
2次元マトリックスコードを利用した拡張現実感の構成手法

Augmented Reality using the 2D matrix code

暦本 純一*

Summary. This paper introduces a new technique for augmented reality that identifies real world objects and estimates their coordinate systems simultaneously. The 2D matrix code used in our method is a square shaped barcode that can identify a large number of objects. It is also used as a landmark to register information on the real world images. Since matrix codes are printable, it is virtually costless to produce and attach codes on various kinds of real world objects. We have developed an AR system based on this method, and demonstrated several potential applications.

1 はじめに

拡張現実感 (augmented reality, AR) は、現実世界をコンピュータによって強化しようというインタフェースの総称である。特に、現実の情報とコンピュータ情報とを合成表示することで、現実のみ、あるいは仮想現実のみでは得られないような種々の新しい応用が可能になる。たとえば機械の保守作業を支援するために、操作すべき部品の位置をハイライト表示したり、機器の内部の情報を表示することが考えられる。建築への応用としては、まだ建設されていないビルを現実の風景にオーバーレイしたり、内装の変更をシミュレートして、室内の実映像に合成するシステムなどが考えられる。

これらのシステムの構築において、現実の空間の持つ座標系に整合したコンピュータ映像の生成は重要な基盤技術である。従来の拡張現実感システムでは、磁気センサや超音波センサなどの3次元センサを利用してしたが、スケラビリティ(計測可能な範囲)、精度の点で問題があった。近年、ビデオカメラから得られる映像を解析して実世界の座標系を得る方法が注目されている。たとえば実世界オブジェクトの表面に複数個のマーカを添付しておき、それをビデオカメラで認識する。このマーカの実世界での位置が既知であるとする、カメラの画像座標上に投影されたマーカの位置から、ビデオカメラの実世界座標系での位置と姿勢を逆算することができる。いったんカメラの視点位置が特定されてしまえば、その情報を利用してCG映像等を生成し、元のビデオ映像とオーバーレイすればよい。この方式(マーカ方式)の利点としては、3次元センサのような特殊な装置を必要とせずに画像のオーバーレイが可能で、マーカとの相対的な位置関係を計測するので、移動する物体に対し

* Jun Rekimoto. (株) ソニーコンピュータサイエンス研究所

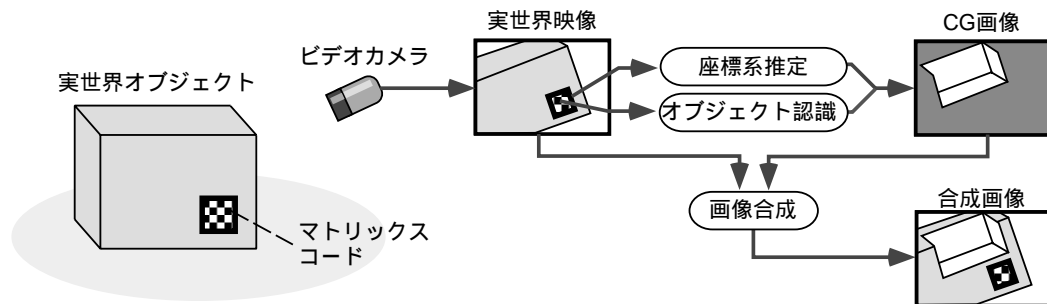


図 1. 提案方式の概要

でも適用できる点、ビデオカメラをビデオシースルー表示にも併用できる点、などがある。

また、他のセンサ方式と比較して、見かけ上の位置整合精度が高くなるのもマーカー方式の長所のひとつである。AR システムでの達成目標は、画面上で実画像とコンピュータ画像の高精度のマッチングをとることであって、カメラの位置情報は途中段階で必要とされるデータにすぎない。すなわち、画面上での「つじつま」を合わせることが目的となる。従来の手法では、カメラ映像と無関係に計測された 3 次元位置情報を利用して、つまり 3 次元位置 カメラ位置 CG 映像合成という順で処理が行われる。この場合、3 次元の計測誤差に加えて、ビデオ映像と 3 次元データの時間的なずれ、3 次元センサとカメラとの位置関係なども位置合わせの「ずれ」を増幅する要因となる。また、カメラの位置だけではなく、実世界オブジェクトの現実世界での位置、姿勢も併せて計測しておかなければならず、ここでも誤差が拡大される。マーカー方式では、画面上でのマッチング精度を尺度としてカメラ位置を推定するので、これらの問題は原理的に存在しない。

