

コンピュータディスプレイ上の自律移動型の 半透明ビジュアルマーカ

岸野 泰恵[†] 塚本 昌彦[‡] 坂根 裕[‡] 西尾 章治郎[‡]

[†] 大阪大学工学部

[‡] 大阪大学大学院工学研究科

概要：拡張現実空間を実現するためには，実空間を撮影しているカメラの位置や方向を正確に知る必要がある．最近では，見かけ上の位置精度が高くなるという理由から，画像処理を用いた研究が注目されている．これらの研究では，実空間内に貼り付けた紙マーカをカメラで撮影し，カメラ画像を解析するのが一般的であるが，マーカの識別が困難であることや，位置の変更は手で行わなければならないという問題があった．筆者らはコンピュータディスプレイにマーカを表示し，一定時間間隔でマーカの色を変化させて，位置情報と共に数十バイトの情報を表現できるマーカとして VCC (Visual Computer Communication) 方式を提案した [10]．従来の VCC では，マーカをディスプレイに表示すると邪魔になり，コンピュータディスプレイを有効に使えないという問題があった．本稿では，この問題を解決するため，マーカを半透明にする，自律的に移動できるようにするなど VCC を拡張し，その性能の評価を行った．

A Translucent Visual Marker with Autonomous Mobility on Computer Display

Yasue KISHINO[†], Masahiko TSUKAMOTO[‡], Yutaka SAKANE[‡] and Shojiro NISHIO[‡]

[†] Faculty of Engineering, Osaka University

[‡] Graduate School of Engineering, Osaka University

Abstract : In order to realize an augmented real space, it is necessary to know the accurate viewing position and the orientation of the camera used to take the image of a real world. Recently, the vision-based tracking technique attracts a lot of attention from researchers for the reason that this method can minimize the visual alignment error. Generally, these systems apply this technique to get the information of camera position and orientation by analyzing the camera images in which there are paper-printed markers placed in the real world. However, this method has two serious problems. One of the problems is that it is difficult to distinguish a marker from other objects. The other problem is that when a user wants to move a marker, the user has to do it physically. In order to solve these problems, we proposed the VCC (Visual Computer Communication) method [10], in which a marker is displayed on a computer screen, and the color of each component of the marker is changed within a certain time interval. Although the location information as well as several bytes of data can be obtained by a computer with a camera by using this system, the previous VCC has a problem: the marker displayed on a computer screen occupies the screen area and becomes an obstacle for the user to utilize the computer screen. Therefore, in this paper, we extend VCC to make a marker translucent and to enable a marker to move autonomously. In addition, we evaluate the performance of the extended VCC.

1 はじめに

近年，仮想物体を現実空間の画像に重ね合わせて提示する，拡張現実空間に関する研究が盛んに行われている [2]．これらの研究では，正しい位置に仮想物体を表示するために，現実空間を撮影するカメラの位置や方向を正確に検出することが重要になる．このような情報を取得する研究として，

GPS や地磁気センサなどのセンサ類を用いる研究が数多く行われている [3, 5]．さらに最近では，画像処理を用いて位置情報を得る研究が注目されている [7]．画像処理を用いると，位置情報を求める際に用いる画像と同じ画像をユーザが使用するため，見かけ上の位置精度を高くできる．この手法では，マーカを実空間のさまざまな場所に貼り付



図 1: VCC 利用状況の例

け、カメラで撮影したマーカの大きさや傾きなどからカメラの位置や方向を算出する。従来研究では、マーカは紙に印刷したものをを用いるのが一般的であり、そのためマーカの移動は手で行わなければならなかった。さらに、マーカを識別するために、マーカの中に数字や 2 次元のバーコードであるマトリックス [8] を印刷する研究もあるが、マーカで表現できる情報は数バイト程度であり、また動的に内容を変更できないという問題があった。

そこで、筆者らはコンピュータディスプレイ上にマーカを表示し、一定の時間間隔で表示色を変化させて任意の情報を表現できる VCC (Visual Computer Communication) 方式を提案した [10]。VCC を用いることで、マーカを写したカメラ画像から、マーカとカメラの位置関係が得られると同時に、マーカから理想的には任意長のデータが得られる。

