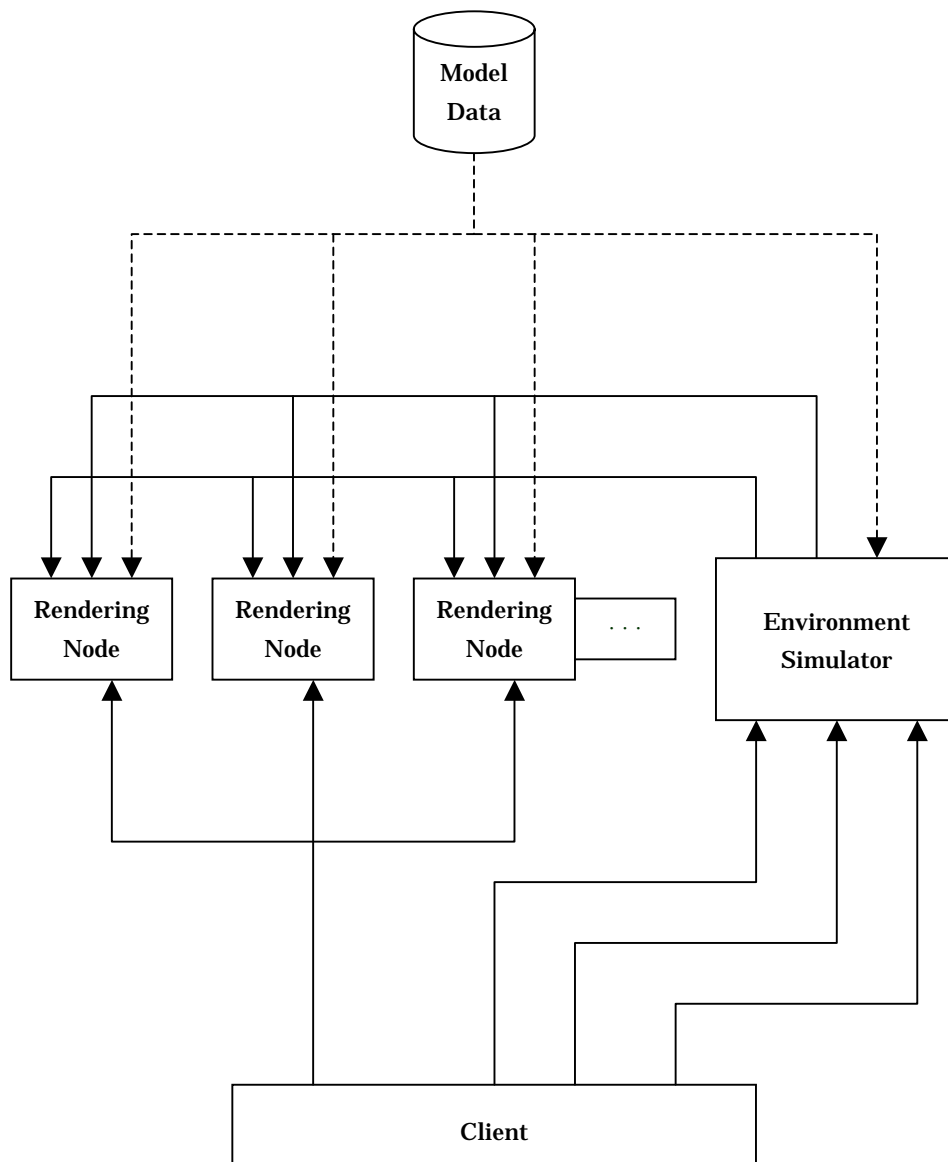


図5 画素変換



画像の分割設定を行う。(結果としてRenderingNodeへの参照が返る)  
 ロボットや能動指標を動作させる。  
 世界の状態の更新が各RenderingNodeに通知される  
 レンダリング要求  
 各RenderingNodeにレンダリング要求を出す。  
 各RenderingNodeから描画結果を取得する。

図 6 中心窩画像合成の流れ

とることにより効率化している。

#### 4.2 実装

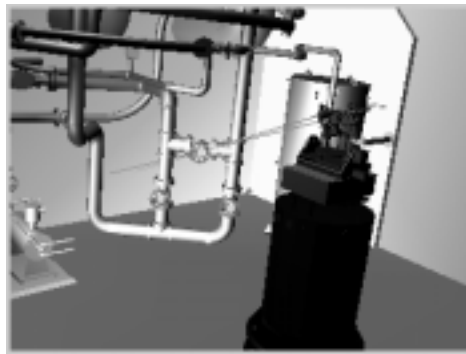
図 6 に実装の全体像を示す。視野を分割し、各々にプロセスを割り付け並列に処理する。さらに高速化のために、これらのプロセスを複数台の PC により処理し、結果を統合する。これまでの透視投影カメラの画像は環境サーバーにて合成していたが、中心窩画像の合成はレンダーサーバーにより行う。レンダーサーバーは環境サー

