マルチメディア通信と分散処理　90－9
コンピュータセキュリティ　2－9
（1998．9．18）

レイアウト構造を利用したページ記述への電子透かし埋め込み手法

天野富夫, 平山唯樹
E-mail: amano@jp.ibm.com, hirayama@jp.ibm.com

日本アイ・ビー・エム東京基礎研究所

あらまし
ディジタルライブラリーや情報配信サービスにおいて流通するコンテンツの大部分は文書であり,文書データへの電子透かし埋め込み技術への需要は大きい.本論文では文字の間隔を変化させて文書のページ記述に透かしを埋め込む手法について報告する．提案手法では，ページ記述内の文字オブジェクトをそのレイアウト構造上で順序付けし，オリジナル文書を必要としない検出や統計的手法の適用を可能にしている．また、文字間隔を比較するさいにはフォントのサイズや各文字の縦横比を考慮した正規化を行ない，使用フォントやテキストの内容の異なる文書に対して同じように埋め込み/検出ができるようにしている．代表的なページ記述であるPDF(Portable Document Format)を対象として本手法をインプリメントした．処理できるオペレータやフォントの種類は限られているものの,本手法の実現可能性を確認することができた.

A method for embedding digital watermark in page descriptions

Tomio Amano and Yuki Hirayama

IBM Japan, Ltd., Tokyo Research Laboratory

Abstract
This paper proposes a method for embedding/detecting a watermark in document page descriptions such as PDF (Portable Document Format). The method uses normalized intervals between two succeeding characters. A watermark can be encoded as slight changes in these intervals. A page layout structure is constructed from the page description and used to order character objects; so that a watermark can be detected without original data. We have implemented a embedding/detecting program which can deal with small set of PDF operators, and investigated the feasibility of our method.

1 はじめに

インターネットやCD-ROMの普及によってディジタルコンテンツを容易に配布,流通させることが可能になった.このことは,インターネットを利用した情報配信サービスなどの新たなアプリケーション/ビジネスの発達を促す反面,コンテンツの不正な複製や改ざんが行われる危険性も増大させた.不正行為を防止する手段としてコンテンツデータ自体に権利等の情報を埋め込む電子透かし技術が注目されている.

電子透かしはコンテンツデータと非可分なかたちで埋め込まれるため通常の流通過程において消去されることはない.コンテンツから透かし情報を検出することにより,著作権者や流通経路(誰が複製を行なったのか)を同定できるため,ディジタルコンテンツの不正利用を抑止する効果があると期待される.また,正しい透かしが埋め込まれているか否かによってコンテンツが改ざんされていないことを他者に保証する[1],レーティング情報を埋め込んでおいてブラウザー/プロキシー側でスクリーニングを行なう(例えば成人向けコンテンツの表示を制限する),等の利用法も検討されている.

ディジタルライブラリーや情報配信サービスにおいて流通するコンテンツの大部分は文書データであり,静止画や動画と並んで文書データへの透かし埋め込み技術が必要である.著者らは特に,文書の「見えかた」を規定するページ記述への透かし埋め込みが重要だと考えている.多種多様なフォーマットで作成された文書がページ記述に変換された後,配布される場合が多いからである.テキストのコード列自体は冗長度が低く透かしの埋め込みは困難であるがページ記述には文字の大きさや位置等のデータが含まれるためそれらの冗長性を利用して透かし情報を埋め込むことが可能である.

本論文では,代表的なページ記述フォーマットであるAdobe社のPDF(Portable Document Format[2])に透かしを埋め込む手法について報告する.PDFはプリンタドライバを経由することによりほとんど全ての文書フォーマットから変換することができ,閲覧用ソフトウェアも無料で提供されている.そのためインターネットやCD-ROMによるページ記述文書の配布形態として広く用いられている.PDFのかたちで書籍や文書情報を販売する電子商店も既に存在する.したがって,ページ記述への透かし埋め込みの具体的対象としてPDFへの埋め込み/検出の実現可能性を検証することの意義は大きいと考える.

本論文では,まず2節でPDFへの透かし埋め込みに関する要件を整理し従来手法を概観する.3節では提案埋め込み手法を,埋め込みの基本単位,その順序付けの方法,統計的手法を用いたビットの埋め込み/検出の順で説明する.4節で実際の埋め込みの例をしめし,実際の文書における特徴の変化の検出方法の妥当性を検証する.5節では,求められている要件に提案手法がどの程度対処しているか考察を行い,6節でまとめと将来の展望を述べる.