本研究では、VCC の利用環境として、生活空間のいたるところにコンピュータが存在し、誰でも自由に利用できるユビキタスコンピューティング環境を想定している。街角に設置されたタッチパネルや、電車の中や駅前にあるようなディスプレイにマーカを表示し、マーカの位置情報とともに、近くに存在する店の URL など、場所に依存した情報を手軽に提示できる。現状では、街角やオフィスに VCC のマーカを自由に表示できるようなディスプレイは存在しないため、VCC を利用するためには通常のコンピュータデスクトップに用いられているコンピュータディスプレイにマーカを表示することが考えられる。しかし、従来の VCC では、ディスプレイ上にマーカを表示すると、画面の大部分を占有し、他の情報を隠してしまう、画面内の移動はユーザが明示的に行う必要があるなどの問題があった。

本稿では、これらの問題を解決するような、目立たなく邪魔にならないマーカを実現するため、マーカを半透明にした。さらにディスプレイ上にある



図 2: 仮想物体の合成例

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	1バイト 送信	ロケータ

図 3: コーディングの例

他の情報を隠さないように、使われない領域へ自律的に移動するようマーカを拡張した。

2 VCC

図 1 では、左奥にあるコンピュータディスプレイ上にマトリックス状のカラータイルを点滅させ、それを手前のノートパソコンに付属した CCD カメラで撮影している。手前のノートパソコンはマーカの色の変化を撮影し、カメラ画像を解析してマーカとカメラの相対的な位置関係を算出する。さらに同時に、マーカが表現している情報を読みとる。

図 2 に、画像処理によって得たマーカの位置と傾きの情報をもとに、マーカから読み取ったファイル名の仮想物体を合成した例を示す。その他にも、商店街などで店頭にあるディスプレイに表示されたマーカから URL を読み取り、ホームページを表示する、人が身につけたマーカから E-Mail アドレスを読み取る、会議の際にスクリーンに投影したマーカから資料ファイルのあるパスを知るといった利用方法が考えられる。

2.1 マーカのコーディング

マーカのコーディングの一例として、図 3 に示すコーディング方法を説明する。本稿では、図のような 4×4 の 16 個のブロックに分割したマトリックス状のマーカを利用する。このマーカを一定時間間隔で変化させ、色の変化の差分を用いてデータを表現する。4 隅のブロックをロケータ部と呼び、対角線上の 2 つが組になって一定時間間隔で交互に白黒と点滅することでマーカの境界を示す。

ロケータ部以外のブロックをデータ部と呼ぶ．第1ビットから第8ビットの8個のブロックをコーディングするのに1色を用いれば，一度の色の变化で1バイトの情報を，2色でコーディングし，2色が重なった色も含めて合計3色を用いれば2バイトの情報を表せる．1色コーディングのときには，1バイト送信ビットを反転させる．パリティビットはエラーのチェックに利用する．データの送信を開始，終了するときには，それぞれ送信開始ビット，送信終了ビットを反転させる．

2.2 マーカの検出方法

受信側のコンピュータは，マーカの変化と同じ間隔で画像を取り込み，前のフレームで取り込んだカメラ画像との差分画像を作成する．これを適当な閾値で色ごとに二値化し，二値化した画像から連結白領域を取り出し，マーカの4隅を検出する．検出したロケータ部をもとにマーカ内の座標を計算し，色情報を読み取る．色情報からマーカの表現しているデータを得る．さらに，画像中のロケータ部の座標を用いてマーカとカメラの間の距離や傾きを求め，法線ベクトルを算出する [4]．

3 VCCの拡張方針

本研究では，従来のVCCに対し，以下のような方針で拡張した．

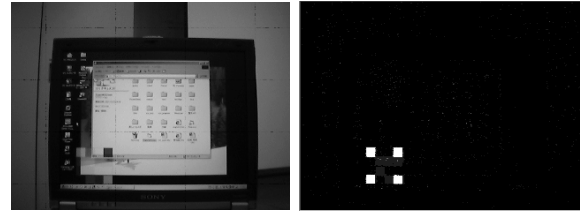
- マーカを半透明にする．
- マーカをユーザが使用していないディスプレイ領域へ移動させる．

