

実空間利用のためのビジュアルなコンピュータ間通信方式

塚本昌彦

大阪大学大学院工学研究科情報システム工学専攻

tuka@ise.eng.osaka-u.ac.jp

<http://www.nishio.ise.eng.osaka-u.ac.jp>

拡張現実感などのシステムで、カメラ画像を通して現実空間を高度に利用するためには、現実空間の座標を正確に同定する必要がある。このような位置登録のためのさまざまな位置マーカ方式がこれまでも提案されているが、従来のマーカ方式は柔軟性が低く検出が難しいという問題があった。本稿では、コンピュータディスプレイを用いてマーカ情報を送信する方式を提案する。提案方式をVCC (Visual Computer Communication) と呼ぶ。マトリックス状のマーカの点滅で座標情報およびアドレスや URL などの情報を提供する。本方式では、カメラのキャリブレーションが不要で、しかも、送信側で柔軟に送信内容やマーカ位置を変更できるとともに、受信側でも簡単な解析で検出できる。コンピュータ画面上の対象のビジュアルな操作を、カメラを通して遠隔から行うような利用方法が考えられる。

A Visual Computer Communication Method for Real Space Utilization

Masahiko Tsukamoto

Information Systems Engineering, Graduate School of Engineering
Osaka University

In order to utilize the real world with computer through camera images in advanced real world applications such as augmented reality systems, it is necessary to identify the concise coordinate information of the taken images. Although several methods for such location registration have been proposed so far, the conventional markers are computationally hard to be detected and their locations or the information cannot be flexibly changed. In this paper, we propose a location marking method, called VCC (visual computer communication), where we use a computer display to send information and a video camera to receive it. In our method, a marker in a matrix shape keeps blinking to provide the coordinate information and the attached information such as an address or a URL. There are several merits of this method which are not provided in the conventional methods: it does not require the camera calibration, it is possible to change the location and the contents of a marker dynamically by a computer application, and the received camera image can be analyzed by little computation to detect the coordinate and the attached information. A typical application is to utilize the display of a remote computer through visual operations on a local computer.

1. はじめに

ウェアラブルコンピューティングや拡張現実感など、現実空間を高度に利用したコンピュータ利用に関する研究が多数行われている。なかでも、カメラで撮影した現実空間にコンピュータ情報をリアルタ

イムに重ね合わせるアプローチは、拡張現実感のアプローチの中でも主流をなすものであり、次のような利用方法が有効であると考えられている[塚本99]。

- ・ 目の前にある場所や建物、人などにアノテーションをつける。人の名前、建物の名前、ものの名前や使い方などを把握できる。

- ・ 目の前にある電気器具やコンピュータ機器を利用できる。プリンタ、ディスプレイ、コンピュータ、電灯、エアコンなどの利用が容易になる。

このような利用を可能にするためには、今カメラで映されている部分が現実空間のどの部分であるのか、物体や人がどの方向を向いてどのような格好をしているのかなどを、カメラ画像上の座標系に基づき正確に知る必要がある[Azuma97]。このような現実空間の位置情報の取得は位置登録 (location registration) と呼ばれ、従来から、GPS、磁気センサ、加速度センサ、ジャイロセンサなどのセンサを用いた手法、PHS や赤外線通信による基地局の ID による識別などの通信を用いた手法などさまざまな手法が提案されてきた[State96, Neumann99 など]。一般に、ビデオ画像と位置情報の重ね合わせを行うためには非常に高い精度が必要である[Azuma93] のに対し、センサ等を用いたアプローチはいずれも誤差が大きいため、実環境での利用は困難であった。

これに対し、最近では画像解析をベースとするアプローチが盛んに行われるようになってきている[佐藤 99, 加藤 99, Nagao95 など]。特にビデオスルー型のシステムでは、ビデオ画像を利用すれば直接画像内の座標系をベースとして計算が行えるため、検出の精度はなくてもカメラ画像との位置合わせは容易であるというメリットがある。

