

ベクター表現の階層符号化に関する検討

河村 圭[†] 山本 勇樹^{††} 渡辺 裕[†]

[†] 早稲田大学大学院 国際情報通信研究科

〒168-0051 東京都新宿区西早稲田 1-3-10 29-7 号館

^{††} 早稲田大学理工学部 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: †{kei,yuuki}@tom.comm.waseda.ac.jp, ††hiroshi.watanabe@waseda.jp

あらまし ディスプレイの解像度や表示端末の処理能力に適した品質や符号量にするため、ベクター変換には符号量制御が必須である。本稿は、より柔軟な符号量制御実現するために、ベクター表現の階層符号化について検討する。まず、オブジェクトの位置や包含関係だけでなく、直線、曲線の統合や座標精度の制御がベクター表現の階層化を実現することを述べる。次に、階層化が符号量制御に有効であることを明らかにする。そして、基本階層と複数の拡張階層に分離、合成する階層符号化を提案する。提案手法により、ベクター表現の柔軟な符号量制御が可能となり、品質スケーラビリティを持たせることが可能となる。

キーワード ベクター表現、階層符号化、画像符号化

A Study on Hierarchical Coding of Vector Representation

Kei KAWAMURA[†], Yuki YAMAMOTO^{††}, and Hiroshi WATANABE[†]

[†] Graduate School of GITS, Waseda Univ., 29-7 Bldg., 1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051
Japan

^{††} Dept. of Elec., Info. and Comm. Eng., Waseda Univ., 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
E-mail: †{kei,yuuki}@tom.comm.waseda.ac.jp, ††hiroshi.watanabe@waseda.jp

Abstract A rate control function is necessary for vector conversion since image quality or vector codes should be adjusted to the resolution of a display or processing power of a terminal. In this paper, we study a hierarchical coding of vector representation to achieve a flexible rate control function. First, we explain that not only positions or inclusive relations of objects but also an integration of straight lines and curves or a coordinate granularity fulfills a hierarchy representation. Next, we show that a hierarchical coding is an effective for rate control function. Then, we propose a hierarchical coding with a base layer and some enhanced layers. The proposed method is able to achieve both a flexible rate control and a quality scalability function.

Key words Vector Representation, Hierarchical Coding, Image Coding

1. はじめに

ベクター表現は端末の解像度に依らず高品質な画像を表示できるため、ベクター表現によるコンテンツの需要が高まると予想される。コンテンツを効率よく蓄積、配信、閲覧するためにはデータ圧縮が必須である。そして、ディスプレイの解像度や表示端末の処理能力に適した品質や符号量にするため、ベクター変換には符号量制御及びスケーラビリティを実現する階層符号化が求められる。

我々は以前から、許容誤差に応じた直線・曲線統合手

法について検討し、ベクター表現において符号量制御を実現する手法を提案している [1], [2]。

本稿では、より柔軟な符号量制御実現するために、ベクター表現の階層符号化について検討する。まず、座標精度の制御や直線・曲線の統合がベクター表現の階層化を実現することを述べる。次に、ベクター表現のスケーラビリティについて述べ、具体的な実現手法を述べる。そして、基本階層と複数の拡張階層に分離、合成する階層符号化を提案する。提案手法により、ベクター表現の柔軟な符号量制御が可能となり、品質スケーラビリティ

を持たせることが可能となる。

シミュレーション実験により、階層符号化の実現性とレートディストーションによる性能を明らかにした。直線・曲線の統合による階層化手法は拡張階層の符号量が大きく、さらなる改善が必要である。また、座標精度による階層化手法は、階層性を持たせない場合の性能と同程度の性能が得られた。これらを組み合わせることにより、柔軟な符号量制御と階層符号化が実現できることが確認された。

