コンピュータビジョンと
イメージメディア 127－2
（2001．5．10）

Hyper Frame Vision: 大容量フレームメモリを用いた常動物体の実時間3次元追跡

角 保志† 石山 豊‡ 富田文明†

†産業技術総合研究所
つくば市梅園 1-1-1 中央第2
{y.sumi,f.tomita}@aist.go.jp

‡スタンレー電気（株）技術研究所
横浜市青葉区荏田西 1-3-1
y.ishiyama@yrd.stanley.co.jp

あらまし 剛体運動をしている物体の位置姿勢 (6DOF) を、初期位置に関する情報を外部から与えることなく、実時間で認識・追跡するシステムについて述べる。本システムにより、常に動いている物体（常動物体）の3次元追跡が可能となる。また、処理の途中で物体を見失っても回復できる。本システムは、基本的に任意形状の物体を扱うことができ、隠れや異物体の混在に対しても比較的頑健である。本システムは、3眼ステレオカメラと数百Mバイトの総容量を持つフレームメモリで構成される。フレームメモリに蓄積される時系列ステレオ画像を、物体認識モジュールと運動追跡モジュールが並列的に処理する。実験結果から、本システムの有効性を示す。

キーワード 3次元運動追跡、3次元物体認識、実時間処理、大容量フレームメモリ、ステレオ視

Hyper Frame Vision: Real-Time 3D Tracking of Moving Objects Using Massive Frame Memory

Yasushi Sumi† Yutaka Ishiyama‡ Fumiaki Tomita†

†National Institute of Advanced Industrial
Science and Technology (AIST)
AIST Tsukuba Central 2, Tsukuba
305-8568 Japan

‡Stanley Electric Co., Ltd. R&D
1-3-1 Eda-nishi, Aoba-ku, Yokohama
225-0014 Japan

Abstract We propose a new system for real-time 6-DOF rigid motion tracking of an object without initial positioning of the object. This system permits tracking of a non-stop moving object and retention of the tracking capabilities even if the target object has been lost during the course of the tracking. In addition, the system can deal with any shape of object. The object can exist together with other objects and can be partially occluded by them. The system hardware consists of calibrated trinocular stereo cameras and hundreds of MB of frame memory. The object recognition task and the motion tracking task work in parallel to process time-sequential stereo images which have been stored to the frame memory. The usefulness of the system is demonstrated by experimental results.

Key Words 3D motion tracking, 3D object recognition, Real-time processing, Massive frame memory, Stereo Vision

1 はじめに

本報告では、3次元物体の運動追跡、すなわち、剛体運動をしている既知物体の位置姿勢 (6DOF) を、実時間で追跡する問題を扱う。3次元運動追跡はコンピュータビジョン研究の重要なテーマの一つであり、様々な研究がなされてきた。しかしながら、これまで、その応用分野はごく限られたものであった。

3次元運動追跡は、時系列画像データのフレーム間における物体の位置姿勢の変位を検出する手法である [1, 2, 3, 4, 5, 6, 7] 。基本的に、フレーム間変位をビデオレート（30fps）以上の処理速度で算出することにより、物体の実時間追跡を可能とする。しかし、これまで提案されてきた手法では、追跡の起点となる物体の「初期位置」をどのようにして求めるかということについて、ほとんど考慮されていなかったため、対象物体の初期位置が既知であるような環境でしか利用できなかった。

初期位置未知の物体を追跡するひとつの方法は、3次元物体認識 [8, 9] を運動追跡の前処理として組み合わせることである。これにより、物体の初期位置を自動検出することが可能となる。しかしながら、3次元物体認識は、シーン全体から物体の位置姿勢を検出する手法であり、M, N をそれぞれシーンとモデルの特徴数とすると、一般に $O(MN) \sim O(M^3N^3)$ の計算量を要するため、ビデオレートでの処理は困難である。つまり、物体認識を組み合わせたシステムでも、常に移動している物体（常動物体）は扱えなかった。

また、運動追跡処理においては、隠れ等の影響により、対象物体を追跡中に “見失う” ことが起こり得る。しかし、従来の手法では、一度見失った物体を再び追跡することは困難であった。

本報告では、3次元物体認識・運動追跡のための新しいビジョンシステムアーキテクチャである、*Hyper Frame Vision* を提案する。Hyper Frame Vision により、常動物体の位置姿勢を、実時間で認識・追跡することが可能となる。物体の初期位置に関する情報を必要としないので、処理の途中で物体を見失っても回復可能である。