一方、マーカー方式の技術課題としては、マーカーそのものを安定的に認識することの難しさが挙げられる。たとえば色を使ったマーカーでは、背景の実映像に似た色があれば誤認識してしまう。また、複数の対象物を取り扱うことが難しい。ある部品に添付されているマーカーと、別の部品に添付されているマーカーを識別できないと、それぞれの部品に対して適切な情報を生成してオーバーレイすることができない。

本論文では、これらの問題を解決するために、2 次元マトリックスコードをマーカーとする方式を提案し、その認識アルゴリズムについて解説する。2 次元マトリックスコードはバーコードの一種であり、通常の紙に印刷可能な矩形のコードである。コードの矩形輪郭がカメラ画像表面に投影された形状を解析して、カメラの位置と姿勢を得ることができる (図 1)。

2 関連研究

3 次元空間中の既知の点列をカメラで撮影し、その点の画像座標からカメラの位置や姿勢を推定するアルゴリズムは古くから研究されている [8, 2]。たとえば平面上の点から座標系を認識するには、少なくとも 4 個の点が必要なことが知られている。ただし、実際の画像から得られた (誤差を含む) 特徴点情報から、できるだけ正確にカ

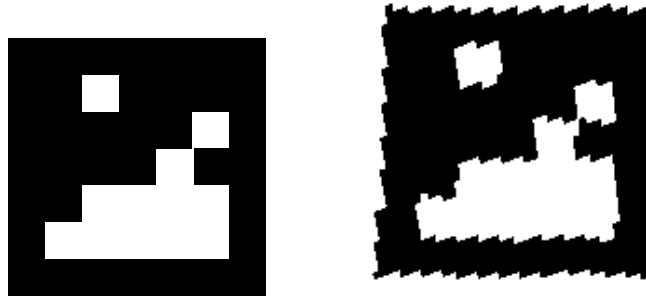


図 2. 2次元マトリックスコードの例 (左: 印刷されたもの, 右: ビデオ映像から認識されたもの)

メラ位置を推定する方式は必ずしも自明ではない。

実世界中のマーカ―をカメラで認識して拡張現実感を構成する方式は Bajura と Neuman によって提案された [1]。Bajura らのシステムでは発光ダイオードをマーカ―として用い、認識を容易にしている。河村らは同様な発想に基づき、カラーマーカ―を用いた方式を提案しているが、画像処理アルゴリズムの制約からか、実時間での合成表示には成功していない [7]。Uenohara らは人工的なマーカ―を用いず、実映像とのテンプレートマッチングを用いた方式を試みている [6]。State らは、マーカ―と磁気センサを併用する方式を提案している [4]。

拡張現実感のシステムでは、オブジェクトの認識のみでも有効な情報を生成することができる。筆者が以前に開発した NaviCam ではバーコードをカメラで認識して実オブジェクトを識別し、そのオブジェクトに関連する情報をオーバーレイする [10, 3]。荒井らの InteractiveDesk では机上の書類をカラーコードによって認識する [9]。しかし、マーカ―を座標系とオブジェクト種別の両方を認識する手段として利用する方法は今まで提案されていなかった。たとえば NaviCam では、画面上のカラーコードの位置を利用して、アノテーション情報と実世界オブジェクトの2次元的な位置合わせを実現していたが、本稿で議論しているような精密かつ3次元的な位置整合は達成されていなかった。

3 2次元マトリックスコードによる拡張現実感

3.1 コード仕様

いわゆる「2次元バーコード」は、現在何種類かの規格提案が成されているが、今回利用している2次元マトリックスコードは本研究用に独自に設計したもので、 5×5 の白黒矩形マトリックスと、同じ太さの黒枠からなる。色情報は利用していない。各セルの白/黒をビットに対応させて、25ビットの情報が格納可能である。現在は16ビットをデータとして、9ビットを誤り判定用に割り当てている。コード形状が正方形なので、同じコードでも回転によって最大4種類の見え方があり得る。これもエラー訂正によって識別し、正しい姿勢を一つ特定する。図2にコードの例を示す。