2. 従来の符号量制御

2.1 ガウスフィルタ

一般的に、ベクター表現からは高品質なビットマップが得られ、しかもベクター表現のファイルサイズは小さい。しかし、スキャナにより高解像度ビットマップを取得し、これをベクター変換してもファイルサイズの小さなベクター表現は得られない。一方で、低解像度ビットマップからはファイルサイズの小さい、十分な主観品質のベクター表現が得られる。

これは以下のような原因による。まず、ベクター変換における近似の許容誤差は画素サイズ依存である。そのため、解像度が高ければ相対的に許容誤差が小さくなる。次に、解像度を上げるとS/N比の悪化が考えられる。2値画像の場合、輪郭線上のノイズが顕著になり、後段の処理である曲線統合がされにくくなるという影響がある。そのため、高解像度ビットマップからのベクター変換はファイルサイズが大きくなる。逆に低解像度ビットマップでは許容誤差が大きく、デジタル誤差によりノイズが目立たないため、ベクター表現のファイルサイズが小さくなる。

実際には低解像度でスキャンする代わりに、高解像度でスキャンしたビットマップに対してガウスフィルタをかけることで等価的に解像度変換（低解像度化）を行う。また、ガウスフィルタの半径を変えることでリニアな解像度変換が可能となり、スムーズな符号量制御が可能となる。

本手法によるレートディストーションを図1 Gaussian-filter に示す。ただし、入力解像度は1200dpiである。本手法はフィルタ半径を大きくするほど座標精度が落ちるために、原画像と比較して位置ずれが発生する。また、図2、図3に示すように輪郭線が隣接している領域では再現性が著しく低下する。

比較のために、ラスタ表現の2値画像符号化であるJBIGによるレートディストーションを図1 JBIG に示す。ただし、JBIGは可逆圧縮である。そこで、ガウスフィルタによる等価的な解像度変換をした画像に対して

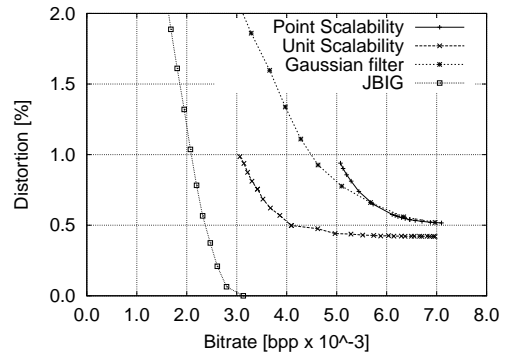


図1 各種法におけるレートディストーション

Fig. 1 Performance of conventional rate control methods



図2 原画像

Fig. 2 Original image



図3 フィルタ適用後画像

Fig. 3 Performed image by the filter

も符号化を行った。つまり、ディストーションはガウスフィルタのみに起因する。

2.2 座標精度

ベクター表現は、画像を線、面、グラデーションなどの幾何的、数学的な要素を組み合わせることで表現される。そして、線や面の輪郭線は通過点と制御点を用いて表現される。通過点同士の間は数式により滑らかに補完される。また、通過点と制御点は適当な座標系で表現される。そこで、座標精度を落とすことで符号量を削減できる。

従来手法[3]では、ベクター変換において直線により輪郭線の最適近似を行う。そして、直線同士の交点の座標が小数点精度で求められる。次に、ファイルに出力する際にはスケールを行い、座標を整数化して出力する。デフォルトでは10倍の座標精度を用いる。

しかし、入力解像度に対して出力解像度が低い場合には、10倍の座標精度を確保しても有効に利用することはできない。実際、本稿の入力解像度として利用している1200dpiのビットマップを表示できるディスプレイは現状では存在せず、150dpiや200dpi程度である。そこで、座標精度を等倍や1/2倍にすることで符号量の削減が可能となる。

本手法によるレートディストーションを図1 Unit Scal-

ability に示す。ただし、入力解像度は 1200dpi である。座標のスケージングによる符号量制御は、通過点同士の距離が十分離れている場合に有効である。また、ガウスフィルタや次に述べる通過点の削減とは独立に制御可能であるため、いずれの方法とも組み合わせることができる。