パー起動前に任意個起動しておく（図 6 の場合 3 個）。この際、レンダーサーバーの個数と、PC の個数は対応していなくてよい。クライアントから環境サーバーに対して中心窩画像カメラ取得の要求がくると、指定された数に視野を分割し、各視野を順繰りにレンダーサーバーに割り振る。したがって、視野の分割数とレンダーサーバーの数が異なってもよい。各レンダーサーバーは画像合成のために 3 次元世界の表現であるシーングラフを独立に保持している。シーングラフは動的に変化しているので同期したシーングラフから画像を合成できるメカニズムが重要である。本実装では、まずすべての PC の時刻を NTP ソフトにより同期をとる。環境サーバーにおける世界の状態の変化をイベントにて中心窩画像レンダリングノードに通知するときイベントの発生した絶対時刻を付加して通知する。さらにクライアントからの中心窩画像合成要求に応じて環境サーバーから各レンダーサーバーへ合成要求を出す際にも絶対時刻を付加する。これにより、各レンダーサーバーは要求時刻以前の世界の変化を的確に反映した画像を合成することができる。投影画像のレンダリングが完了したら中心窩画像変換プログラムを呼び出して、変換後画像を得る。環境サーバーは JAVA を用いて構築しているが、透視投影画像から中心窩画像への変換は多くの実数演算を要するので、中心窩画像変換プログラムは C++ で記述して高速化をはかった。環境サーバーからレンダーサーバーへの合成要求から各レンダーサーバーにおける合成変換はすべて並列に行われるので、描画に必要な時間は最も遅いレンダーサーバーの処理時間となる。レンダーサーバーで合成変換された分割した視野の画像は、環境サーバーを経由せずクライアントに直接わたされるので、無駄なコピーが発生しない。クライアントでの画像の受け取りをスレッドにより実装すれば、1 つの分割画像を取得する時間ですべての分割画像を取得できる。画像は分割のパラメーターとともに受け渡されるので、クライアントはこれらを容易に貼り合わせて 1 枚の画像に合成できる。

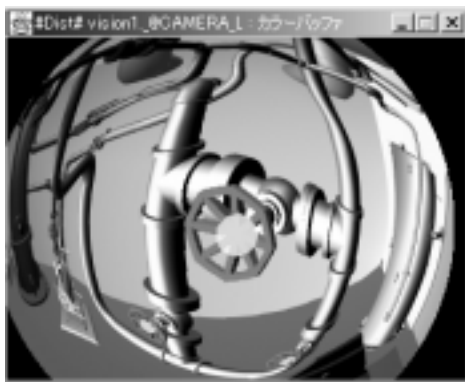
## 5. 実験

### 5.1 中心窩画像合成

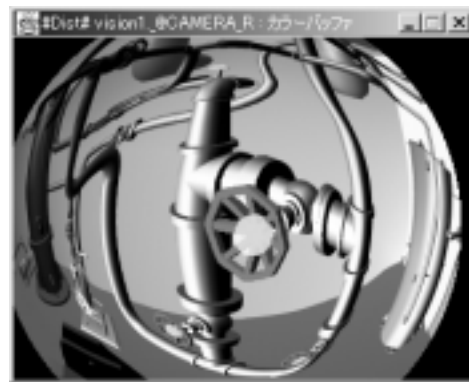
図 7 に環境サーバー中の ESCHeR の左右カメラの中心窩画像を合成した結果を示す。CPU が PentiumIII



(a)



(b)



(c)

図 7 中心窩画像合成例

800MHz Dual、グラフィックカードが SPECTRA8400 (NVIDIA GeForceII、AGP4 倍速) の PC4 台、それぞれで 1 つのレンダラーサーバーを走らせた。それらは 100Mbps イーサネットにより環境サーバーなどと通信する。視野分割は水平方向に 2、垂直方向に 1 としたので、左右画像について計 4 つの画像の合成が、各 PC で走る 4 つのレンダラーサーバーそれぞれに割り付けられた。各レンダラーサーバーで透視投影画像の生成とメインメモリへの転送に約 550msec、中心窩画像への変換に約 80msec を要している。クライアントから環境サーバーに中心窩画像合成要求を出してから、クライアントが各レンダラーサーバーから 4 つの歪み画像を受け取り左右の中心窩画像を表示するまで、合計約 700msec であり、1.4 フレーム / 秒程度の合成を実現できた。

## 5.2 注視制御の比較

次注視点決定の場を、透視投影視野から、中心窩視野にすることによる効果を調べるために、条件が同じ状況において、両者の次注視点決定過程を実験した例を図 8 に示した。図の上段、(a)から(c)が透視投影視野を用いた場合、下段(d)から(f)が中心窩視野を用いた場合であり、各々次注視点の決定に関する 4 つの画像(いずれも 360x240 画素の濃淡 8 ビット画像)を示した。左上が左実視野画像、左下が左仮想視野画像、右上が実視野を処理して抽出した反射的注視刺激の画像であり、これと仮想視野から高輝度として抽出した意図的注視刺激を融合した視野が右下である。反射的注視刺激は簡単な為 2 枚の入力画像(左上に示した入力画像とその数 100 ミリ秒後に入力した画像)の時間差分を閾値処理し、変化領域を輝度 255 として得ている。次注視点は右下の融合画像をラベル付け処理し微小領域を取り除いた後、領域の属性を用いて求めた優先度の高い順にステレオ対応の存在をチェックして決定した。