2 ページ記述への電子透かしの要件

不可視な電子透かしが満たすべき基本的な要件としては,埋め込みによってコンテンツの外観が不自然にならないこと,アプリケーションが必要とするだけの容量の情報を埋め込めること,が挙げられる.さらに実際の運用を考えた場合には,以下の3つの要件も考慮する必要がある.

(1) オリジナル文書無しでも透かしの検出が可能:透かしはコンテンツ内のなんらかの特徴値の変化として記述される.この変化を見つける最も簡単な方法は埋め込み前のオリジナルコンテンツと検出対象を比較することである.しかし,この方式では透かしを埋め込んだ側で将来の検出のために全てのオリジナル文書を保持している必要があり,電子透かしシステムの運用上大きな負担となる.適当な「キー」を所有していれば複数の文書データに対して検出処理が行える埋め込み方式が望ましい.

(2) 日常的な操作に対する耐性:透かしは流通過程でうける通常の操作に対して頑健である必要がある.PDFに対するこの種の操作としてはページ単位での抜き出しやフォーマット変換(PDF→PostScript→PDF)がある.コメントやアノテーションの機能を利用した透かしはフォーマット変換により消去されてしまうため,より文書データの内容に結びついたかたちで透かしを埋め込む必要がある.改ざん防止に透かしを用いる場合はこのレベルの耐性があれば利用可能である.

(3) 意図的な攻撃に対する耐性:著作権管理等のアプリケーションに用いる場合には意図的に透かしを消去・破壊しようとする攻撃に対しても頑健である必要がある.

ページ記述への電子透かし埋め込みについては既にいくつかの提案が行なわれている.渋谷ら[3]はPostScriptやPDFの表現の多様性に着目し透かしを埋め込むことを提案しているが,当該論文では具体的

なインプリメントまでは行なっていない.特許公報[4, 5]にはページ記述内の文字や行の位置情報を利用して透かしを埋め込む手法が開示されている.しかし,上記の方法は[5]の文字行のベースラインを操作する方法を除いて,透かしの検出に関してはオリジナル文書と比較する方式を想定している.

3 PDFへの透かし埋め込み手法

本節では,埋め込みプリミティブとプリミティブの順序付け[6]という二つの観点からPDFへの透かし埋め込み手法を概説する.提案手法では,埋め込みプリミティブとして連続する二つの文字の間隔を用い順序付けの方法として各文字をページのレイアウト構造にマッピングしたときの位置(何行目の何文字目か)を利用している.順序付けを行うことによって,オリジナル文書を必要としない透かし検出,統計的手法の導入による頑健性の獲得などのメリットが得られる.

3.1 埋め込みプリミティブ

埋め込みプリミティブとは透かし埋め込み処理のためになんらかの特徴値を変化させる単位である.ページ記述内には文書の外観を規定する文字やフォント・線分などのオブジェクトが含まれているが,われわれは連続する2文字間の間隔–ある文字の左端から次の文字の左端までの水平距離–を埋め込みプリミティブとして用いている.

水平距離の値はフォントの種類やサイズに影響され,また文字コードによっても変化する(例えば"l"のような細長い文字が並んでいる場合と"m"や"w"が並んでいる場合とで値は異なってくる).使用フォントや内容の異なる文書間でも同一の基準で比較ができるよう,埋め込みおよび検出にはフォントごと文字ごとの標準的な幅で正規化した間隔(正規化間隔と呼ぶことにする)を用いている.

文字は一般文書のページ記述において最も多く出現するオブジェクトであり,文字行や線分を埋め込みプリミティブに使う場合に比較して多くの情報を埋め込むことができる.文字(列)の位置の記述は多くのページ記述フォーマット中に存在するため,PDF以外への埋め込みにも適用が期待できる.

3.2 プリミティブの順序付け

埋め込みプリミティブの集合を用いて透かしをコード化するためには,プリミティブを順序付けする適当な方法が必要である.透かし検出時にあるプリミティブの変化が観測されたとき,どんな意味があるのか解釈するためにはそれが埋め込み時に使われたプリミティブ集合のどの要素なのか識別しなくてはならないからである.オリジナル文書と比較するということは,ファイル内でのオブジェクトの出現順によってプリミティブを識別し「n番目の文字と次の文字の間がひらいているからビット1が埋め込まれている」といった判定を行なうことを意味する.しかし,ファイル内での出現順による識別は文書データごとに固有のものとなってしまい前節で述べたようなオリジナル文書の管理という運用上の問題が発生する.さらに,同じ外観が得られるならばファイル中におけるオブジェクトの記述の順番は任意でよいため,フォーマット変換や透かしへの攻撃によってその順番が変更されるかもしれない.埋め込みを行なうためには同じ外観を与える複数のページ記述に対してもオブジェクトの順序が一意に決められるようにしておく必要がある.