2.3 直線・曲線の統合

ガウスフィルタによる符号量制御は、必ずラスタ表現を経由しなければならないという制約がある。すでにベクター表現に変換された画像に対して、さらに符号量を削減するにはラスタライズ、ガウスフィルタ適用、ベクター変換、という 3 ステップが必要である。また、目標符号量にするためには、ガウスフィルタの半径を適切に選択する必要がある。

これらの問題点を解決した符号量制御として、直線・曲線の統合による符号量制御がある [2]。以下に、簡単な原理を述べる。ベクター変換では、最初に輪郭線を直線で近似し、最適多角形を得る。そして、多角形の頂点、または辺を適応的に滑らかな曲線で置き換える。ここで、大きな曲線は複数の直線に分割されて表現される。次に、凸方向が同一の連続する曲線のみを統合して、符号量の削減をはかる。高解像度ビットマップでは S/N が悪化しているため、ノイズにより曲線の凹凸が頻繁に変化してしまう。大局的に見て同一の曲線であっても統合できない。これらの条件を緩和することで、直線や凸方向の異なる曲線を統合できる。統合による通過点の削減により、符号量削減が可能となる。

本手法は、ベクター表現のまま通過点を削減し、符号量を削減する。そのため、ラスタライズする必要がない。また、許容誤差に応じて統合される線がシーケンシャルに決定されるため、符号量制御が容易である。

本手法によるレートディストーションを図 1 Point Scalability に示す。ただし、入力解像度は 1200dpi である。ビットレートが高い間はガウスフィルタによるレートディストーションと同程度の性能である。しかし、大幅な符号量の削減は、主観品質が悪化するためにできない。ガウスフィルタは輪郭線上のノイズを削減するだけでなく、輪郭線そのものを削除できる。しかし、本手法は輪郭線の削除を伴う符号量制御はできない。

2.4 従来手法のまとめ

ラスタ表現を経由するガウスフィルタによる符号量制御では、各品質における通過点の対応関係を調べてから差分を抽出しなければならない。そのため、ベクター表現の階層化には向かない。

一方、ベクター表現のみを利用している座標精度や直線・曲線の統合による符号量制御からは、容易に階層化できる。そこで、本稿では以上の 2 つの手法について、

階層符号化を実現する手法について述べる。

3. ベクター表現のスケラビリティ

3.1 スケラビリティ

スケラビリティとは、符号化データの一部を復号しても、空間的に一部の画像や品質の低い画像を復元できるように符号化データを構成しておく方法である。階層化した符号化データを用意しておけば、一部の符号化データしか復号できない処理能力が低い端末や低速の伝送路に接続された端末と、全部の符号化データを復号できる高機能端末や高速の伝送路に接続された端末との双方で相応な画像を復元することが出来る [4]。

静止画像には様々なレベルのスケラビリティが存在する。ベクター表現ではオブジェクトやその包含関係、位置関係などセマンティックな階層化や、符号量制御による品質の階層化が考えられる。

例えば、マンガは 1 ページに複数のコマが存在する。1 ページ 1 ファイルとして符号化されている場合にも、空間的な位置によるスケラビリティがあれば、1 コマ単位で閲覧が可能となる。また、全体を俯瞰したい場合には、様々な品質の画像を得られるように符号化されていれば、ディスプレイに表示可能な解像度（品質）までを復号することで必要以上の復号処理が低減される。

ベクター表現では、空間的なスケラビリティの確保は容易である。以下では、本稿で扱う品質スケラビリティとして、座標精度による階層化、直線・曲線の統合による階層化について述べる。

3.2 座標精度によるスケラビリティ

ディスプレイの表示可能な解像度とベクター表現のもつ解像度が異なる場合、解像度変換が必要となる。一般に、ベクター表現ではディスプレイの解像度に合わせたラスタライズを行うので、解像度変換は必要ないと考えられている。しかし、ディスプレイで利用可能なベクター表現の解像度には上限が存在すると考えられる。