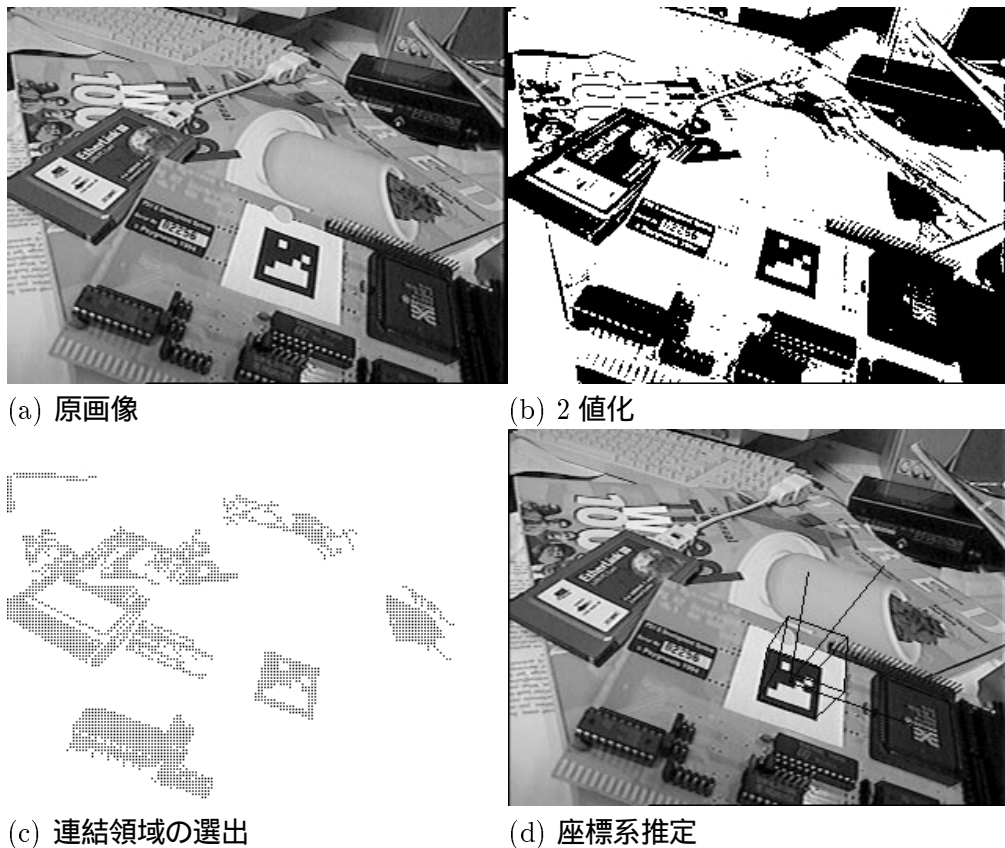


図 3. マトリックスコード認識の過程

3.2 認識アルゴリズム

機器の表面や壁などに添付されたマトリックスコードのビデオ映像からコード番号および空間座標系を得るアルゴリズムについて説明する。本アルゴリズムは以下の6段階から成る。

(1) 前処理

まず画像（グレースケールあるいはカラー）を適当な閾値で2値化する（図3-(b)）。コードは白地に黒で印刷されているので、固定閾値でもかなり安定的にコード領域を分離することができる。次に、黒ピクセルの連結領域ごとにラベルづけをしていく。マトリックスコードの外枠は、ラベルづけされた連結領域のいずれかに含まれることになる。連結領域の外接四角形の大きさと縦横比を考慮して、コードが含まれている可能性の低い領域は除去する（図3-(c)）。

(2) コード枠の当てはめ

前処理の結果得られた連結成分の各要素に対して、コード枠の当てはめを行なう。外接四角形の各辺から内側に向かって黒領域を探索し、枠の点列を得る。この点列に対して最小二乗法で線分を当てはめる。

(3) コード認識

コード枠の4頂点を正方形の頂点に射影する変換行列を求める。実世界中の平面上の点 $(x_i, y_i, 0)$ を、ある並進・回転運動によって移動し、それを透視変換で画像座標系に射影した点を (X_i, Y_i) とする。両者の間には

$$X_i = \frac{a_1 x_i + a_2 y_i + a_3}{a_7 x_i + a_8 y_i + 1}$$

$$Y_i = \frac{a_4 x_i + a_5 y_i + a_6}{a_7 x_i + a_8 y_i + 1}$$

なる関係がある。 a_1, \dots, a_8 は未知の係数で、カメラの外部パラメタ (位置と方向)、および焦点距離などの内部パラメタを表現している。これらの値は、実世界中の既知の点と、その画像座標上での対応点が4組あれば、以下の方程式を解くことによって得られる。