ファイル内での出現順にかかわらず一意に決まる順序付けとして,筆者らは文字→行→段組→ページといったレイアウト構造上での順序(何行目の何文字目か)を用いることにした.このようなレイアウトの論理構造はページ記述には含まれていないが,文書画像の解析技術(例えば[7])を使えば文字オブジェクトの物理的な座標情報から再構築することができる.流通過程で個々の文字オブジェクトの出現順が変化してもレイアウト構造は変化しないためそれに基づくプリミティブの順序付けも影響をうけない.

3.3 埋め込みアルゴリズム

図1に,提案手法に基づく透かし埋め込み/検出の流れをしめす.埋め込みにあたっては,まずPDFファイル中の文字オブジェクト集合を解析して対象ページのレイアウト構造を構築する.構築されたレイアウトは人間が理解するように正しいものである必要はなく,埋め込み時と検出時で同じ結果が得られればよい.続いてあるキー(シード)に基づいて発生させた乱数列によって透かしの各ビットを埋め込む位置を決定する.この位置はレイアウト構造上で,x段目のy行目z文字目といった値で表現される.そこにある文字オブジェクトを動かして正規化間隔を変化させることにより埋

め込みを行なう.透かしの検出にあたっては,まず埋め込みと同様にレイアウト構造を構築する.続いて埋め込みに用いたのと同じシードから発生させた乱数–埋め込みプリミティブとして操作された文字オブジェクトの場所をしめすことになる–から得られた文字と次の文字の間の間隔を調べることによりビットごとの埋め込み判定を行なう.

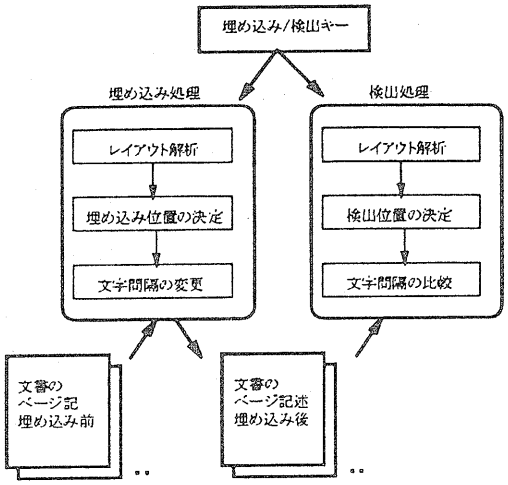

図 1: PDFへの透かし埋め込み/検出処理フロー.

3.1で述べたような正規化を行っても,行の右端をそろえる処理,あるいは禁則処理による文字の追い込み/追い出しが原因で埋め込みプリミティブの値が変動することがある.実際のワープロやフォーマッタの出力はこのような変動を含んでいることが多い.オリジナルとの比較なしに間隔の変化が透かしをあらわしていることを判定するためにはなんらかの推定を行わなくてはならない.ここでは,1ビットを埋め込むために複数のプリミティブを用いて統計的な推定によってビットを検出する方法[8]を採用している.

統計的手法では次式で計算される値によって1ビットの判定を行なう.

$$d = (1/N)\sum_{n}(a_n - b_n) \qquad (1)$$

a_n,b_nは乱数によって指定された文字列オブジェクトにおける正規化間隔の値である.1ビットを埋め込むためにa_n,b_nの組をN個用い,1を埋め込む場合にはa_nを大きくしb_nを小さくする,0を埋め込む場合にはa_nを小さくしてb_nを大きくする,といった操作を行っておく.通常,式(1)のdは0付近の値をとるが,埋め込みが行われた組に関しては正または負の大きな絶対値の値をしめす.この値が0から十分離れていれば閾値処理によってビットの値を検出することができる.単独の組で大きな絶対値を得られなくてもNを大きくとることにより,dの標準偏差を減少させて閾値処理の確度を高めることができる.統計的手法の利点は,検出の誤り確率と埋め込める容量の間のトレードオフをアプリケーションの要請に応じて設計できることである.

4 透かし埋め込み例

基本的な要件に関して本手法を評価するため,PDFのオペレータやフォントの種類を限定して埋め込み/検出プログラム(Java言語で約3,300行)をインプリメントした.