例えば、ディスプレイの 1 画素に対して 0.1 画素の精度（10 倍の精度）で座標を指定しても、アンチエイリアシングを用いてその差を利用することは現実的でないと想像できる。つまり、必要な解像感に合わせた解像度を持つベクター表現を提供することが求められる。

一方で、一部分を拡大することにより、ディスプレイの解像度は相対的に高くなる。そこで空間的に必要な領域のみ、解像度の高いベクター表現を提供することで符号量やラスタライズの処理コストの削減が可能となる。

以上のように、ベクター表現では座標精度による品質スケラビリティを持たせることが重要である。

表 1 ベクター表現のファイル構造

Table 1 File structure of vector representation

X	Y	Info.	note
0	70131		Start (absolute) coordinate
0	-147	i	Relation coordinate
24	0	60 c	
0	-15	7 c	
-32	-32	12 7 C	
33	-33	20 15 C	
-25	0	12 c	
0	-25	7 c	
32	18	13 c	
0	-13	9 c	
39	-18	11 c	
⋮	⋮	⋮	

3.3 直線・曲線の統合によるスケーラビリティ

座標精度と同様に，ディスプレイで利用可能なベクター表現における通過点の間隔には下限が存在すると考えられる．そして，通過点の間隔は，直線・曲線の統合による通過点削減により調節可能となる．これは，許容誤差をパラメータとして制御される．

ただし，曲線同士が作る間隔については，本手法では削減できない[2]．そのため，曲線同士の位置関係による曲線そのものの削減手法が必要である．

4. 階層符号化

4.1 ファイル構造

ベクター表現における1本の閉パス（閉じた曲線）は，表1のように絶対座標における開始点と，相対座標による通過点及び曲線の種類や制御点情報の繰り返しにより構成される．

座標は入力解像度に対してスケーリングを施し，整数化されている．相対座標を用いるのは冗長性を削減するためである．

4.2 座標精度による階層構造

座標は通過点の削減を伴わない．そこで，各座標値を適当な数値（量子化ステップ）で割り，商を四捨五入する．さらに，真値との差分（量子化誤差）を保存することで容易に階層化可能である．これは式2のように表される．

$$\text{coord}_{\text{base}} = \text{Int}(\text{coord}/Q_{\text{step}} + 0.5) \quad (1)$$

$$\text{coord}_{\text{ext}} = \text{coord} - \text{coord}_{\text{base}} \times Q_{\text{step}} \quad (2)$$

商は基本階層へ，差分は拡張階層へ振り分ける．基本階層に対して量子化を繰り返すことで，複数の拡張階層を生成できる．

ファイル内では，冗長性を削減するために相対座標が利用されている．相対座標に対して量子化を行うと量子

表 2 座標の階層化の仕方

Table 2 Sample of making a hierarchical coordinate data

Absolute coordinate			Quantized coordinate		
x	y	Info.	x'	y'	Info.'
0	70131		0	35066	
0	69984	i	0	34992	i
24	69984	60 c	12	34992	30 c
24	69969	7 c	12	34985	4 c
-8	69937	12 7 C	-4	34969	6 4 C
25	69904	20 15 C	13	34952	10 8 C
0	69904	12 c	0	34952	6 c
0	69879	7 c	0	34940	4 c
32	69897	13 c	16	34949	7 c
32	69884	9 c	16	34942	5 c
71	69866	11 c	36	34933	6 c
⋮	⋮	⋮	⋮	⋮	⋮

Base layer			Enhanced layer		
x	y	Info.	x	y	Info.
0	35066		0	-1	
0	-74	i	0	0	
12	0	30 c	0	0	0
0	-7	4 c	0	-1	-1
-16	-16	6 4 C	0	-1	0 -1
17	-17	10 8 C	-1	0	0 -1
-13	0	6 c	0	0	0
0	-12	4 c	0	-1	-1
16	9	7 c	0	-1	-1
0	-7	5 c	0	0	-1
20	-9	6 c	-1	0	-1
⋮	⋮	⋮	⋮	⋮	⋮