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -X_1 x_1 & -X_1 y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -X_2 x_2 & -X_2 y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -X_3 x_3 & -X_3 y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -X_4 x_4 & -X_4 y_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -Y_1 x_1 & -Y_1 y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -Y_2 x_2 & -Y_2 y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -Y_3 x_3 & -Y_3 y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -Y_4 x_4 & -Y_4 y_4 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{pmatrix}$$

特に、 $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ を一辺の長さ1の正方形の4頂点とすると、求めた係数によって、画面上の矩形頂点を正方形の各頂点に射影することができる。すなわち、画面上のコード枠は、カメラの姿勢や透視投影によって歪んで撮影されているが、これを正方形に射影しなおすことができる。図2(左)に、このようにして復元したコード画像を示す。射影した2次元コード画像の、各セル領域内にある黒と白のピクセル数の比率によって、セルのビットを決定する。得られたコード値を誤り判定ビットでチェックして、エラーがない場合はコードが認識されたものとする。

(4) コード枠推定の最適化

前段までの処理で、コードのID情報は認識された。次に、コード枠の形状から実世界座標系を推定する。前段階で求めた係数をそのまま利用して変換行列を求めるともできるが、画像認識やピクセル量子化の誤差の影響で、推定する座標系が直交しないものになってしまう。より精度を高く座標系を推定するために、矩形コードの外枠が正方形であるという制約条件を利用して、その制約条件の下で誤差を最小にするように変換行列を推定する。ここでは[12]で提案されている方式を利用する。

コードの各辺に相当する(カメラ座標系での)方向ベクトルを $\vec{v}_0, \vec{v}_1, \vec{v}_2, \vec{v}_3$, 対角線に相当する方向ベクトルを \vec{v}_4, \vec{v}_5 とする。正方形の制約条件は、隣接する二辺および二つの対角線がそれぞれ直交することなので、

$$E = (\vec{v}_0 \cdot \vec{v}_1)^2 + (\vec{v}_1 \cdot \vec{v}_2)^2 + (\vec{v}_2 \cdot \vec{v}_3)^2 + (\vec{v}_3 \cdot \vec{v}_0)^2 + (\vec{v}_4 \cdot \vec{v}_5)^2 = 0$$

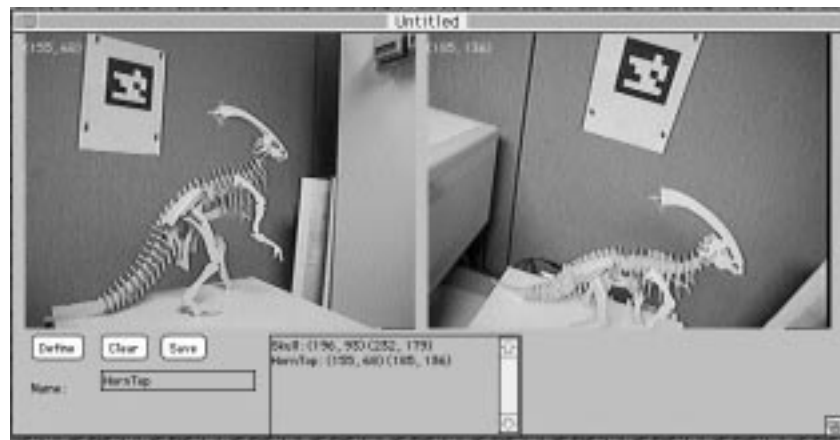


図 4. オーバーレイ情報定義ツール

と表現することができる (\cdot はベクトルの内積)。 $\vec{v}_0, \dots, \vec{v}_5$ をコードが含まれる平面のカメラ座標系における法線ベクトル \vec{n} で表現し、 $E(\vec{n}) \rightarrow 0$ とするように \vec{n} を推定する。

(5) 座標系推定

前段の処理によって新たに得られた法線ベクトル \vec{n} から、 $\vec{v}_0, \dots, \vec{v}_5$ を再計算する。空間中の点の実世界座標系での座標 $(x, y, z)^T$ とカメラ座標系での座標 $(X, Y, Z)^T$ は、以下の変換行列で関連づけられる。

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z & \vec{e}_t \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