さらにこれらを組み合わせることで，より目立たなく，邪魔にならないマーカが実現できる．

3.1 マーカの半透明化

マーカを半透明にすることで全体的に色が薄くなり目立ちにくくなる．さらに，これまで不可能であったマーカの下に隠れている部分が見れるようになる．デスクトップ上にマーカを表示した場合では，マウス情報を透過させることで，マーカごしのウィンドウ操作が可能になる．

図4(a)は半透明なマーカを撮影した様子である．ディスプレイ左下にマーカを表示しているが，あまり目立たないことが確認できる．しかし，適切な閾値を選択することで，図4(b)のような見た目よりはっきりした差分画像が得られ，従来のアルゴリズムでも正しくマーカを検出できる．



(a) 元の画像 (b) 差分画像

図 4: 半透明にしたマーカの例

3.2 マーカの移動

マーカをディスプレイ上で使用しない領域に表示すれば，それほど邪魔にならないことが想定される．通常のコンピュータデスクトップの場合に，使用しない領域がどこであるか推定するための条件として，次のようなものが挙げられる．

- ユーザによる指定
- マウスカーソルの位置
- アクティブウィンドウの位置
- キャレットの位置
- 文字や画像の位置

これらの条件の中で，文字や画像のある位置を検出するためには，デスクトップ画像をキャプチャし，画像処理を行う必要がある．しかし，今回は実時間でマーカの表示位置を決定する必要があり，画像処理に時間がかかるこの方法は用いない．そこで，実時間で処理でき，自律的にマーカの位置を変更できるよう，マウスカーソルの移動履歴とアクティブウィンドウの位置とサイズ，キャレット位置の3つの条件から現在使用されていないディスプレイ領域を推定することにした．

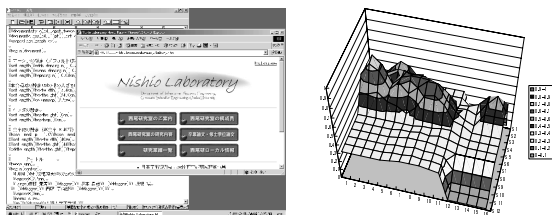
処理を簡単にするため，ディスプレイ領域をいくつかのブロックに分割し，それぞれのブロックについて使われる可能性 P を求め，この値の低い位置にマーカを表示する．

マウスカーソル： P_m

時刻 t における P_m を P_m^t とし， Δt ごとに利用可能性を計算する．これまでの履歴のうち，最近の情報を重視するように，

$$P_m^t = w P_m^{t-\Delta t} \quad (0 \leq w \leq 1)$$

と w をかけ，時間が経つにつれて P_m を単調に減少させる．マウスカーソルと重なっているブロックには，使用可能性を高くするため， ΔP_m を，その8近傍のブロックには $\Delta P_m/2$ を加える． ΔP_m は，0 から 1 の値をとるように， $P_m > 1$ のときは $P_m = 1$ とする．



(a) デスクトップ (b) 利用可能性
図 5: 利用可能性の計算結果

アクティブウィンドウ: P_w

P_w は, P_m と同様に, w によって一定時間間隔で単調に減少し, アクティブウィンドウに覆われているブロックには ΔP_w を加え, アクティブウィンドウと一部重なるブロックには $\Delta P_w/2$ を加える.

キャラット: P_c

P_c は, P_m と同様に, w によって一定時間間隔で単調に減少し, キャラットが乗っているブロックには ΔP_c を, その 8 近傍のブロックには $\Delta P_c/2$ を加える.

この P_m, P_w, P_c を適当な割合で加えて P を算出する. それぞれの割合を表す変数 r を用い, $P = r_m P_m + r_w P_w + r_c P_c$ とする ($r_m + r_w + r_c = 1$). あるブロックが使用される可能性は P が 1 に近いほど高く, 0 に近いほど低いことを意味する.

例えば, 図 5(a) のようにウェブページを参照しながら文章を書くとき, 使用可能性は, P_m, P_w, P_c を同じ割合で加えて算出すると, 図 5(b) のようになる. 図 5(b) では使用可能性 P を高さ軸にしており, 交互にアクティブになる 2 つのウィンドウの部分が全体的に高くなり, それに加えてマウスカーソルやキャラットのあった部分が高くなっている.