化誤差が蓄積し，最終的に破綻する．そこで，絶対座標に対して量子化を行い，量子化後の相対座標を基本階層に保存する．

表2に量子化ステップ2における階層化の具体的手順を示す．絶対座標（Absolute coordinate）に対して量子化を行う（Quantized coordinate）．次に，開始点からの相対座標を計算し，基本階層とする（Base layer）．また，量子化誤差を計算し，拡張階層とする（Enhanced layer）．

表3に4階層にした場合の例を示す．表2の基本階層に対して量子化ステップ2で階層化を繰り返すことにより得られる．

4.3 直線・曲線の統合による階層構造

直線・直線の統合処理により，輪郭線の最適多角形は図4，図5のようになる．この図から直線・曲線の統合による階層化について説明する．まず，統合に隣接する点の座標は変わらない．次に，統合によって内部の点が削減される．そして，新しい点が追加される．制御点情

表 3 座標の 4 階層化例

Table 3 Sample of 4 hierarchical coordinate data

Base layer			Enhanced 1			Enhanced 2			Enhanced 3		
x	y	Info.	x	y	Info.	x	y	Info.	x	y	Info.
0	8767		0	-1		0	0		0	-1	
0	-19	i	0	0		0	0		0	0	
3	0	8 c	0	0	-1	0	0	0	0	0	0
0	-1	1 c	0	-1	0	0	-1	0	0	-1	-1
-4	-4	2 1 C	0	-1	-1 0	0	-1	0 0	0	-1	0 -1
5	-5	3 2 C	-1	0	-1 0	-1	0	0 0	-1	0	0 -1
-4	0	2 c	0	0	-1	0	0	0	0	0	0
0	-3	1 c	0	0	0	0	0	0	0	-1	-1
4	3	2 c	0	-1	0	0	-1	-1	0	-1	-1
0	-2	2 c	0	-1	-1	0	0	-1	0	0	-1
5	-2	2 c	0	-1	-1	0	-1	0	-1	0	-1

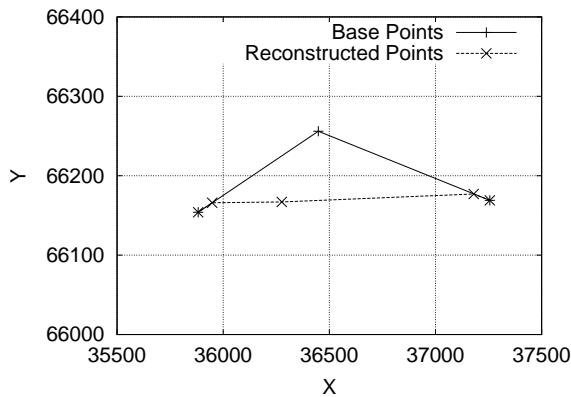


図 4 統合処理例 1

Fig. 4 Integration sample 1

報は，追加される点と統合に隣接する点の一方が影響を受ける（表 4，表 5）．

統合前の状態に着目して，統合前後の点数における定式化を行う．まず，統合する点の直前に c_1 個の点が存在する．次に， m 個の点が削減され， n 個の点が追加される．そして， c_2 個の統合に関係ない点が存在する．ここで， c_1, c_2 は任意定数である．また， $m > n$ は自明である．

次に，統合による階層化を定式化する．基本階層には，まず c_1 個の点が存在する．次に n 個の点，そして c_2 個の点が存在する．拡張階層には，統合と関係ない点数が c_1 個，置き換えられる点数が n 個，追加する点数が m 個であることが記録される．そして実際に置き換える m 個の点情報が記録される．

表 4 では，統合の直前に 1 個の点が存在する．次に 1 個の点が削減され，3 個の点が追加される．同様に，表 5 では，統合の直前に 1 個の点が存在し，次の 4 個の点が削減され，9 個の点が追加される．