ただし $\vec{e}_x, \vec{e}_y, \vec{e}_z$ はカメラの回転を表す成分、 \vec{e}_t はカメラの移動を表す成分で、

$$\vec{e}_x = N(\vec{v}_5 - \vec{v}_4),$$

$$\vec{e}_y = N(\vec{v}_4 + \vec{v}_5),$$

$$\vec{e}_z = N(\vec{n}),$$

$$\vec{e}_t = dist \times N(\vec{p}),$$

$dist$: カメラ原点から矩形中心までの距離,

\vec{p} : カメラ原点と画像上の矩形中心座標を結ぶ方向ベクトル

である (N は正規化演算)。

図 3-(d) は、このようにして推定したカメラの位置・姿勢情報を利用して、コード枠を底面とする立方体を生成して座標変換し、ビデオ映像上に投影した例である。実用上問題ない程度の精度で位置合わせが達成されているのがわかる。

なお、実映像に情報をオーバーレイするためには、カメラの位置や方向のような外部パラメタだけではなく、カメラの焦点距離、歪み、CCD 画素面とレンズ光軸との位置関係等の内部パラメタも求めておく必要がある。これらの内部パラメタは (カ

2次元マトリックスコードを利用した拡張現実感の構成手法

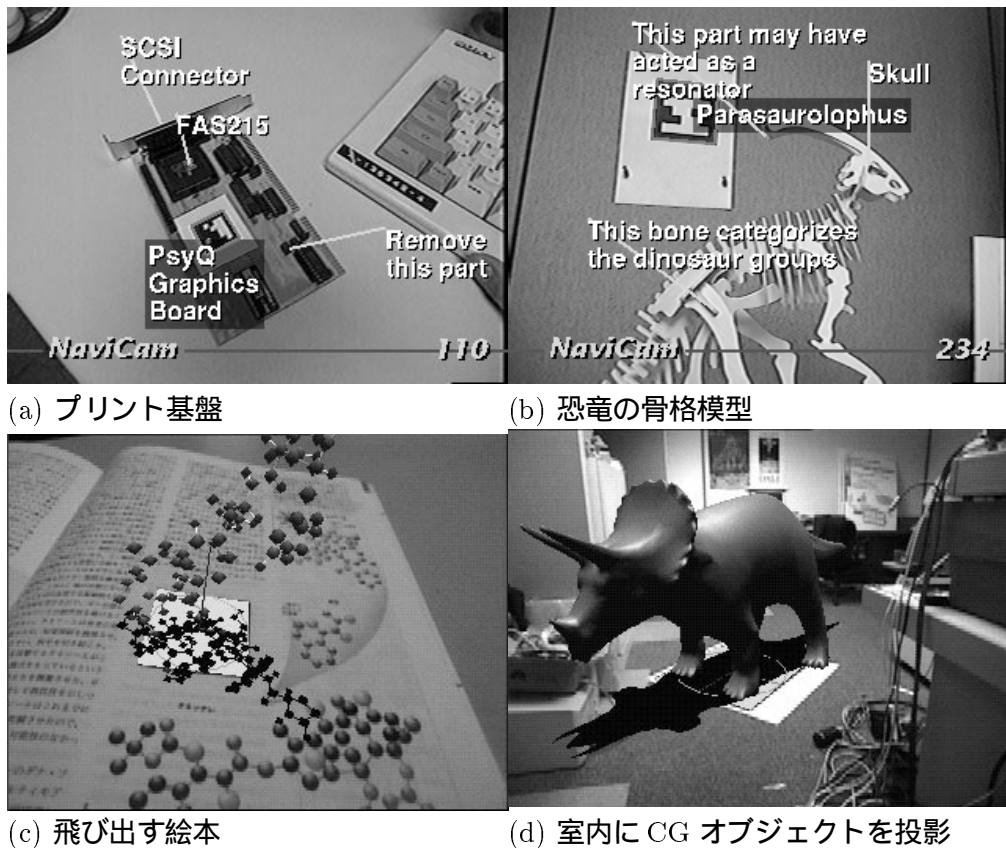


図 5. 合成映像の例

メラのズームングをしなければ) 実行時中は変化しない。したがって、カメラキャリブレーション用のアルゴリズム、たとえば Tsai の方法 [5] 等によって事前に測定しておくことができる¹。

3.3 オーバーレイ情報の定義

前節で説明したアルゴリズムによってオブジェクトが認識され、その座標系が求められると、3次元的に整合された付加情報をオーバーレイすることができる。たとえば恐竜の模型に対して、骨格の各部分に説明情報を添付することができる。しかし、説明を添付する点の実世界における3次元座標を計測して登録するのは簡単な作業ではないし、3次元デジタイザのような特殊な計測装置を必要とする。