マーカ検出の画像処理には差分画像を用いているため, 突然マーカが離れた位置に移動すると正しく検出できない. そこで, どこにマーカを移動させるかが決まれば, 次にマーカの移動経路と速度が問題になる. 移動する方向は, 移動先への最短経路全てについて, 経路上の使用可能性を足し合わせ, 合計が最小の方向とする. 経路は一定距離進むごとに再計算し, マーカは常に最小の利用可能性の部分へ移動しようとする.

マーカが移動する様子を図 6 に示す. これはマウスカーソル(図中右上の曲線), アクティブウィンドウ(右側の長方形), キャラット(右中央の直線)の位置からマーカの表示位置(左上の正方形)を, 実

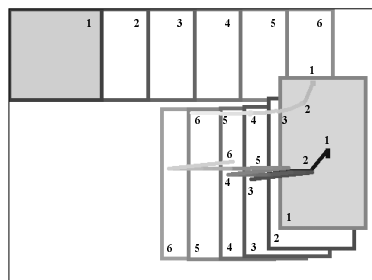


図 6: マーカが移動する様子

際にデスクトップ上で作業した場合の情報を元に, シミュレートした結果である. 画像中の数字は時間の変化を意味し, 現在の位置を 1 とした. アクティブウィンドウが移動するとマーカがそれを避けるように移動していることが確認できる.

4 実装

マーカをディスプレイ上に表示するプログラムは, ソニー社バイオノート PCG-C1XF (Pentium2 400MHz, メモリ 128MB) で Windows98 上に VisualBasic6.0 を用いて実装した. 8.9 型の液晶ディスプレイ上に解像度 1024×480 ピクセル, 色は RGB24 ビット(各色 8 ビット)を用いてマーカを表示した. 各色の階調は 256 である.

マーカを検出するプログラムは, ソニー社バイオノート PCG-C1VJ (Crusoe 600MHz, メモリ 128MB) で, WindowsMe 上に VisualBasic6.0 で実装し, 画像処理には VisualC++ を用いた. カメラはこのノートパソコンに付属している CCD カメラを使用した. カメラは 1 画素を RGB24 ビットで表した 160×120 ピクセルの大きさの画像を撮影する.

4.1 マーカの半透明化

ディスプレイ上に, 透明度 α を用いて, 全てのピクセルで色ごとに以下のように画素値を計算し, 半透明のマーカを重ね合わせた. RGB それぞれの色は g_R, g_G, g_B 階調で表示されるものとする.

$$D = s \times \left(1 - \frac{\alpha}{g_i - 1}\right) + d \times \frac{\alpha}{g_i - 1} \quad (i \in R, G, B)$$

ここで, D はディスプレイに表示する画素値, d はマーカの下に表示されている背景の画素値, s は重ね合わせるマーカの画素値である. 実装した環境では RGB いずれも 256 階調なので, α が 255 のときはマーカは全く見えず, α が 0 のときはマーカのみが表示されて背景は全く見えないことになる.

これまで VCC で情報が正しく得られない場合は, ロケータ部とデータ部の区別ができずロケー

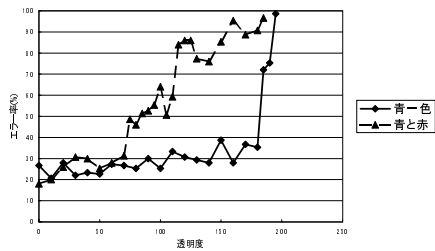


図 7: 透明度とエラー率の関係(青色の背景)

タ部の検出に失敗していることが多かった．そこで，ロケータ部をデータ部より少し濃く表示し，読み取り精度の向上を図ることにした．差分画像において，ロケータ部がはっきり読み取れ，検出が容易になるので，閾値を厳密に選ぶ必要がなくなる．

4.2 マーカの移動

実装には，解像度 1024×768 ピクセルで表示できる 12.1 型のディスプレイを持つソニー社バイオノート (PCG-R505) を用いた．マーカの移動速度は 1 つのマーカの幅の $1/4$ とした．使用可能性 P は，画面上を 16×12 のブロックに分けて算出した．マーカの大きさは 256×256 ピクセルとした．