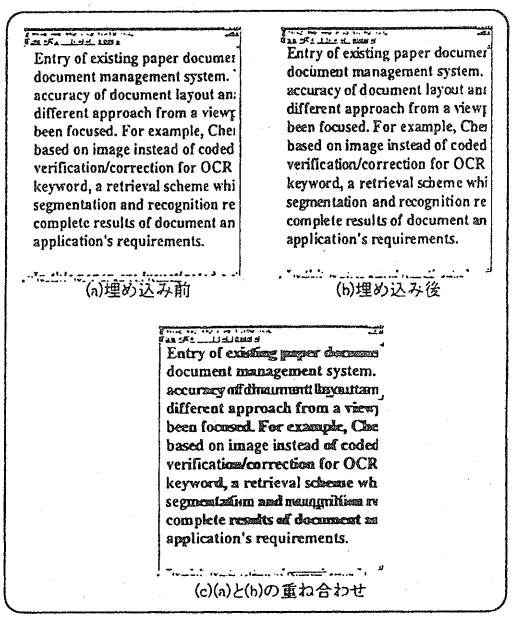

図 2: PDFへの透かし埋め込み例.

埋め込みのために文字間隔1ヶ所に与えた変更は,正規化前の値で対象文字のポイント数×0.05である.このプログラムを用いて,ページの記述に"Tokyo Research Laboratory"という文字列を透かしとして埋め込んだ例を図2にしめす.この例では,12ポイントのサイズで印刷される文字を0.6ポイントずつ左右に動かすことにより透かしを埋め込んでいる.埋め込み前と後の2つの表示画面を比べてもはっきりした違いはわからないが,重ね合わせてみると文字の位置が微妙に変化していることがわかる.多種の文書に対する定量的評価を行なう必要はあるものの,埋め込み処理が「ひとめで

わかるような」不自然さをもたらさないことは確認できた.

表 1: 文書データにおける正規化間隔の変動

正規化間隔	ワープロ	間隔拡張	間隔縮小
平均	0.996	1.054	0.946
分散	0.0000835	0.0000415	0.0000361

表 2: 正規化間隔標準偏差と推定誤り率の関係

	標準偏差 無/有	比	誤り率
N=1	0.0129/0.0156	3.78	0.0156 %
N=2	0.00913/0.0110	5.35	0.00000898 %
N=3	0.00746/0.00903	6.55	0.000000006 %

禁則処理等が原因で正規化間隔が変動していても透かしの埋め込み/検出ができることを確認するため, 実際にワープロで正規化間隔が変動しているPDFを作成し変動の幅を透かしによる変化と比較した. ワープロ出力から生成されたPDFは今回インプリメントしたプログラムでは処理できないオペレータを含んでいたため, 別にプログラムを用意して正規化間隔の値のみを計算している. 表1は, 正規化間隔の平均と分散を (a) 何も埋め込まれていない (しかし正規化間隔が変動している) ワープロ出力, (b) 埋め込みのために幅を広げたデータ, (c) 埋め込みのために幅を狭くしたデータ, に関してもとめたものである. (b),(c) の埋め込みは間隔の変動のないデータに対して行っている. 表1の値から式(1)のdの値の平均は埋め込み無しのデータに対して0, 有りのデータに対して± 0.108であることがわかる. 表2は表1の分散の値から分散の性質

$$\sigma^2(a+b) = \sigma^2(a) + \sigma^2(b) \quad (2)$$

$$\sigma^2(ka) = k^2\sigma^2(a) \quad (3)$$

より式(1)のdの分散(標準偏差)を計算したものである. 実際の埋め込みは, (a) のようなばらつきを含むデータに対して行われるため埋め込み有りの分散の値には (a) の分散の値も加算してある. N=1,2,3 の場合に対して埋め込み有りと無しの平均値の差0.108(1.054-0.946) が2つの標準偏差の和の何倍になるかという比をしめす. さらにその比を使って埋め込み有り/無しを判定する閾値を決めたときのFalse positive error(埋め込みが無いのに有りと判定してしまう確率) を正規分布を仮定して計算した. この表から, 埋め込み対象の正規化間隔の変動が今回調査したワープロ出力程度であれば, N=2とすれば誤り確率10^{-7}以下で埋め込み/検出が可能であると期待できる.

5 考察

2節で述べた要件と本手法の適合性について考察する.

1ページに埋め込める透かしの容量は, ページ内の文字数に依存する. 統計的手法を用いてN=2とした場合,1ビットの埋め込みには4個所の文字間隔を変更する必要がある. A4判1ページの文字数は図や写真が無い場合で1600から2000文字程度であり, 単純計算で400〜500ビットの透かしを埋め込むことができる. アノテーションを入れるといった目的には足りないが, 購入者識別用のID, 連絡先URL, デジタル署名等を埋め込むことは可能な容量である. 図や写真が大きな面積を占めるページへの埋め込みに関しては, 他のオブジェクトへの埋め込み手法との併用が必要である.