ビデオスルー型のシステムにおける位置登録のためのとして、マトリックスコードと呼ばれる位置マーカ―を利用するものがある[Rekimoto98]。マトリックスコードとは、バーコードの2次元バージョンであり、コードの枠形状から空間座標も読み取ることができるようになってきている。しかし、このアプローチにも次のような問題点がある。

- ・ マーカ―をプリントアウトして適切な場所に貼らなければならない。また、貼った後で情報を動的に変更できない。特に、コンピュータアプリケーションから、直接情報を貼ったり動かしたりということができない。
- ・ 情報量に限界がある。マトリックスコードでは5×5ビットである。2次元バーコードの規格には大容量のものもあるが、大容量になれば情報を表現するドットの粒度が小さくなるため、距離が離れると識別できなくなる。
- ・ 実世界画像は一般に非常に複雑で色や動きが多様であるため、十分な精度が得られない。逆に解析を正確に行おうとすると計算量が大きくな

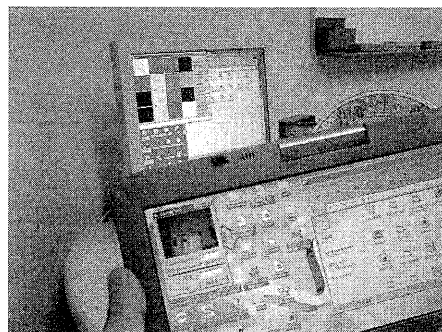


図1. VCC 利用状況の例

り、ノート PC (Personal Computer) や PDA (Personal Digital Assistants) での利用が困難になる。

これらの問題点を解決するため、本稿では動的な位置マーカ―手法を提案する。コンピュータディスプレイを用いて送信することで、柔軟に送信内容を変更できるようにし、時間方向での変調を利用することで、従来の方法よりも簡単な処理でロバストな通信を実現することをねらいとする。

以下、2章で方式を示し、3章で実装について述べる。4章で問題点を含めたさまざまな点について議論を行い、5章でまとめを行う。

2. 通信方式

2. 1 方式概要

本稿で提案する方式を VCC (Visual Computer Communication) 方式と呼ぶ。VCC では、位置マーカ―として、コンピュータディスプレイ上に表示したマトリックスコードを用いる。情報は表示されている内容(振幅変調)ではなく、決められた間隔で点滅したときの差分(位相変調)で行う。受信側は同じ間隔でカメラをスキャンし、前回受信画像との差分画像を解析して検出を行う。利用状況の例を図1に示す。デスクトップ PC のディスプレイ上にカラーのマトリックスコードが点滅しており、受信側のカメラ付きノート PC でそれをリアルタイムに解析している。

紙に印字されたマトリックスコードと比べると、ディスプレイを利用するメリットとして次のような点が挙げられる。

- ・ コンピュータにより動的にデータの送信内容を

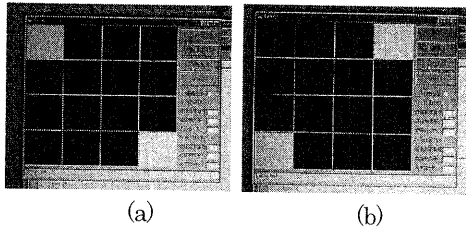


図2. 送信画面例(データ非送信時)

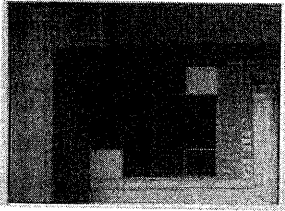


図3. 取得画像(データ非送信時)

変更できる。

- ディスプレイ内であれば表示位置を自由に変更できる。
- 時間軸を利用することで、任意長のデータを送信できる。
- 受信側では時間差分をとることで、他の物体の影響をあまりうけず、ロバストなデータ受信が行える。
- 必要時以外はマーカを表示しないことができる。