その結果、上段においては 2 番目の注視点に視点移動した(c)において、次に注視すべき特徴の候補が視野内に表れず、次の注視点を決定できなかった。一方、下段においては 2 番目の注視点に視点移動した(f)においても、次に注視すべき特徴の候補がいくつか視野内に存在した。さらに、実視野を処理した結果から、注視点から離れたところで起こっている実世界での事象(おもちゃの自動車の移動)も、次注視点の候補として表れている。

これらの実験から、とりまく環境のより広い範囲に対して注意を払い得るようになったことがわかる。

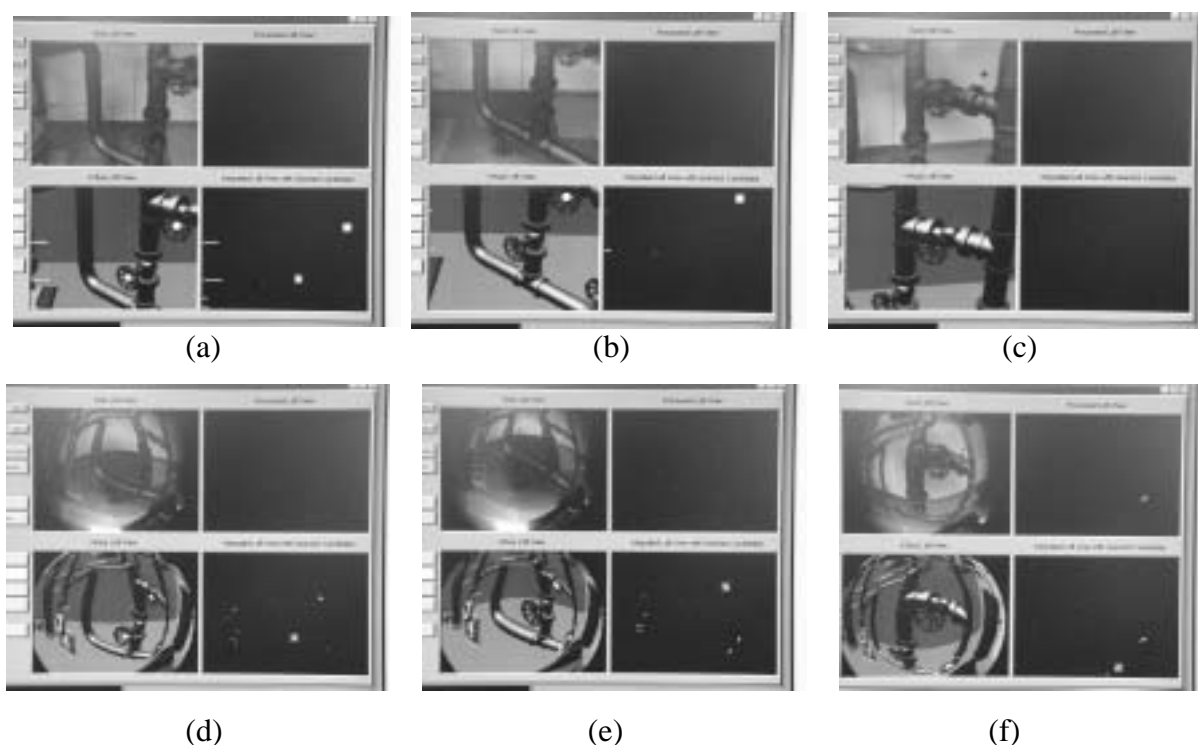


図 8 次注視点決定過程

## 6. まとめ

今回、拡充したシステムを用いて、今後実地的な注視制御の評価を行なう予定である。

**謝辞** 複数透視投影画像からの中心窩投影画像への分割変換の実装についてソリッドレイ研究所に、環境サーバー全般の実装、今回特に中心窩画像合成機能の組み込みについて（株）カーネルに尽力いただいた。本研究は文科省の委託による原子力基盤クロスオーバー研究のもとでおこなった。

### 参考文献

- [1] 喜多伸之：反射的・意図的注視制御の統合的な実現、信学論 D-II, Vol. J84-D-II, No.8, 1701-1709(2001).
- [2] Y. Kuniyoshi, N. Kita, S. Rougeaux, T. Suehiro: Active Stereo Vision System with Foveated Wide Angle Lenses, Proc. of Second Asian Conference on Computer Vision (ACCV95), pp.359- -363, 1995.