文字オブジェクトの物理的位置情報はページ記述のなかではテキストのコード列に次いで本質的な情報であり通常の流通過程では変更されない. この種の情報は, ページ記述一般に共通しておりフォーマット変換によっても失われる可能性は低い. 実際, 透かしを埋め込んだPDFをPostScriptに変換した後もう一度PDFに変換しても透かしを検出することができる.

インプリメントしたプログラムは, 埋め込み/検出の実現可能性を検証するためのもので意図的な攻撃に対処するための処理は行っていない. 強化を行ってどこまで頑健にできるか, いくつかの攻撃方法について検討する.

ランダム攻撃とは文字間隔によって透かしが埋め込まれていることを知る者がランダムに (文書の外観を変化させない範囲で) 文字オブジェクトを動かしてしまう攻撃をいう. 採用した統計的手法は, この種の攻撃に耐性をもつ[8]. 攻撃による幅の増減が式(1)に加わったとき, 打ち消しあってしまうからである. ただし, そのためにはある程度大きなNを使う必要があるため埋め込める容量は低下する.

結託攻撃とは, 同じ文書を複数の人間が購入し比較することにより, 埋め込まれたFinger Print情報(誰に販売したのかをしめす情報) の場所を同定し破壊する攻撃である. 本手法の場合には, 埋め込んだオブジェクト以外の文字オブジェクトもランダムに動かして透かしを埋め込んだ場所をわからなくする, 文字オブジェクトのファイル中での出現順序を変えて (外観にはなん

の影響もでない) 単純な比較ができないようにする, 等の対処が考えられる. 他に埋め込みの手法ではなく, ID等を符号化するときに冗長な表現を用いて符号化データの一部が破壊されても結託ユーザーを特定できるようにする対処策[9]も検討されている.

ページ記述フォーマットの仕様が公開されている場合, 原理的にはテキスト等の内容を抜き出して(PDFには暗号化の機能があるが透かしによって管理したいのは復号された後の文書の流通である), 新たな文書として再構成することが可能である. その意味で消去不可能な透かしは存在しない. しかしながら, 透かしの消去を試みる者の目的が複製した文書を販売して不当に(安易に)利益を得ることであるとすれば, 上述の出現順の変更などで透かしの消去に要するコストを彼(彼女)らが得るであろう利益に対して相対的に大きくすることにより一定の抑止効果が得られると期待できる.

6 まとめ

文字の間隔を操作することによりページ記述に電子透かしを埋め込む手法を提案した. 本手法では, 物理的な文字オブジェクトの情報をレイアウト構造に構成するととにより埋め込みプリミティブである文字オブジェクトを順序付けし, オリジナル文書無しでの検出や統計的手法の導入を可能にしている.

限定されたインプリメントではあるが, PDF文書への埋め込み/検出が可能であることを確認した. ワープロやフォーマッタが個別の文字の位置を操作している文書へも埋め込みが可能であるとの知見を得ることができた. 多様な文書への適用可能性を検証することが今後の課題である.

意図的な攻撃に対して本手法をどこまで頑健にできるかは, より詳細に検討してく必要がある. また, 埋め込み/検出のパフォーマンスについても今後, 評価・改良を行っていきたい.

参考文献

[1] 沼尾雅之 清水周一 森本典繁:データハイディングによるデジタル署名技術, 情処第53回全国大会 1M-13, (1996).

[2] Bienz,T. Cohn,R. and Meehan,James R.:Portable Document Format Reference Manual Version 1.2, Adobe Systems Inc. (1996).

[3] 渋谷竜二郎 楫勇一 嵩忠雄:PostScript およびPDF文書に対するデジタル透かしの提案, SCIS'98 9.2.E, (1998).

[4] 同定コード埋込み装置, 特開平6-324625.

[5] ドキュメントコピー防止方法, 特開平7-222000.

[6] 大渕竜太郎 増田宏 青野雅樹:3次元データへの情報の埋め込み, 情報処理学会 グラフィクスとCAD研究会/画像電子学会 Visual Computing '97 合同シンポジウム, (1997)

[7] 平山唯樹:複雑なカラム構造をもつ文書イメージの領域分割法, 信学論 (D-II), Vol.J79-D-II,No.11, pp.1790–1799, (1996)

[8] 小出昭夫:Data Hiding 技術とその応用, 精密工学会 画像応用技術専門委員会研究報告 Vol.12 No.4, pp.26–33, (1998).

[9] 吉田淳 岩村恵市 今井秀樹:画質劣化が少なく結託攻撃に強い電子透かし法, SCIS'98 10.2.A, (1998).