2. 2 ロケータ部

本稿では座標系の検出のための領域のことをロケータとよぶ。[Rekimoto98]では四角形の4辺を黒枠で囲み、さらにその周辺領域を白にすることで、マーカ検出側(カメラ側)で四角形を検出しているが、本稿ではマトリックス上のデータ領域を広く取るために、4隅のみをロケータとする。紙に印刷されたマトリックスコードと比べて、ディスプレイを用いる場合には、時間軸を利用できるので、カメラ画像の時間差分をとった場合に、ノイズがかなり除去できることが期待できるためである。

図2にロケータ部分の送信例を示す。(a)を表示した後、一定時間(例えば500ms)後に(b)、同じ間隔の後に再び(a)という具合に、(a)、(b)を交互に表示する。図3に取得したカメラ画像を、図4に差分画

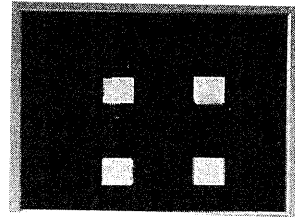


図4. 差分画像(データ非送信時)

表1. コーディング

ロケータ	送信開始	送信終了	ロケータ
第1ビット	第2ビット	第3ビット	第4ビット
第5ビット	第6ビット	第7ビット	第8ビット
ロケータ	パリティ	2バイト目無効	ロケータ

像を2値化したものを示す。図から明らかなように、取得画像がぼんやりしていても、差分画像は比較的検出しやすくなっている。

2. 3 データ部

データの送信はロケータ以外の領域で行う。本稿で用いているのは4x4のマーカなので、ロケータ領域の4個所を除けば、残り12個所の領域が利用できる。

コーディングの方法を表1に示す。ロケータ部は白・黒を反転し、データ部には赤と緑の成分を組み合わせて用いる。ロケータの初期値は対角頂点を同色、隣接頂点を異色とするが、これは受信側では特に利用しない。データ部には、第 t 回目($t=1, \dots, [(n+1)/2]$)の点滅で、赤は第 $2t-1$ バイト、緑は第 $2t$ バイトをコーディングする。 n は送信バイト数、 $[x]$ は x を超えない最大の整数を表す。パリティは偶数、赤、緑それぞれに第 $2t-1$ バイト、第 $2t$ バイトのパリティを入れる。赤と緑の両方のビットが1のときには黄色(=赤+緑)を表示する。第1回目の点滅で送信開始部の赤を反転し、第 $[(n+1)/2]$ 回目の点滅で送信終了部の赤を反転する。 $n=1,2$ の場合、最初の点滅で両方を反転することになる。2バイト目

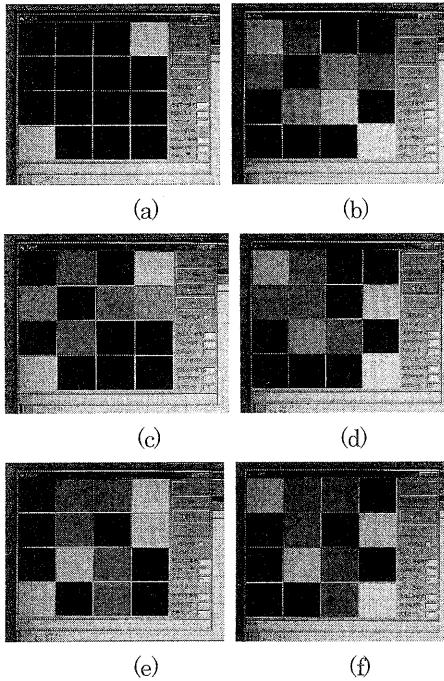


図5. 送信画面例(データ送信時)