5 評価実験

5.1 性能

拡張した VCC の評価実験を行った．実験では IP アドレスを想定したランダムな 4 文字の文字列を用いてエラー率を測定した．マーカが表示する情報をカメラで撮影して解析して，得られた文字列が 1 ビットでも誤っていた場合をエラーとし，その割合をエラー率とした．マーカの色の変化とカメラで画像を取り込むタイミングによって精度が変わるため，1 つの文字列を表示し終わるたびにランダムな時間マーカを停止させた．マーカとカメラの間の距離や角度とエラー率の関係についてはすでに実験を行っており [6]，この実験において最もエラー率が低かった 60cm の距離にマーカとカメラを設置し，コーディングには青一色と，青と赤を用いた．マーカの色が変化する間隔は 200 ミリ秒とした．カメラ画像の色は RGB それぞれ 1 色につき 256 階調で表され，画像処理における閾値は 0 から 255 の間の値を取る．実験は屋内で行った．

5.1.1 透明度と背景の影響

マーカを半透明にしても，正しく検出できるか調べるため，透明度とエラー率の関係を調べた．背景は無地で青色と緑色を用いた．透明度を 0 から

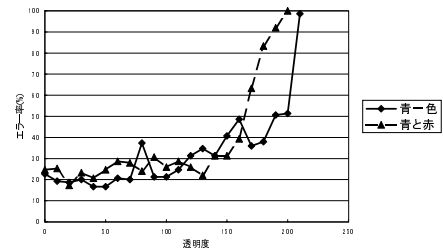


図 8: 透明度とエラー率の関係(緑色の背景)

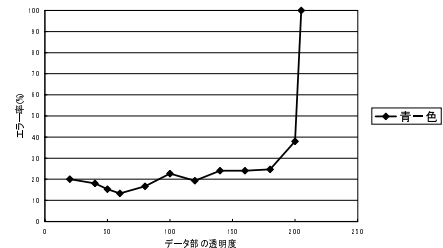


図 9: 透明度とエラー率の関係(ロケータ部を濃く表示した場合)

次第に上げていき，どこまで上げられるのか調べた．マーカのコーディングは青一色と，青と赤を用いた場合について実験した．閾値は透明度を変化させるごとに最適な固定の値を手動で選択した．

結果は，背景が青色の場合が図 7，背景が緑色の場合が図 8 となった．これらのグラフより，透明度を上げていくと青と赤を使った場合は，図 7 では透明度 80 を，図 8 では透明度 150 を超えると，エラー率が高くなり，青一色でコーディングした場合は，どちらの背景でも透明度 180 を超えると，エラー率が高くなることが分かった．

ロケータ部をデータ部より濃く表示した場合の実験結果を図 9 に示す．ロケータ部の透明度はデータ部よりも 20 下げた値を用いた．結果はマーカを青一色でコーディングした場合のものである．背景は緑色にした．ロケータ部をデータ部より濃く表示したことで，図 8 と比較すると，透明度 120 から 180 では 10% 以上エラー率が低くなっている．

5.1.2 マーカの移動とエラー率の関係

マーカの移動中もマーカを認識できるか調べるため，1 フレームでのマーカの移動距離とエラー率の関係を調べた．閾値は以前の実験で最もエラー率の低かった赤 100，青 150，緑 150 に固定した．

実験結果は図 10 のようになった．横軸が 1 フレーム (200 ミリ秒) におけるマーカの移動速度で，単位はピクセルである．1 フレーム間におけるマーカの移動距離が 1 ブロックの幅の約 $1/3$ である 25 ピクセルより小さい場合は，移動しない場合と比較し