Hyper Frame Vision システムは、ステレオカメラと数百Mバイトの総容量を持つフレームメモリから構成される。従来の実時間処理システムが、過去のシーンやオブジェクトに関する情報を用いて、「現在の」フレームを高速処理することを基本としているのに対し、Hyper Frame Vision では、大容量フレームメモリに動的に蓄積されていく時系列画像データを並列的に処理することを基本とする。これにより、3次元物体の初期位置推定のような「重い」処理を、運動追跡処理に組み合わせることが可能となる。

物体認識と運動追跡は、3次元ビジョンシステム VVV[10] の処理モジュールを利用する。VVV システムは、基本的に任意形状の物体を扱うことができ、隠れや異物体の混在に対しても比較的頑健である。

以下、VVV の物体認識と運動追跡の各モジュールについて概観した後、Hyper Frame Vision の構成と処理の流れについて述べ、実験結果から、常動物体の追跡が可能であること、また、一時的に物体を見失っても追跡を回復できることを示す。

2 3次元物体の認識と追跡

2.1 3次元物体認識

3次元物体認識は、シーン中から対象物体を検出し、その位置姿勢を求める処理である。Hyper Frame Vision では，セグメントベーストステレオを用いたモデルベースト物体認識手法 [11, 12] により，物体の位置姿勢 (6DOF) を認識する。

物体認識処理は，シーンの3次元輪郭線データの復元 [13, 14, 15]、データとモデルの特徴点照合に基づく物体位置に関する候補の生成、モデル代表点の最近傍データ点探索と最小自乗法の繰り返し処理に基づく候補の検証・微調整からなる.

この手法は、基本的に任意形状の物体を扱うことができ、隠れや異物体の混在に対しても比較的頑健であるという特長がある。認識の処理量は $O(MN)$ であり、単純なシーンでは、画像入力から認識終了まで1秒程度* で処理できる。また、認識精度は、カメラ設定に依存するが、カメラ間距離 30cm、対象までの距離 2m の設定で、25mm 相当のレンズを用いた実験では、1mm 以下程度の精度が得られている [16]。

物体認識モジュールは、認識結果として、物体モデルを world 座標系に移動させる 4×4 座標変換行列 $\boldsymbol{T}$ を出力する。このとき、認識結果を以下のように評価する。$\boldsymbol{T}$ に基づき、物体モデルを3次元シーン中に配置したとき、モデル輪郭線上の代表点のうち、シーン中に対応点を持つ点の割合

$$S_{rec} = \frac{n_{rec}}{n_{vis}} \tag{1}$$

を認識結果の評価値とする。ここで、n_{vis} はモデル代表点のうちカメラ位置から観測可能な点数、n_{rec} はそのうち近傍に3次元データ輪郭線が存在する点数であり、ともに物体認識の処理過程で導出される。

図1(a) に入力3眼ステレオ画像（640×480 pixels, 256 gray-levels）と物体モデルの例を示す。同図 (b)

* PentiumIII 866MHz

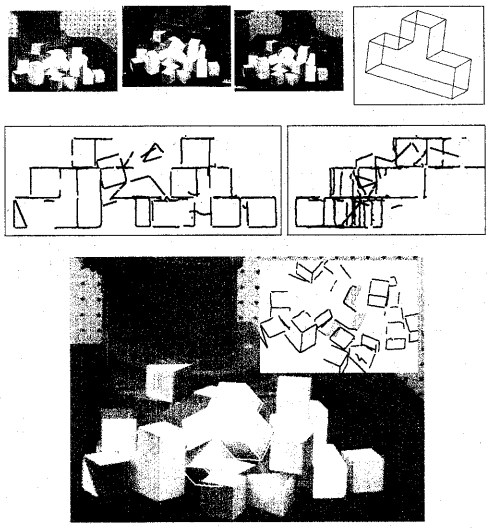

図 1: (a) 上：入力 3 眼ステレオ画像と物体モデル、(b) 中：復元された 3 次元輪郭線（正面図、側面図）、(c) 下：物体認識結果

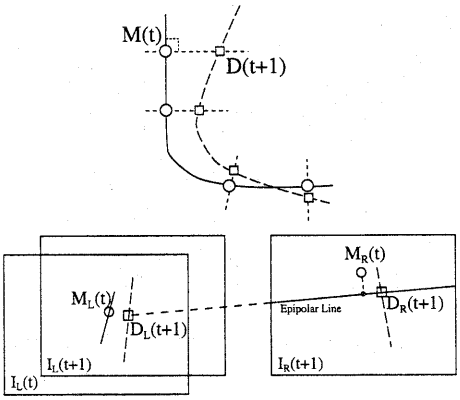

図 2: (a) 上：対応エッジ点探索、(b) 下：ステレオ対応探索

は，セグメントベーストステレオによって復元された 3 次元輪郭線の正面図と側面図である。(c) は推定された物体の位置姿勢を示している.