表 4 統合前後の座標データと階層化の仕方 1

Table 4 Coordinate of both before and after integration and making hierarchical coords 1

Base coordinate			Reconstructed coordinate		
x	y	Info.	x	y	Info.
⋮	⋮	⋮	⋮	⋮	⋮
37256	66169	10 c	37256	66169	10 c
36448	66256	30 17 C	37180	66177	30 17 C
35882	66154	533 61 C	36276	66167	528 451 C
			35948	66166	207 c
			35882	66154	90 c
⋮	⋮	⋮	⋮	⋮	⋮

Base layer			Enhanced layer		
x	y	Info.	x	y	Info.
⋮	⋮	⋮			
-9	-14	10 c			1 1 3 E
-808	87	30 17 C	-76	8	30 17 C
-566	-102	533 61 C	-904	-10	528 451 C
			-328	-1	207 c
⋮	⋮	⋮			90 c

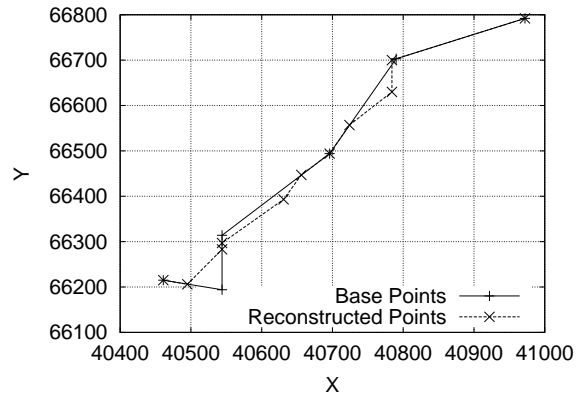


図 5 統合処理例 2

Fig. 5 Integration sample 2

5. シミュレーション実験

5.1 座標精度の階層化によるレートディストーション提案手法を実装し，階層符号化の実現性についてシミュレーション実験を行った．基本階層は 1/2 倍精度とし，拡張階層を加えるごとに等倍，2 倍，4 倍，8 倍精度となるようにした．また，階層化する前の最高精度として，8 倍，4 倍，2 倍のベクター表現ファイルを利用した．基本階層と拡張階層のファイルサイズを合計したときのレートディストーションを図 6 に示す．また，階層化しない場合のレートディストーションを合わせて示す．

実験結果より，座標精度による階層化ではオーバーヘッドがほとんど発生していないことが明らかとなった．た

表 5 統合前後の座標データと階層化の仕方 2

Table 5 Coordinate of both before and after integration and making hierarchical coords 1

x base	y base	Info.	x recon	y recon	Info. recon
⋮	⋮	⋮	⋮	⋮	⋮
40972	66792	89 41 C	40972	66792	89 41 C
40790	66703	34 19 C	40784	66700	34 19 C
40696	66494	178 80 C	40784	66630	62 36 C
40544	66314	23 17 C	40724	66557	20 c
40544	66194	23 -37 C	40696	66494	20 c
40461	66215	66 24 C	40656	66447	17 c
			40631	66393	15 c
			40544	66297	26 c
			40544	66283	26 c
			40495	66206	21 c
			40461	66215	31 c
⋮	⋮	⋮	⋮	⋮	⋮

x base	y base	Info.	x ext	y ext	Info. ext
⋮	⋮	⋮			
-110	-102	89 41 C			1 4 9 E
-182	-89	34 19 C	-188	-92	34 19 C
-94	-209	178 80 C	0	-70	62 36 C
-152	-180	23 17 C	-60	-73	20 c
0	-120	23 -37 C	-28	-63	20 c
-83	21	66 24 C	-40	-47	17 c
			-25	-54	15 c
			-87	-96	26 c
			0	-14	26 c
			-49	-77	21 c
⋮	⋮	⋮			31 c