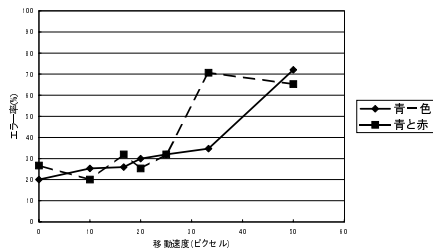


図 10: マーカの 1 フレーム間における移動距離とエラー率の関係

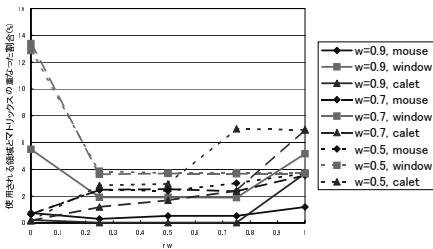


図 11: マーカと作業領域が重なった割合

てエラー率が同程度であるが、それよりも大きくなるとエラー率が高くなることが分かる。

5.2 マーカの自律的な移動

マーカを自律的に移動させた場合に、使用している領域とマーカがどれだけ重なるかを確認する実験を行った。約 1 時間 PC (解像度 1024×768 ピクセル) を使用し、1 秒おきにマウスカーソルとアクティブウィンドウ、チャレットの位置を取り、マーカの表示位置を算出した。

実験結果は図 11 のようになった。 $\Delta P = 0.5$ とし、 $r_m = r_c$ とした。横軸は r_w を表し、縦軸はマーカとマウスカーソル、アクティブウィンドウやチャレットが重なった割合を表す。アクティブウィンドウがデスクトップである場合や、マーカを表示する余地がない程大きい場合は、マーカをどこに表示しても重なるので、そのような場合は除いている。

結果より、3 つの条件を合わせて使用可能性を求めると、マウスカーソル、アクティブウィンドウ、チャレットのいずれともマーカがあまり重ならないことが分かる。また r や w を調節することで、重なる割合を低く抑えられることが分かった。

5.3 拡張したマーカのユーザによる評価

マーカをディスプレイに表示し、半透明にしたり、自律的に移動させた場合、ユーザがどのように感じるかを調べた。実験は、マーカが不透明の場合と半透明の場合、固定の場合と自律的に動く場合を組み合わせ、不透明固定、不透明移動、半透明固定、半透明移動の 4 通りについて行った。それぞ

表 1: マーカの状態で手動で移動させた回数

マーカの状態		平均移動回数
不透明	固定	7.6 回
不透明	移動	3.6 回
半透明	固定	3.2 回
半透明	移動	1.6 回

れの場合について約 15 分間マーカを表示し、実験中にマーカが邪魔になり、何回手動で移動させたかを調べた。マーカの色が 4 回変化することにより、マウスカーソル、アクティブウィンドウ、チャレットの位置を調べた。被験者はパソコンを日頃からよく使っている 20 代の男性 5 人であり、普段どおりにコンピュータを使用してもらった。

結果は、表 1 のようになった。結果より、不透明より半透明の方が、固定より自律移動の方が移動させる回数が減っている。マーカを提案する方法で拡張することは、ユーザの作業の邪魔になることを防ぐ上で効果があるものと考えられる。

6 考察

6.1 マーカの半透明化と精度

実験結果から青一色でコーディングした場合は、透明度を 180 程度まで上げられることが分かった。透明度が 180 より大きくなると、全体的にマーカの色が薄くなり、検出や色情報の読み取りが困難になる。青と赤を用いた場合、透明度の高いところでは、赤が非常に暗くなり赤の色情報が読み取れていなかった。背景の色が緑の場合の方が青の場合よりもエラー率が低く、背景で用いられていない色でマーカをコーディングした方がエラーが少なくなることが分かった。ロケータ部をデータ部より少し濃く表示した図 9 では、そうでない図 8 よりエラー率が少し下がっている。差分画像においてデータ部が全体的に暗くなり、ロケータ部の検出が容易になるためである。この拡張によって、ロケータ部の透明度を 20 下げても見た目はほとんど変わらないが、精度が向上することが分かった。

今後、背景の色から動的にマーカに使用する色を選ぶことで、エラーを少なくしたい。また、影響を受けにくい背景の領域を選んでマーカを表示する方法も、エラーを減らすために有効である。

マーカを半透明にすることで、背景が激しく変化したときに、差分画像にその変化が現れ、マーカを検出できない可能性が生じる。マーカの形状が正方形から大きく歪んでいる場合、読み取った

表 2: データの繰り返し受信回数とエラー率

繰り返し回数(回)	1	3	5	7	10
エラー率(%)	26	21	3	0	0

結果は無視するなどの対策が必要になる。

6.2 マーカの移動と精度

実験結果から 1 フレームにおける移動の幅がマーカの 1 ブロックの $1/3$ よりも大きくなると、エラー率が高くなることが分かった。画像処理に差分画像を使っているため、マーカの移動が速すぎ前回のフレームと重なる部分が少なくなると、マーカの検出が困難になるためである。