2.2 3 次元運動追跡

物体の運動追跡は、ある時刻 t における物体の特徴点の、時刻 $t+\Delta t$ への変位を求める処理の繰り返しである。Hyper Frame Vision では、ステレオビジョンを用いたモデルベースト運動追跡手法 [17, 18] により、物体のフレーム間変位 (6DOF) を実時間で算出する。ここで、物体が 3 次元剛体運動をし、物体のフレーム間変位が微小であることを仮定する。

運動追跡のフレーム処理は、モデル代表点の対応データ点探索と最小自乗法による位置推定からなる。対応データ点探索は、以下のようにして行なう。まず、図 2(a) に示すように、時刻 $t+1$ におけるフレームの基準画像 $I_L(t+1)$ に投影した、時刻 t のモデル代表点 $M_L(t)$ の近傍において、対応点 $D_L(t+1)$ を探索する（対応エッジ点探索）。次に、同図 (b) に示すように、対応画像 $I_R(t+1)$ に投影したモデル代表点 $M_R(t+1)$ から、$D_L(t+1)$ のステレオ対応点 $D_R(t+1)$ をエピポーラ拘束を用いて探索する（ステレオ対応探索）。

得られた $\{D_L, D_R\}$ から、対応データ点の 3 次元座標を算出する。この対応点探索処理を、全ての観測可能なモデル代表点について実行し、モデル代表点と対応データ点の距離が最小となる変位を最小自乗法によって算出する。上記の対応データ点探索と位置推定処理を繰り返すことにより、時刻 $t+1$ におけるモデル代表点位置が推定できる。

この手法は、前述の物体認識と同様、基本的に任意の形状を扱うことができ、隠れなどにも比較的頑健である。追跡精度も物体認識精度とほぼ同等である。処理時間は、物体モデルの複雑さに依存するが、フレームあたり数ミリ〜100 ミリ秒程度が実現できている。

運動追跡モジュールは、追跡結果として、物体のフレーム毎の位置姿勢 $\boldsymbol{T}(t)$ を出力する。このとき、追跡結果の評価値を、

$$S_{tr}(t) = \frac{n_{tr}(t)}{n_{vis}(t)} \tag{2}$$

とする。ここで、$n_{vis}(t)$ は、フレーム $F(t)$ での対応点探索において、カメラ位置から観測可能なモデル代表点数であり、$n_{tr}(t)$ はそのうち対応点探索に成功した点数である。

3 Hyper Frame Vision

3.1 Hyper Frame Vision の構成

Hyper Frame Vision は、図 3 に示すように、ステレオカメラと、数百Mバイト以上の総容量を持ち、数百フレームのステレオ画像を蓄積可能な大容量フレームメモリから構成される。入力されるステレオ

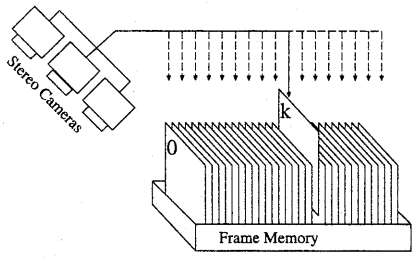

図 3: Hyper Frame Vision のシステム構成

画像は、フレームメモリの先頭から順に蓄積されていく。入力された画像データ量がフレームメモリの総容量を超えると、フレームメモリの先頭から画像データが上書きされ、これが繰り返される。

フレームメモリに蓄積される時系列ステレオ画像に対し、物体認識と運動物体追跡の処理モジュールが、並列的にアクセスする。最大フレーム数を N とし、k 番めのフレーム $F(k)$ に画像入力中とすると、各ビジョンモジュールは、データ書き込み中の $F(k)$ を除き、$F(k-1)$ を最新とする、過去 $N-1$ フレームの時系列ステレオ画像の組に対してアクセスすることが可能となる。

3.2 基本プロセス

図 4 に Hyper Frame Vision の基本プロセスを示した。図中、横軸は時間の経過を表している。また、図の上段はフレームメモリへの画像データ入力を、中段は物体認識モジュールと運動追跡モジュールの動作を、下段はシステムのモード遷移をそれぞれ示している。