表2. 図5で表示されている色

	(a)	(b)	(c)	(d)	(e)	(f)
ロケータ	R	W	R	W	R	W
送信開始	B	R	R	R	R	R
送信終了	B	B	B	B	R	R
ロケータ	W	B	W	B	W	B
第1ビット	B	G	G	R	B	B
第2ビット	B	B	B	R	R	R
第3ビット	B	G	R	B	B	B
第4ビット	B	R	G	Y	Y	Y
第5ビット	B	B	B	B	B	B
第6ビット	B	G	R	G	Y	Y
第7ビット	B	Y	B	R	R	R
第8ビット	B	B	B	B	B	B
ロケータ	W	B	W	B	W	B
パリティ	B	B	B	B	B	B
2バイト無効	B	B	B	B	R	R
ロケータ	B	W	B	W	B	W

無効部は、第 $[(n+1)/2]$ 回目の点滅時に2バイト目(緑成分)が有効か無効かを示す。データ部の初期値は任意とする。

図5に送信画面例を示す。(a)から(f)まで決められた間隔で表示する。(f)の後は、(e)と(f)を繰り返し表示する。(b)が送信の開始で、 H (文字コード72)と

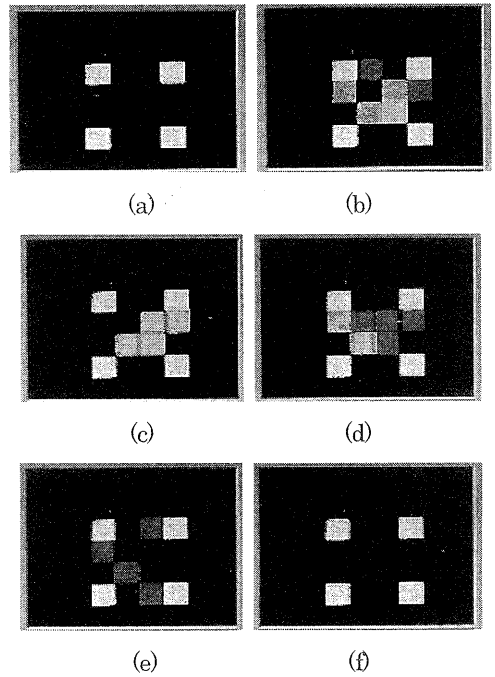


図6. 差分画像(データ非送信)

表3. 図6で表示されている色

	(a)	(b)	(c)	(d)	(e)	(f)
ロケータ	W	W	W	W	W	W
送信開始	B	R	B	B	B	B
送信終了	B	B	B	B	R	B
ロケータ	W	W	W	W	W	W
第1ビット	B	G	B	Y	R	B
第2ビット	B	B	B	R	B	B
第3ビット	B	G	Y	R	B	B
第4ビット	B	R	Y	R	B	B
第5ビット	B	B	B	B	B	B
第6ビット	B	G	Y	Y	R	B
第7ビット	B	Y	Y	R	B	B
第8ビット	B	B	B	B	B	B
ロケータ	W	W	W	W	W	W
パリティ	B	B	B	B	B	B
2バイト無効	B	B	B	B	R	B
ロケータ	W	W	W	W	W	W

‘e(101)’の文字コードを送っている。(c)は $\text{I}(108)$ と $\text{I}(108)$, (d)は $\text{o}(111)$ と $\text{I}(33)$, (e)は最終文字 $\text{I}(33)$ 一文字を送っている。表2に図5で表示されている色を示す。Bは黒, Wは白, Rは赤, Gは青, Yは黄色を表す。

対応する差分画像を図6に示す。(b)で送信開始を

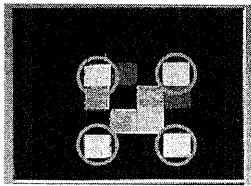


図7. 白領域のクラスタリング

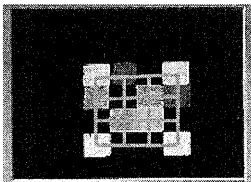


図8. コード枠の当てはめ

検出すると同時に、データ受信を開始し、2 バイトずつ読み取って、(e)で1バイトを読むと同時に送信終了を検出し、合計7バイトの受信を終了する。表3に図6で表示されている色を示す。