そこで、オーバーレイ情報を添付する対象を複数視点から撮影して、その映像上の点(2次元座標)の組から3次元座標を推定するツールを開発した。原理はステレオ視と同じである。位置と方向が既知²のカメラで撮影された画面上の1点は、3次元空間中ではカメラの原点を通る直線となる。したがって複数視点から同じ場所を

¹ 次節で示す試作システムでは、内部パラメタのうち、焦点距離のみを計測して用いている。

² この情報は画像中の2次元コードから得られる。

指定した場合、複数の直線の交点が求める点となる。実際には誤差によって直線が厳密に1点では交差しないので、最小二乗法によってすべての直線からの距離が最も短い点を算出する。定義ツールの例を図5に示す。オーバーレイ情報を添付した箇所を複数の画像に対してマウスでピックして指定する。

4 結果

4.1 試作システムの概要

前節で示したアルゴリズムに基づいてシステムを試作した。装置構成はNaviCam[10]と同様、パームトップ型のディスプレイとTVカメラを組み合わせた携帯型装置を用いており、画像認識と情報の生成はビデオケーブルによって接続されたワークステーションが担当する。システム利用者は、携帯型ディスプレイを通じて、ビデオ越しの現実映像とコンピュータ映像との合成表示を見ることになる。この構成は、液晶画面付きのビデオカメラやデジタルスチルカメラ等と親和性が高く、利用者にとって親しみやすい形態だと考えている。なお、本論文のアルゴリズム自身は、携帯型ディスプレイのみならず、ビデオ透過型HMD(head-mounted display)にも適用可能である。

画像処理に利用したワークステーションはSilicon Graphics Indigo² IMPACT(CPUはMIPS R10000 200MHz)に画像入力ボード Indigo² Videoを付加したものである。この機器構成で、ビデオキャプチャー、画像認識、イメージ合成の一連の処理を、毎秒7~10フレーム程度の頻度で行なうことができ、対話的な拡張現実感アプリケーションを構築することが十分可能であった。処理時間のうち、約半分の時間が単に画像をキャプチャーして表示するだけに費やされていた。認識アルゴリズムでは、処理(1)の連結成分をラベルづけする箇所が最も多く時間を消費している。一般的な映像では、連結領域の個数は100個以上になるが、処理(1)で述べた簡単な判定基準によって、処理(2)以降を行なう領域の個数は1~5個程度になる。認識できる距離は、コードの一辺の長さの約10倍であった(たとえば1辺10cmのマトリックスコードは、約1mの距離から認識できる)。

4.2 利用例

図5-(a, b)は、実世界オブジェクトにアノテーション情報を付加した例である。オーバーレイする情報は、認識されるコードIDによって検索可能なデータベースに蓄積されている。登録されている情報は[アノテーションを与える3次元座標、引き出し線の方向、アノテーション内容]の組から成っている。

図5-(c, d)は、現在、実世界に整合したCGオブジェクトの生成し、表示した例である。(c)は、本に印刷されたマトリックスコードから、対応する3次元モデルをデータベースから検索し、合成表示する電子的な「飛び出す絵本」である。紙という2次元情報を、コンピュータによって3次元的に拡張していることになる。この方式の延長として、紙に印刷された建築物の平面図に、立体的な建造物をオーバーレイさせたり、地図の上に3次元的な気象モデルを合成するような応用例が考えられる。

(d)は、床に置かれたコードから、CGオブジェクトを合成する例である。CGオブジェクトサイズは、利用されるコードの1辺の長さによって相対的に決定される。したがって、あらかじめサイズの分かっているコード利用すれば、現実物とCGオ

プロジェクトのサイズを比較することができる。たとえば、WWW を使って家具をオンライン購入する場面を想定すると、その家具が自分の部屋のサイズにマッチするかどうかをオンライン情報だけから判断することはむずかしい。しかし、本研究の方式を利用すれば、家具の3次元モデルをネットワークからダウンロードし、現実の部屋の映像に合成してみて、あたかもそのオブジェクトが現実にあるかのようにして、その大きさを確かめることができる。本方式ではカメラ以外の特別な計測装置を必要としないので、将来的にはデジタルカメラ程度の機器構成で利用可能となることが期待でき、家庭内でも充分利用可能になると考えている。