図に示したように、時刻 $t_{start}=0$ に処理を開始したとき、フレームメモリの第 0 フレーム $F(0)$ に画像データが入力される。しかし、各モジュールは、まだ処理を開始することができず、「待ち」の状態にある。図では、待ち状態のモジュールを細線の帯で示している。

t_f 秒後に $F(0)$ への画像データ蓄積が完了すると、物体認識モジュールは、$F(0)$ に対して処理を開始する。ここで、t_f はフレーム周期（一般に $t_f=1/30$）である。処理中のモジュールは太線の帯で示している。また、破線矢印は、各モジュールからフレームメモリへのアクセスを、太線帯中の数字は処理中のフレーム番号をそれぞれ示している。

物体認識処理は、前述のようにビデオレートでの処理は困難である。ここで、認識に要した処理時間を t_{rec} とすると、時刻 t_{start} から t_f+t_{rec} まで、システムは物体の位置姿勢についての情報を持っていなかったことになる。この状態を “LOST モード” と呼ぶことにする。

運動追跡モジュールは、物体認識モジュールの処理結果を受けて、図 4 に示すように、時刻 t_f+t_{rec} に処理を開始する。図中実線矢印はモジュール間のデータ授受を表す。運動追跡モジュールは、物体認識モジュールによって算出された、$F(0)$ における物体の位置姿勢 $\boldsymbol{T}(0)$ を基準として、$F(i|\ i=1,2,\ldots)$ における物体の位置姿勢 $\boldsymbol{T}(i)$ を順次算出していく。

運動追跡の開始時点では、システムが追跡している物体の位置姿勢は、現在時刻よりも t_f+t_{rec} 秒前のデータに基づいている。このように、物体の過去の位置姿勢を追跡している状態を “LOCK-ON モード” と呼ぶことにする。LOCK-ON モードでは、画像データ入力中のフレーム番号 n_{in} と運動追跡の処理対象となるフレーム番号 n_{tr} は、時刻 t において、それぞれ、

$$n_{in}=\frac{t}{t_f},\quad n_{tr}=\frac{t-(t_f+t_{rec})}{t_{tr}} \tag{3}$$

（少数以下繰り上げ）となる。ここで、t_{tr} は運動追跡処理におけるフレームあたりの平均処理時間である。

n_{tr} と n_{in} の関係は以下のようになる。すなわち、運動追跡の開始時刻 t_f+t_{rec} には、確実に $n_{tr}<n_{in}$ であるが、

$$t_{tr}<t_f \tag{4}$$

であれば、n_{tr} は時間の経過とともに徐々に n_{in} に近づき、時刻

$$t_{catchup}=\frac{t_f+t_{rec}}{t_f-t_{tr}}t_f \tag{5}$$

において、$n_{tr}=n_{in}$ となる（図 4 では、$n_{in}=k$）。ところが、このとき $F(n_{in})$ はデータの書き込み中なので、運動追跡モジュールからのアクセスは lock されている。したがって、$F(n_{in})$ への書き込みが終わるまで、追跡モジュールは「待ち」になる。この状態を “TRACKING モード” と呼ぶことにする。TRACKING モードでは、物体の「現在位置」が追跡されている。

3.3 追跡失敗からの回復

運動追跡処理においては、対象物体を追跡中に見失うことが起こり得る。しかし、Hyper Frame Vision では、物体の初期位置に関する情報を必要としないので、処理の途中で物体を見失っても回復可能である。

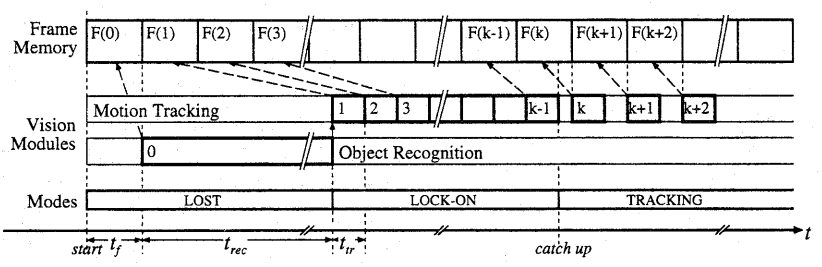

図 4: Hyper Frame Vision の基本プロセス

システムが LOCK-ON または TRACKING モードにあるとき、あるフレーム $F(i)$ での運動追跡結果の評価値 $S_{tr}(i)$ が低く、閾値 θ_S に対して