2. 4 受信アルゴリズム

受信側の動作手順を示す。受信側は送信側と同じ間隔でビデオ画像を RGB カラーでキャプチャする。1 回前にキャプチャした画像との差分画像を作り、適当な閾値で2値化する。2 値化画像から連結白領域をクラスタリングして、大きな領域を4 領域選び、それぞれの中心を結んで四角形をつくる。図6(b)を対象画像としたとき、図7の丸で囲んだ領域が連結白領域として検出される。

検出した四角形にコード枠に当てはめる。[Rekimoto98]と同様の方法を用いて変換行列を求めることもできるが、ここでは図8のように単純に各辺を3等分して格子状につないだ。4隅の頂点を除く格子点がコード領域となる。各コード領域の近辺から数点をサンプリングして、その平均値(あるいはメジアン値)からコードを読み取る。

送信開始ビットが設定されているとデータバッファをクリアしてデータ部の読み取りを開始する。送信終了ビットが設定されているとそのデータで読み取り終了とする。データを読み取った後はパリティチェックをし、失敗すれば受信を中止する。

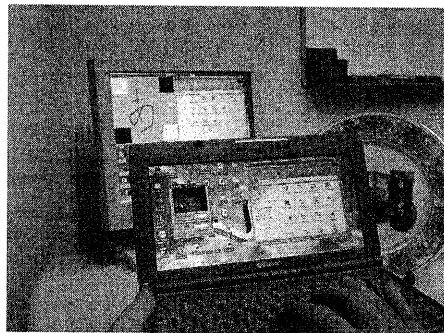


図9. 遠隔ドローツールの利用状況例

3. 実装

VCC に基づくシステムを、Windows98 上で VisualBasic6.0 を用いて実装した。受信側は、ソニー社バイオノート PCG-C1S (MMX Pentium 266MHz, メモリ 128MB)を用いて、160×120 ピクセル、24bitRGB で500ms/frame で動作を確認した。送信側はソニー社バイオノートPCG-505SX (MMX Pentium 300MHz, メモリ 128MB)で、400×400 ピクセル送信領域でキャプチャし、500ms/frame で動作を確認した。また、送信側は100ms 程度までは正常に動作することを確認した。それ以上早くするとキャプチャ画面でマーカーがきれいに塗りつぶされていない場合が見られた。送信部の大きさを小さくするともっと早く送信ができるようになるが、遠くから検出できなくなる。

さらに、サンプルアプリケーションとして、遠隔ドローツールを作った。送信側、受信側はともにインターネット接続されており、送信側は、1 秒周期(2回送信分)でIPアドレス(4バイト)を繰り返し送信する。受信側はアドレスを検出すると、そのアドレスにソケット接続する。接続を受けた送信側はアドレス送信を止め、データ領域をすべて書き込み用にクリアする。ロケータ部の点滅は継続する。受信側は、カメラ画像として画面に表示されたりリモートマシンのデータ領域をマウスでドラッグすると、リモートマシン側の対応する領域に線分が描かれる。受信側からリモートマシンへのマウスのドラッグ座標の情報は、データソケット通信で送信している。

図9に利用状況の例を示す。手前のコンピュータが受信側、奥のディスプレイが送信側である。図10は受信側の画面で、検出したマーカー位置を線で

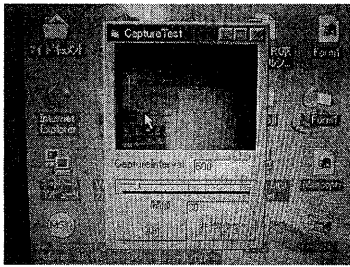


図1 0. マーカー受信側(操作側)の画面

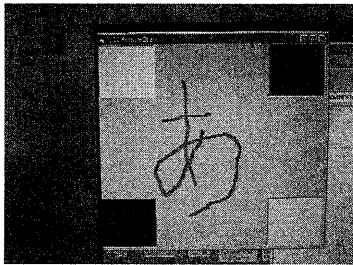


図1 1. マーカー送信側(リモート側)の画面

囲っている。マウスを用いて奥のコンピュータのカメラ画像に対してドラッグすることで文字を書き込んでいる。図1 1は書き込みが行われた奥のコンピュータの画面である。

4. 考察

通信性能

現在の実装では通信レートは 32bps 相当である。一般に 1 フレームの受信処理時間を P 、送信間隔を $T (> P)$ 、総受信データ量を n バイトとすると受信にかかるトータルの時間は、