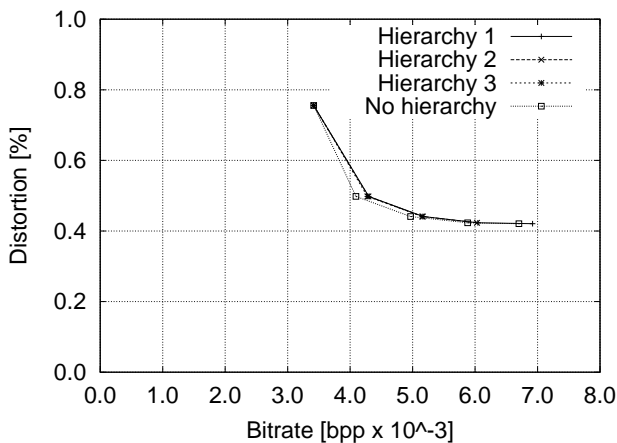


図 6 座標精度による階層化のレートディストーション
Fig. 6 Performance of hierarchy using unit scale.

だし、座標精度が 1/2 倍から等倍にする拡張階層は他の拡張階層に比べて符号量が多い。これは、座標精度が 2 倍以上の場合には実質的な情報が発生していなかったためであると考えられる。

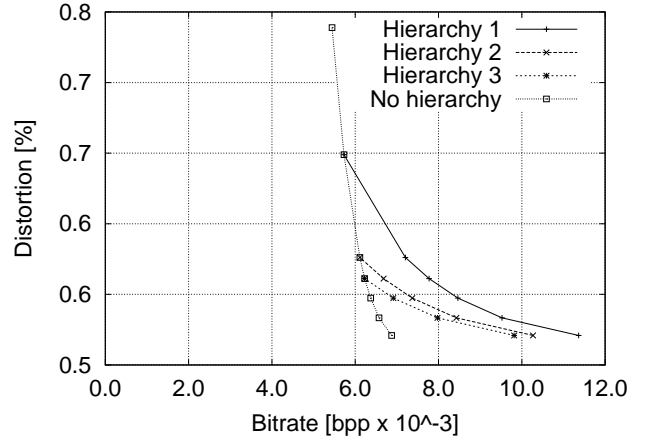


図 7 直線・曲線の統合による階層化のレートディストーション
Fig. 7 Performance of hierarchy using integration.

5.2 直線・曲線統合の階層化によるレートディストーション

最大許容誤差を 2.0 とし、拡張階層を加えるごとに 1.0, 0.8, 0.6, 0.4, 0.2 となるよう階層化を行った。そして、基本階層の許容誤差を 2.0 から 8.0 まで変化させて実験を行った。基本階層と拡張階層のファイルサイズを合計したときのレートディストーションを図 7 に示す。また、階層化しない場合のレートディストーションを合わせて示す。

実験結果より、直線・曲線統合による階層化では、オーバーヘッドが大きいことが明らかとなった。これは、基本階層から削減されて利用できない点が多数存在するためであると考えられる。直線・曲線の統合手法に起因するものであり、避けられない。

6. むすび

ベクター表現において、符号量を制御する手法を利用して、階層符号化を実現する手法を提案した。座標精度を利用した階層化では、階層化しない場合と同程度の性能を得られることが明らかとなった。また、直線・曲線の統合手法を利用した階層化では、拡張階層の符号量が大きいことが明らかとなった。

提案手法はベクター変換時だけでなく、すでにベクター表現に変換された画像に対しても適用可能である。

文 献

- [1] 河村, 渡辺, “ベクター変換における曲線最適化アルゴリズムの一検討”, FIT2004 第 3 回情報科学技術フォーラム, 3S-2, Sep. 2004.
- [2] 河村, 渡辺, “ベクター表現における符号量制御に関する検討”, 画像符号化シンポジウム資料 19th, pp. 67-68, P-5.04, 2004.
- [3] “potrace,” <http://potrace.sourceforge.net/>
- [4] 大久保, 川島, MCR, “H.323/MPEG-4 教科書”, IE インスティテュート, Apr. 2001.