以上の例は、単なる実世界映像と仮想情報の合成表示に留まっているが、現在、ボタン操作や音声コマンドに回答した、インタラクティブな情報の生成の実現を計画している。たとえば恐竜の骨格モデルを部分的にCGによって肉づけして見せたり、利用者の選択によって説明の内容が変化するような応用を想定している。

4.3 考察

2次元マトリックスコードのようなマーカー方式には、マーカーが常にカメラの視野内に収まらなければならないという制約がある。この制約を回避する方法として、以下の3種類を方式を検討中である。

- 他の3次元センサを補助的に利用する。マーカーが視野外に出た場合、3次元センサの情報によってトラッキングを継続させる。補助センサとしては、磁気センサのように絶対的な位置、姿勢が得られるもの [1, 4]、ジャイロセンサのように相対的な移動、回転が計測できるものが考えられる [11]。とくにジャイロセンサのように、自立型ではあるが定期的に絶対位置を取得する必要のある3次元センサとの併用は効果的であろう。
- 複数のマーカーを利用する。同一オブジェクトに対して複数のマーカーを添付して、そのどれかが視野内に入っていれば計測が可能であるようにする。複数マーカー間の位置関係はあらかじめ計測しておくものとする。この場合、ID認識と併用する必要のあるマーカーは一つだけで、他は単なるカラースポットのようなものでもよいかも知れない。
- 最初にコードを認識した時点で(データベースに登録されている)周囲の特徴点情報を引き出し、それを追跡すべきテンプレートとして利用する。この方式は画像処理におけるテンプレートマッチング [6] とID認識との組み合わせであると考えられる。制限のない環境で画像情報のみを用いた場合、マッチすべきテンプレートを絞り込むのに膨大な計算量を要するが、本方式では最初にIDを認識した時点で、(そのオブジェクトに関連する)テンプレートのみを選択できる点がより実用的だと考えられる。

5 結論

ビデオカメラのみで特別な計測装置を必要としない拡張現実感システムの構成手法について報告した。2次元マトリックスコードを、実世界オブジェクトの識別と空間座標系の認識の両方の目的に利用する方式を提案し、認識アルゴリズムについて解説した。提案方式に基づいて実際にシステムを構築し、通常のワークステーション程度の計算能力で実時間で実行可能であることを確認した。今後はこの方式を、機

器保守システムや、美術館案内システム等の種々の拡張現実感システムに適用していく予定である。

参考文献

- [1] Bajura, M. and Neumann, U.: Dynamic Registration Correction in Augmented-Reality Systems, *Virtual Reality Annual International Symposium (VRAIS) '95*, 1995, pp. 189–196.
- [2] Bogart, R. G.: View Correlation, *Graphics Gems II*, (1991), pp. 181–190.
- [3] Rekimoto, J. and Nagao, K.: The World through the Computer: Computer Augmented Interaction with Real World Environments, *Proceedings of UIST'95, ACM Symposium on User Interface Software and Technology*, November 1995, pp. 29–36.
- [4] State, A., Hirota, G., Chen, D. T., Garrett, W. F., and Livingston, M. A.: Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, *SIGGRAPH'96 Proceedings*, 1996.
- [5] Tsai, R. Y.: An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1986, pp. 364–374.
- [6] Uenohara, M. and Kanade, T.: Real-Time vision based object registration for image overlay, *Journal of the Computers in Biology and Medicine*, (1995), pp. 249–260.
- [7] 河村貴弘, 舘村純一, 坂内正夫: 実映像中のランドマーク情報を利用した拡張現実感の生成手法, 情報処理学会第52回(平成8年前期)全国大会, 1996.
- [8] 金谷健一: 画像理解 – 3次元認識の数理, 森北出版株式会社, 1990.
- [9] 荒井俊史, 葛貫壮四郎: 実世界での操作に反応する機型の対話環境, *インタラクティブシステムとソフトウェア II*, 1994, pp. 19–28.
- [10] 暦本純一: *Augmented Interaction: 状況認識に基づく新しいインタラクションスタイルの提案*, 近代科学社, 1994, pp. 9–17.
- [11] 暦本純一: 簡易性とスケーラビリティを考慮した拡張現実感システムの提案, 近代科学社, 1995, pp. 49–56.
- [12] 高橋章, 石井郁夫, 牧野秀夫, 中静真: VR インターフェースのための単眼による長方形マーカー位置高精度実時間推定法, *3次元画像コンファレンス'96*, 1996, pp. 167–172.