$$T \times [(n+1)/2] + P$$

で表される。例えば、 $P=400\text{ms}$ 、 $T=500\text{ms}$ 、 $n=4$ バイト(IP アドレス)のとき、1.4 秒で識別が行え、これくらいの時間なら実用の範囲内であるといえる。

同じデータを繰り返し送信する場合、最終バイト送信後、次のフレームで新たに送信を開始する場合、受信を開始してからデータを読み取るまでの平均時間は、

$$1.5 T \times [(n+1)/2] + P$$

となる。前述の例では、受信を開始して平均 1.9 秒でアドレスを認識できることになる。

マトリックスサイズについては再考の余地がある。

表4. マトリックスサイズと同時送信可能バイト数

マトリックスサイズ(c)	5	6	7	8	9	10
同時送信可能バイト数(n)	4	6	9	12	16	20

今回の提案では 4×4 の格子を用いたが、データ領域が増えてもサンプル点の読み取りだけなので画像処理の付加はそれほど変わらない。そのためマトリックスサイズをもっと大きくしても動作速度にはそれほど影響はないものと考えられる。むしろビデオキャプチャとバッファコピー、枠領域の検出に時間がかかっている。ただし、サイズを増やすと領域の大きさが相対的に小さくなり、遠くから読み取りにくくなる。

次に、マトリックスサイズと同時送信可能バイト数の関係について考える。今回と同じビット構成を利用すると、 n バイト送信する場合に必要なビット数は、

$$9n + 2 + \log_2 n$$

となる。 $c \times c$ のマトリックスで送信できるデータは $2(c^2 - 4)$ ビットであることから、マトリックスの大きさに対する同時可能送信バイト数(n)を求めることができる。その関係の一部を表4に示す。例えば、 8×8 のマトリックスを用いて 500ms 間隔で送信すれば、192bps 相当となる。

閾値の決定方法について

24 ビット RGB カラーの画素データは、RGB それぞれ0から255の値をとるが、差分画像の2値化においては適切な閾値を決める必要がある。今回の実験では終始固定値 30 を用いたが、まわりに動く物体がなく、受信側をきちんと静止していれば特に問題なく受信できた。ただし、送信側のディスプレイがカメラに対して斜めに向いているときには、カメラ画像の場所ごとに明るさがかなり異なっていた。現在の実装ではこのような場合にはうまく受信ができないが、閾値を画面上の場所ごとに変えることで対処できる可能性がある。また、RGB によって閾値を変えることも検討の余地がある。

マーカーの形状について

マーカーの形状は正方形ではなく、 1×8 などのような棒状のものでもよい。色をうまく使えば 1×1 のタイル状のものも可能である。また今回は、ロケータ部を4隅だけにしたが、[Rekimoto98]のように枠全体にする方が雑音などに対してロバストになるが、それだけデータ領域が狭くなり、データレート

表5. 各色に対する検出値の例

	Bl	R	G	Y	Br	M	C	W
R	31	111	48	112	15	111	48	112
G	24	32	103	95	71	56	128	112
B	16	16	31	35	112	111	112	96

が落ちることになる。

符号化方法について

今回はプロトタイプ実装の試行錯誤の過程で符号化方式を決定したため、方式細部の決定は論拠に乏しい。今後は綿密な実験に基づき、より最適な符号化を考えなければならない。パリティビットだけでなくもっと冗長な符号化も有効であるものと考えられる。大きなデータを送る場合には送信側と受信側で同期がずれてくる可能性がある。送信側の倍以上のレートで受信してロケータを見て同期をとるか、数フレームごとに同期を取り直す調歩同期の方式を取り入れることを検討する必要がある。