$$S_{tr} < \theta_S \tag{6}$$

であれば、図 5 に示すように、システムは物体を「見失ったかもしれない」とみなして LOST モードに移行する。

LOST モードからの回復のために、2 つの方法が考えられる。ひとつは、LOST モードになってから、物体認識モジュールを起動する方法である。図 5(a) のように、フレーム $F(i)$ に対する追跡処理のスコア $S_{tr}(i)$ によって LOST モードに移行したとき、物体認識モジュールを $F(i)$ に対して実行する。$F(i)$ に対する認識処理が終了したとき、認識のスコア $S_{rec}(i)$ と $S_{tr}(i)$ とを比較し、

$$S_{rec} > \omega S_{tr} \tag{7}$$

ならば、LOCK-ON モードに移行する。ここで ω は重み係数である。そうでなければ、$F(i)$ では認識不能であったとみなし、図のように、最新のフレーム $F(j)$ に対して改めて認識を実行する。

もうひとつは、図 5(b) のように、モードにかかわらず、認識モジュールを常に起動しておく方法である。あるフレームに対する認識処理が終了する毎に条件 (7) の判定を行い、認識結果が優れていれば直ちに LOCK-ON モードに移行する。この方法では、(a) の方法と比べて CPU パワーを浪費してしまう場合もあるが、回復に要する時間は短くなる可能性が高い。また、異なるフレームに対する認識処理は互いに独立しているので、多数の CPU を持つ並列処理環境で並列処理を行なうことができれば、パフォーマンスを低下させることなく、さらに回復時間を短縮することが可能となる。

ところで、運動追跡モジュールは、LOST モードになっても追跡処理を中止しない。なぜなら、LOST モードに移行した場合でも、必ずしも物体を見失っ

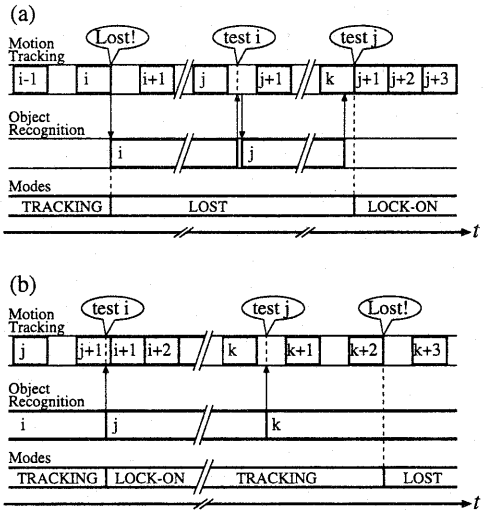

図 5: 追跡失敗からの回復プロセス

ているとは限らず、部分隠れなどの影響で、一時的に評価値が低くなっているだけかもしれないからである。隠れの影響が少なくなり、S_{tr} が閾値を超えれば、再び TRACKING モードに回復できる。

4 実験

4.1 Hyper Frame Vision の実装

Hyper Frame Vision による常動物体の追跡、および追跡失敗からの回復の実験を行った。システムのハードウェア構成は以下の通りである。

PC: IBM-PC 互換, Dual PentiumIII 866MHz, Linux/SMP

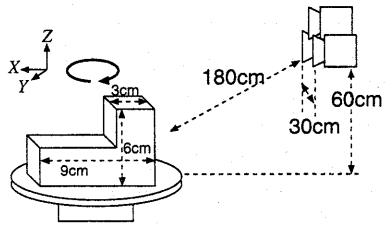

図 6: 回転テーブルを用いた運動追跡実験

画像入力ボード: Microtechnica MTPCI-CD
ステレオカメラ: Sony XC-7500 × 3, Canon J10X10R-II 電動ズームレンズ（25mm レンズ相当に設定）使用

ここで、画像入力ボードとして用いた MTPCI-CD は、フレームメモリを 768MB に増設することで、RGB 画像（640 × 480 pixels, 8 × 3 bits）を最大 873 フレームまで連続して蓄積することができる。本システムでは、各フレームの RGB プレーンに 3 台のカメラからの画像をそれぞれ入力する。

システムのソフトウェアは、物体認識と運動追跡のための各ビジョンモジュールと、フレームメモリへの画像入力制御モジュールの、3 つのモジュールから構成される。ここで、画像入力制御モジュールは、フレームメモリへの画像データ入力を監視し、フレームメモリへの循環的な画像データ書き込みを保証する。各モジュールは C 言語を用いて記述した。モジュールの並列動作は、POSIX スレッドによる並行プログラミングによって実現した。