この実験により、マーカがどの程度の速度で移動できるのか分かったので、マーカの移動中に、その内容を読み取るような応用が可能になった。

6.3 VCC の性能

現在の VCC ではエラー率が低い部分で、20%程度である。この程度の精度ならば、数回繰り返し同じデータを読み取り、何回分かのデータを比較することで、短い情報ならばマーカから正しく読み取れる。表 2 は、4 バイトの同じデータを繰り返し読み取り、その中で最も数の多いデータを正しいデータとした場合に、どの程度エラー率が下がるかを実験した結果である。表 2 に示すように、約 $1/4$ のエラーがある場合でも 5 回繰り返し読み取れば、ほとんどエラーがなくなることが分かる。

VCC で利用を想定しているのは URL や IP アドレスといったデータである。4 バイトの IP アドレスを 5 回繰り返すのには 4 秒かかる。マーカを数秒間カメラの前で静止させるようなアプリケーションでは、この程度の性能でも十分だと考えられる。

6.4 マーカのコーディング

本稿で説明したコーディング方法では、マーカの一部が隠れる場合や、マーカの上下が入れ替わった場合には対処できない。これらの問題点を解決するようなコーディング方法も考えられるが、実際のデータ以外の情報が増えるため、1 度の点滅で伝える情報量を現在と同等にするならばマーカのサイズが大きくなる、マーカのサイズを現在と同等にするならば一度の点滅で伝えられる情報量が減るなどの問題があり、本稿では扱っていない。

6.5 マーカの自律的な移動

実験結果より、提案手法を用いると、マウスカーソルやアクティブウィンドウ、キャレット位置とそ

れほど重ならずに、使用しない領域へマーカを移動できることが分かる。しかし、アクティブでないウィンドウを参照しながら、別のウィンドウで文章を書くような場合、参照しているウィンドウの上にマーカが移動する可能性があるため、 w や ΔP の値を調節する必要がある。常にウィンドウをフルサイズにするユーザの場合、マウスカーソルとキャレット位置のみからマーカの表示位置を決定するため、邪魔になる位置にマーカが移動する可能性がある。この場合、文字や画像の場所情報を利用することが必要になる。文字や画像はユビキタスディスプレイにおいてもユーザが特に注目する部分になると考えられる。

6.6 ユーザによる評価

5.3 節で述べた実験の後に被験者にマーカに関するアンケートを行った結果、半透明化については、肯定的な感想が得られたが、自律的な移動については、特に文章を読んでいるときに、注目している部分をマーカが横切ると邪魔になるという意見があった。これに対しては、文字や画像の場所情報を利用すれば、改善できると考えている。また、4 隅のみにマーカを表示したいという意見もあり、ディスプレイの用途によるアルゴリズムの変更が必要であることが分かった。さらに、マーカが動くことによって集中力が失われるという指摘もあり、マーカでデータ中断の情報を伝えて、使用可能性の低い位置に移動させた後、再開するなどの対策が必要になる。その他に、タスクバーの上にマーカがあると特に邪魔だという意見もあった。使用可能性 P_w を求めるときに、タスクバーの位置と大きさも参考にするなど、ユビキタスディスプレイ上で常に見えていることが重要なコンテンツとマーカが重ならないようにする必要がある。このようなコンテンツとしては、ヘルプを起動するためのアイコンなどが考えられる。

6.7 ハイブリッドなセンサ

拡張現実感の分野で、位置情報を得るための方法として、画像処理とセンサ、あるいは複数のセンサを組み合わせた研究が行われている [9, 11]。VCC でも他のセンサと組み合わせることで位置情報の精度を向上させたり、センサでカメラの移動を測定し画像中のマーカ位置を予測することで、画像処理速度を向上させることが期待できる。

6.8 関連研究

関連研究として, Cyber Code[8], 赤外線 LED を用いる Balloon Tag[1] などがある. Cyber Code では紙のマトリックス状のマーカをカメラで撮影し, 画像を解析して ID を読み取り, 対応するコンピュータ上の情報を利用できる. 紙のマーカを用いるので VCC よりも手軽にマーカを利用できるが, 表現できる情報量は数バイト程度である.