色情報の利用について

今回は赤と緑をデータ領域で使用したがこの点に付いても再考の必要がある。一般に、赤を表示してもカメラで取得した画像では緑や青の成分も検出される。また、黒から赤を反転させるのと、緑から赤を反転させるのでは、差分値の特性は異なる。このようなことから、ある条件下での各色に対するRGB成分の平均検出値(0~255)を測定してみた。その結果を表5に示す。Blは黒、Brは青、Mはマゼンタ、Cはシアンを表す。他の記号の意味は表2と同じである。網掛け部分は理想的には255となるべき部分、それ以外の部分は0となるべき部分である。青の表示時に緑の成分がかなり出ていること、緑でも青成分が出ていること、反転させた場合も初期値によって差分が異なることなどがわかる。環境によってこれらの値は大きく変化するので、青を使う場合には緑の閾値の設定が難しくなる。これらの点をよく考慮してプロトコルを設計する必要がある。

明暗(階調)の利用について

前述の検出値ばらつきの問題もあり、RGBで階調を利用するのは閾値を決めるのが難しい。YUVに変換して色相を用いる(周波数変調)ならもう少し細分化できるかもしれない。

実装について

今回の実装では、画面内にマーカーが複数存在することには対処していない。検出したロケータをど

のようにつなぐかが難しいが、4点の初期値を変えることで対処できるものと考えている。現時点でも隣接頂点の初期値を変えているので、4点をつなぐときのヒントになるものと考えられる。上下逆方向からの読み取りについても同様の問題がある。鏡像のマーカーが複数ある場合の読み取りについては対処が難しい。パリティエラーとして処理されるだろう。描画とビデオキャプチャのタイミングがちょうど合ってしまったら、いつまでもデータが読めない可能性がある。受信側はランダムにキャプチャタイミングをずらす必要があるだろう。

計算パワーを要するアプリケーション内での利用

アプリケーションが計算パワーを要する場合、受信側のキャプチャと解析が間に合輪なくなる可能性がある。送信間隔は十分ゆっくり行うしか対処方法はないだろう。あるいは受信を専用ボードで行うのもよいだろう。

ディスプレイ上の作業領域の確保

タイルを小さくすると遠くから見えにくくなる。大きいとディスプレイ上の領域を占有するため、ほかの作業の邪魔になる可能性がある。電子透かしの技術が使えるかもしれないが、たぶん遠くからは検出できなくなる可能性がある。

別のセンサとのハイブリッドな方式

マーカーを磁気センサやジャイロセンサなどと組み合わせるアプローチがよく行われている[Auer98, You99, 椎尾99など]。VCCに関してもこのアプローチは有効である。

カメラ制御機構との組み合わせ

カメラズームングやパン・チルトがPC側で制御できれば、遠くのデータでも読め、座標系も組み合わせで利用できる。このような制御が可能なカメラは定点カメラではよくあるがノートPCに装着するものでは現時点では見当たらない。

コスト

VCCはプリント型のマーカーと比べるとコスト高である。いろいろなものにぺたぺたとマーカーを貼るという利用方法は向かないと考える。ただし、電車の中や駅前など、どこにでもディスプレイがあるようになってきたことや、液晶が安くなってきたこと、既存のディスプレイを利用できることなどから、VCCでもそれほどコストをかけずに利用できる。また、VCCを用いたLEDや機械式の送信機も考えられる。

関連研究

Timex 社の腕時計には、コンピュータディスプレイでデータを送信して時計で受信する機能を持つものがある[Timex]。また、家庭用ゲーム機のシューティングゲームでは、テレビ画面に向けて撃つ拳銃がインタフェースとして広く用いられている[Sega など]。前者はディスプレイからデータ送信をするという意味で、後者はディスプレイから座標情報を読み取るという意味で VCC と類似性がある。前者は空間座標を読み取らない点、後者はデータ送信がなくブラウン管ディスプレイでしか利用できない点が異なる。LED を用いた位置マーカーもいくつかの研究で利用されている[Bajura95 など]。この方式では位置情報は取れるが、点滅しかできないので識別情報としては不十分である。