4.2 実験結果

回転テーブルを用いた運動追跡実験結果を示す。図 6 に示すように、カメラ間距離を約 30cm、カメラから対象までの距離を約 1.8m とし、回転テーブルの面を world 座標系の XY 平面と平行に設置した。追跡対象として、図に示した L 型の積み木を用いた。回転テーブルを毎秒約 18deg. の速度で常時回転させ、A) 隠れのない場合、B) 周期的に全隠れが発生する場合のそれぞれについて処理を行った。

図 7 には、追跡実験 A,B で得られた対象物体の位置姿勢の変位に基づき、物体上のある 1 点の X 座標に関する軌跡を示した。ここで、物体が正しく実時間追跡されていれば、軌跡は約 20 秒周期の正弦曲線を描くはずである。図 8 には、図 7 に点線で示した時刻における、物体の 3 次元位置姿勢推定結果を、入力ステレオ画像の左画像上に投影して示した。

実験 A において、A0 は画像入力がスタートした時刻を示している。A1 はフレーム $F(0)$ に対する物体認識が終了した時刻である。ここで、A0 から A1 までが物体認識に要した時間 t_{rec} であり、この実験では約 2.3 秒であった。A1 以後、LOCK-ON モードにおける追跡モジュールの処理が実行され、時刻 A2 に追跡処理が画像入力に追い付いている。ここで、A2 での入力フレームは $F(99)$ であり、A1 から A2 までに要した時間は約 0.73 秒、1 フレームあたりの平均処理時間 t_{tr} は約 7.4 ミリ秒であった。A2 以後は、TRACKING モードに移行し、実時間追跡が実行されていることがわかる。なお、$t = 0$ から A0 までは、システムのハードウェア初期化等の処理に要した時間である。

実験 B では、対象物体のほぼ全体が一時的に隠され、追跡が失敗するようにした。物体が隠される時間は、約 20 秒の回転周期のうち、約 4 秒とした。3.3 節で述べた方法のうち、今回は、方法 (a) による実験を行った。式 (6) に示した追跡失敗を判別するための閾値は、$\theta_S = 0.5$ とした。ノイズ等の影響を考慮し、評価値 S_{tr} が 3 フレーム連続して閾値を下回った場合、LOST モードに移行することとした。

図 7 から分るように、実験 B においても、処理開始から 6 秒後程度までは、実験 A とほぼ同様の軌跡が得られているが、隠れのため時刻 B3 で追跡失敗と判定され、LOST モードに移行している。このとき、物体認識モジュールは、直ちに B3 の入力フレーム $F(177)$ に対する認識処理を開始し、時刻 B4 に処理を終了している。しかし、その認識結果は条件 (7) を満たさなかったため、改めて時刻 B4 の入力 $F(322)$ に対する認識処理を開始している。その結果、時刻 B5 に LOCK-ON モードに移行し、追跡失敗から回復することができた。ここで、$F(177)$ に対する最初の認識に失敗したのは、図 8 からも分かるように、物体のほぼ全体が隠されていたためである。

以上は、等速回転運動に対する実験結果であったが、対象物体を人間の手で動かす実験も行い、本システムが物体の任意動作についても対応できることを確認した。本システムの運動追跡モジュールはフレーム間の物体の変位が微小であることを仮定しているため、人間の手によって物体を急速に動かせば追跡に失敗する。しかし、このような場合でも、実験 B と同様の過程を経て回復することが可能であった。

4.3 考察

物体認識のための処理時間の上限は、フレームメモリ容量とフレーム周期 t_f に拘束される。フレーム

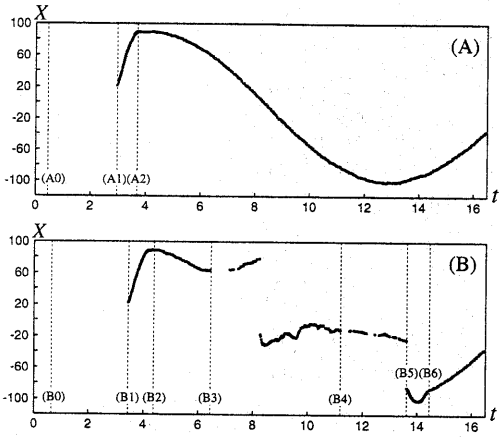

図 7: 運動追跡結果: (A) 隠れのない場合、(B) 全隠れのある場合

メモリのフレーム数を N とすると、フレーム $F(0)$ に対する認識処理を開始して Nt_f 秒後には、フレーム $F(1)$ の画像データが $F(N+1)$ のデータで上書きされてしまうため、たとえ $F(0)$ に対する物体認識が正常に終了しても、運動追跡処理は破綻してしまう。従って、物体認識時間 t_{rec} は、

$$t_{rec} \leq Nt_f \tag{8}$$

でなければならない。

実験で使用したフレームメモリでは、$Nt_f = 873 \times 1/30 = 29.1$ である。今回の実験では、$t_{rec} \simeq 2.3$ であり、十分に条件を満たしている。