Balloon Tag では赤外線 LED を撮影したカメラの画像からタグの位置を解析し, その情報を読み取る. 本稿では, マーカを目立たなくするために, 半透明にしたが, Balloon Tag では目に見えない赤外線領域を用いている. 比較すると, クロック信号を利用するのでデータ通信速度によらない情報送受信が可能であり, 目立たないという利点があるが, データ送信速度が遅いという問題がある. さらに, アプリケーションで用いるカラー画像は別に取得する必要がある. ユーザにとってもマーカを認識できず, カメラでマーカを捉えにくい.

また, 電波無線タグを用いる研究[12]も, 目立たないという利点があるが, タグと受信機間の明確な距離や角度が取得できないという問題がある.

7 まとめ

本稿では, 半透明で自律的に移動するビジュアルなマーカを実現した. さらに, 評価実験を行い, その性能を調べた. 今回提案した方式は, 主に通常のコンピュータデスクトップを利用してマーカを表示することを想定するものであり, 将来ユビキタスディスプレイを利用できるようになるまでの間は特に有効であるものと考える.

今後の課題として, 画像処理に用いる閾値を自動的に選択する手法の実現や, 読み取りの性能を向上させることが挙げられる. 性能を向上させるためには, 例えばカメラで画像を撮影する速度を2倍以上にする必要がある.

謝辞

本研究の一部は, 日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(プロジェクト番号: JSPS-RFTF97P00501), 文部科学省科学技術振興調整費研究課題「モバイル環境向 P2P 型情報共有基盤の確立」, および, 文部科学省特定領域研究(C 013224064)によっている. ここに記して謝意を表す.

参考文献

- [1] 青木 恒: “カメラで読み取る赤外線タグとその応用,” インタラクティブシステムとソフトウェア VIII (日本ソフトウェア科学会 WISS 2000), 近代科学社, pp.131–136 (2000).
- [2] Azuma, R.: “Survey of Augmented Reality,” Presence: Teleoperators and Virtual Environments, Vol.6, No.4, pp.355–385(1997).
- [3] Azuma, R., Hoff, B., Nelly III, H., and Sarfaty, R.: “A Motion-Stabilized Outdoor Augmented Reality System,” Proc. of IEEE VR '99, pp.252–259 (1999).
- [4] 出口 光一郎: “センシング / 認識シリーズ第 5 巻 画像と空間,” 昭晃堂 (1991).
- [5] Höllerer, T., Feiner, S., and Pavlik, J.: “Situating Documentaries: Embedding Multimedia Presentations in the Real World,” Proc. of Third International Symposium on Wearable Computers (ISWC '99), pp.79–86 (1999).
- [6] 岸野 泰恵, 塚本 昌彦, 坂根 裕, 西尾 章治郎: “コンピュータディスプレイを用いたビジュアルマーカの実現,” 情報処理学会シンポジウムシリーズ マルチメディア, 分散, 協調とモバイルシンポジウム, Vol.2001, No.7, pp.259–264 (2001).
- [7] Park, J., Jiang, B., and Neumann, U.: “Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking,” IEEE International Workshop on Augmented Reality, pp.3–12 (1999).
- [8] Rekimoto, J. and Ayatsuka, Y.: “CyberCode: Designing Augmented Reality Environments with Visual Tags,” Proc. of ACM Designing Augmented Reality Environments (DARE2000) (2000).
- [9] State, A., Hirota, G., Chen, D. T., Garrett, B., and Livingston, M.: “Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking,” ACM SIGGRAPH '96, pp.429–446 (1996).
- [10] 塚本 昌彦: “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,” 情報処理学会研究報告 (モバイルコンピューティング研究会 2000-MBL-12) Vol.2000, No.14, pp.25–32 (2000).
- [11] You, S. and Neumann, U.: “Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration,” IEEE Virtual Reality 2001, pp.71–78 (2001).
- [12] Want, R., Fishkin, P. K., Gujar, A., and Harrison, L. B.: “Bridging Physical and Virtual Worlds with Electronic Tags,” Proc. of ACM Conference on Human Factors in Computing Systems (CHI '99), pp.370–377 (1999).