5. おわりに

本稿では、コンピュータディスプレイを用いて情報を送信し、カメラで受信してデータを読み取る方式 VCC を提案した。これによってカメラのキャリブレーションが不要で、しかも、送信側で柔軟に送信内容やマーカー位置を変更できるとともに、受信側でも簡単な解析で検出できる。

VCC はアノテーションや機器利用など多くの拡張現実感のシステムで利用できるだけでなく、コンピュータ画面上の対象のビジュアルな操作がカメラを通して遠隔から行える点が特徴的である。

現在、BER や通信範囲については実験を行っている。また、VCC を用いてさまざまなアプリケーションを構築している。

参考文献

- [Auer98] T. Auer, A. Pinz, M. Gervautz, Tracking in a Multi-User Augmented Reality System, Proc. of the IASTED International Conference, Computer Graphics and Imaging, pp.249-252 (1998).
[Azuma93] R. Azuma, Tracking Requirements for Augmented Reality, Communications of the ACM, Vol.36, No.7, pp.50-51 (1993).
[Azuma97] R. T. Azuma, A Survey of Augmented Reality, Presence: Teleoperators and Virtual Environments, Vol.6, No.4, pp.355-385 (1997).
[Bajura95] M. Bajura, U. Neumann, Dynamic

Registration Correction in Augmented-Reality Systems, Virtual Reality Annual International Symposium (VRAIS) '95, pp.189-196 (1995).

[加藤 99] 加藤博一, M. Billingham, 浅野浩一, 橘啓八郎, マーカー追跡に基づく拡張現実感システムとそのキャリブレーション, 日本バーチャルリアリティ学会論文誌, Vol.4, No.4, pp.607-616 (1999).

[Nagao95] K. Nagao and J. Rekimoto, Ubiquitous Talker: Spoken Language Interaction with Real World Objects, Proc. IJCAI'95, pp.1284-1290 (1995).

[Neumann99] U. Neumann, S. You, Y. Cho, J. Lee, J. Park, Augmented Reality Tracking in Natural Environments, Mixed Reality - Merging Real and Virtual Worlds, Y.Ohta and H.Tamura eds., Ohmsha and Springer-Verlag, pp.101-130 (1999).

[Rekimoto98] J. Rekimoto, Matrix: A Realtime Object Identification and Registration Method for Augmented Reality, Proc. of Asia Pacific Computer Human Interaction 1998 (AOCHI'98) (1998).

[佐藤 99] 佐藤清秀, 山本裕之, 田村秀行, カメラと 3 次元センサの組み合わせによる現実空間と仮想空間の位置合わせ手法, 日本バーチャルリアリティ学会論文誌, Vol.4, No.1, pp.295-301 (1999).

[Sega] Sega, Saturn Products - Peri バーチャガン, <http://www.sega.co.jp/sega/saturn/main/products/vgun.html>.

[椎尾 99] 椎尾 一郎, 増井俊之, 福地健太郎, FieldMouse による実世界インタラクション, インタラクティブシステムとソフトウェア VII, 近代科学社, pp.125-134 (1999).

[State96] A. State, G. Hirota, D. T. Chen, B. Garrett, M. Livingston, Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking, SIGGRAPH 96, pp.429-446 (1996).

[Timex] Timex, Datalink, http://www.timex.com/html/data_link.html.

[塚本 99] 塚本昌彦, モバイル先進アプリケーション, bit, 共立出版, Vol.31, No.1, pp.31-38 (1999).

[You99] S. You, U. Neumann, Integrated Inertial and Vision Tracking for Augmented Reality Registration, IEEE Virtual Reality Conference (VR99) (1999).