しかしながら、物体認識モジュールの処理量は、前述のように $O(MN)$ であるため、シーンとモデルが複雑になると、条件 (8) を満たすことができなくなる。このような場合への対処としては、PC クラスタなどを用いた並列処理 [19] により、物体認識の処理時間を短縮することなどが考えられる。

運動追跡モジュールの処理時間 t_{tr} は、フレーム周期 t_f によって拘束され、条件 (4) が成り立つときのみ、正常な処理を行える。もし、$t_{tr} = t_f$ であれば、追跡処理は画像入力に「追い付く」ことができず、LOCK-ON モードが永久に続くことになる。さらに、$t_{tr} > t_f$ であれば、追跡処理の開始から $n_{lap}t_{tr}$ 秒後には、追跡処理対象フレーム $F(n_{lap})$ の画像データが $F(n_{lap}+N)$ のデータで上書きされるため、追跡処理は破綻してしまう。ここで、$n_{lap} = \frac{Nt_f}{t_{tr}-t_f}$ である。

今回の実験では、前述のように、$t_{tr} \simeq 0.0074$ であり、条件を満たしている。

運動追跡モジュールの処理量は $O(M)$ であるが、やはり非常に複雑な物体モデルを扱う場合には、条件 (4) を満たすことが困難となる。特に、自由曲面体追跡 [18] は、現状ではほとんどの場合ビデオレートでは処理できない。このような場合に対処するために、対応点探索に使用するモデルの特徴点数を動的に増減させる、時間あたりの処理フレーム数を変化させるなどの手法が考えられる。このようなシステムの開発は今後の課題である。

5 むすび

3 次元物体認識・運動追跡のためのビジョンシステムアーキテクチャ Hyper Frame Vision を提案した。Hyper Frame Vision により、常動物体の実時間追跡が可能になることを実験により示した。また、運動追跡処理において避けられない、追跡失敗からの回復が可能になることを示した。

本手法を用いることで、これまではごく限られていた 3 次元運動追跡の応用範囲を拡大することが可能であろうと考えている。

Hyper Frame Vision は、3 次元物体の運動追跡だけなく、様々な実時間ビジョンに適用できる。今後は、Hyper Frame Vision を用いたヒューマンインタフェースや ITS などのシステム開発も考えて行きたい。

謝辞

日頃有益な助言と討論を頂く，産総研 3 次元視覚システム研究グループと VVV 研究会の各位に感謝致します.

参考文献

[1] R. S. Stephens: Real-time 3D object tracking, *Image and Vision Computing*, Vol. 8, No. 1, pp. 91–96 (1990).

[2] D. G. Lowe: Robust model-based motion tracking through the integration of search and estimation, *Int. J. of Computer Vision*, Vol. 8, No. 2, pp. 113–122 (1992).

[3] D. Gennery: Visual tracking of known three-dimensional objects, *Int. J. of Computer Vision*, Vol. 7, No. 3, pp. 243–270 (1992).

[4] H. Kollnig and H.-H. Nagel: 3D pose estimation by fitting image gradients directly polyhedral models, *Proc. ICCV'95*, pp. 569–574 (1995).

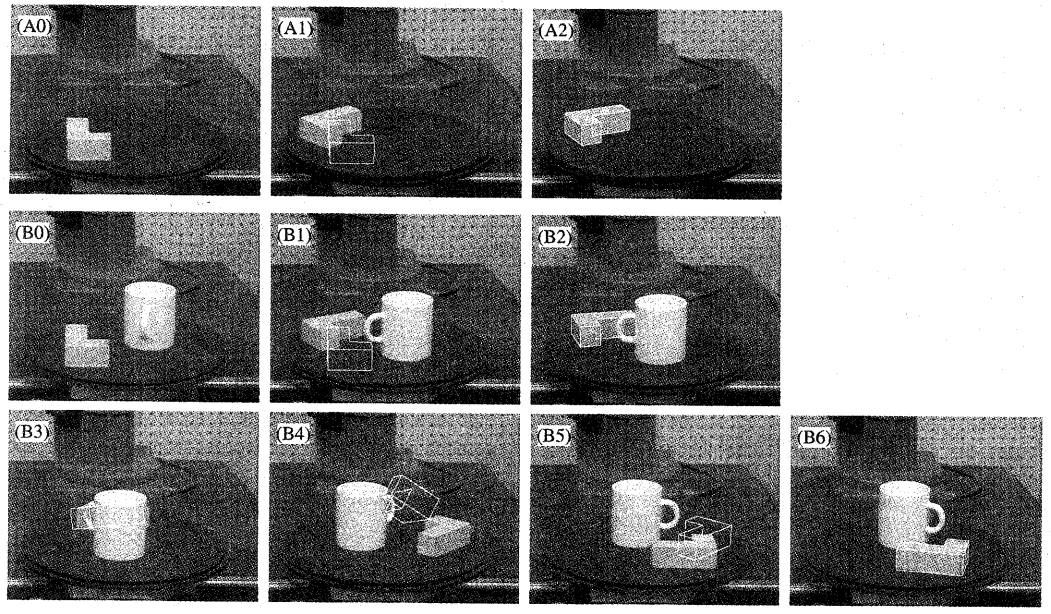

図 8: 運動追跡結果: (A0) Frame-0 [LOST], (A1) Frame-77 [LOCK-ON], (A2) Frame-99 [TRACKING], (B0) Frame-0 [LOST], (B1) Frame-90 [LOCK-ON], (B2) Frame-113 [TRACKING], (B3) Frame-177 [LOST], (B4) Frame-322 [LOST], (B5) Frame-394 [LOCK-ON], (B6) Frame-418 [TRACKING]

[5] 日浦, 山口, 佐藤, 井口：動距離画像の計測と生成による任意形状物体の実時間追跡, 信学論 (D-II), Vol. J80-D-II, No. 6, pp. 1539–1546 (1997).

[6] E. Marchand, P. Bouthemy, F. Chaumette and V. Moreau: Robust real-time visual tracking using a 2D-3D model-based approach, *Proc. ICCV'99*, Vol. I, pp. 262–268 (1999).

[7] M. Tonko and H.-H. Nagel: Model-based stereo-tracking of non-polyhedral objects for automatic disassembly experiments, *Int. J. of Computer Vision*, Vol. 37, No. 1, pp. 99–118 (2000).

[8] W. E. L. Grimson: *Object recognition by computer: The role of geometric constraints*, The MIT Press (1990).

[9] F. Arman and J. K. Aggarwal: Model-based object recognition in dense-range images — A review, *ACM Computing Surveys*, Vol. 25, No. 1, pp. 5–43 (1993).

[10] 富田ほか：3 次元視覚システム VVV 研究開発 —概要—, 情処研報, CVIM-109-1, pp. 1–8 (1998).

[11] 角, 富田：ステレオビジョンによる 3 次元物体の認識, 信学論 (D-II), Vol. J80-D-II, No. 5, pp. 1105–1112 (1997).

[12] 角, 河井, 吉見, 富田：セグメントベーストステレオによる自由曲面体の認識, 信学論 (D-II), Vol. J81-D-II, No. 2, pp. 285–292 (1998).

[13] 石山, 角, 河井, 植芝, 富田：セグメントベーストステレオにおける対応候補探索, 映像情報メディア学会誌, Vol. 52, No. 5, pp. 723–728 (1998).

[14] 河井, 植芝, 石山, 角, 富田：セグメントベーストステレオにおける連結性に基づく対応評価, 情処学論, Vol. 40, No. 8, pp. 3219–3229 (1999).

[15] T. Ueshiba, Y. Kawai, Y. Ishiyama, Y. Sumi and F. Tomita: An efficient matching algorithm for segment-based stereo vision using dynamic programming technique, *Proc. MVA'98*, pp. 61–64 (1998).

[16] 角, 河井, 吉見, 富田：ステレオビジョンシステムのためのモデルベースト物体認識, 電気学会システム・制御研資, SC-00-15, pp.25–30 (2000).

[17] 石山, 角, 富田：ステレオビジョンによる三次元物体の三次元運動追跡, 日本ロボット学会誌, Vol. 18, No. 2, pp. 213–220 (2000).

[18] 角, 石山, 富田：ステレオビジョンシステムのためのモデルベーストアプローチによる自由曲面物体の実時間運動追跡, 画像の認識・理解シンポジウム (MIRU2000) 講演論文集, I, pp. 39–44 (2000).

[19] 大上, 杉本, 北村, 角, 富田：ネットワーク型並列計算環境における物体認識, 信学論 (D-II), Vol. J82-D-II, No. 12, pp. 2307–2315 (1999).