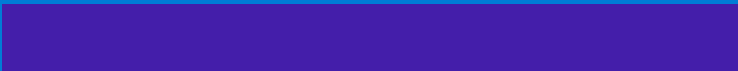


-
-
-
-
-
-
-
-
-
-
-

情報処理システム論 (21)



-
-
-
-
-
-
-
-

・
・
・

データ圧縮

- ・ ビット数を減らす
 - ディスクの節約
 - ・ ファイルのサイズ
 - ネットワークの帯域の節約
 - ・ 転送時間の節約
 - ・ お金の節約

近年のCPUの高性能化の恩恵

・
・
・
・
・
・
・
・

・
・
・

圧縮の2つの世界

- 損失のない圧縮
 - 完全に元に戻らなければならない情報
 - プログラム、テキスト、データベース
 - 1ビット変化しても大きな問題が出る
- 損失のある圧縮
 - 音声
 - 低音・高音を落とす
 - 画像
 - 細かな絵をぼやかす

・
・
・
・
・
・
・
・

•
•
•

身近な圧縮ファイル

- .ZIP
 - .LZH
 - .ARC
 - .ZOO
 - .gz
 - .GIF
 - .JPG
- 圧縮する
 - 複数のファイルをまとめる
 - アーカイブ

⋮

モデルと符号化

- モデル
 - 統計情報
 - データごとに異なる
 - 時間とともに変化する(適応型)
- 符号化

情報理論ふたたび

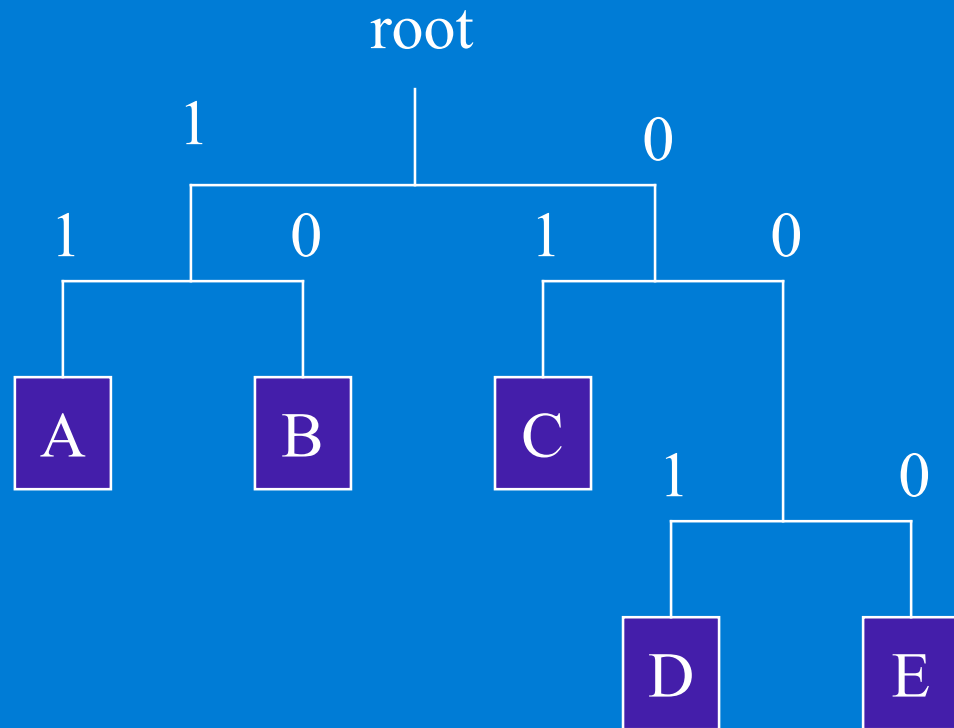
- 各符号の出現確率が異なると
 - それぞれの符号のもつ情報量が異なる
- 出現確率が与えられると、
 - 符号ごとのビット数を可変にする
 - 出現確率の少ない符号 - 長いビット数
 - 出現確率の多い符号 - 短いビット数
 - 一意に復号可能なコード

最小冗長符号: Shannon-Fano 符号

Shannon-Fano 符号(1)

- 符号と符号木の例

文字	符号	頻度
A	11	15
B	10	7
C	01	6
D	001	6
E	000	5



Shannon-Fano 符号(2)

- 符号の作り方
 - 記号ごとの出現頻度表をつくり、頻度でソート
 - 表を2つに分割
 - 上半分の頻度の合計と下半分の頻度の合計の差ができる限り小さくなるようにする
 - 表の上に1を、下に0を割り当てる
 - 分割された表ごとに、分割できなくなるまで手続きを繰り返す

Shannon-Fano 符号(3)

- 実際に作ってみよう

文字	頻度			
A	15	1	<u>1</u>	
B	7	<u>1</u>	0	
C	6	0	<u>1</u>	
D	6	0	0	<u>1</u>
E	5	0	0	0

Shannon-Fano 符号(4)

- どれくらい効果的か

文字	頻度	情報量	bits	符号長	bits
A	15	1.38	20.68	2	30
B	7	2.48	17.35	2	14
C	6	2.70	16.20	2	12
D	6	2.70	16.20	3	18
E	5	2.96	14.82	3	15
		85.25			89

⋮

Huffman 符号(1)

- Shannon-Fano より効率的
- 作り方
 - リストから頻度最小の記号2つを順にとりだす
 - 一方に0、もう一方に1を割り当てる
 - 下位ビットからの割り当てとなる
 - 頻度を合計したもの1つを新たに作り、戻す
 - リストの内容が1つになるまで繰り返す

⋮

Huffman 符号(2)

- 実際につくってみよう

文字	頻度	
A	15	
B	7	0
C	6	1
D	6	0
E	5	1

文字	頻度	
A	15	
BC	13	0
DE	11	1

文字	頻度	
A	15	0
BCDE	24	1

Huffman 符号(3)

- 結果

文字	頻度	
A	15	0
B	7	100
C	6	101
D	6	110
E	5	111

Huffman 符号(4)

- どれくらい効果的か

文字	頻度	情報量	bits	SF	bits	H	bits
A	15	1.38	20.68	2	30	1	15
B	7	2.48	17.35	2	14	3	21
C	6	2.70	16.20	2	12	3	18
D	6	2.70	16.20	3	18	3	18
E	5	2.96	14.82	3	15	3	15
		85.25			89		87

⋮

頻度情報をどうやって得るか

- 最初に全部を調べる
 - 固定的な頻度表を最初に作る
- 前から順番に辿りながら符号化
 - 頻度表を変更しながら符号化
 - 適応的 Huffman 符号
 - 圧縮側と復号側で最初に頻度表を共有せずにする
 - 圧縮後のデータ量の削減

•
•
•

画像の圧縮

- BMP形式
 - 圧縮しない形式
- GIF形式
 - 損失のない圧縮
 - パソコンで作成した図画に向く
- JPG形式
 - 損失のある圧縮
 - 写真(風景画像など)向き

• • • • • • • •

・
・
・

GIF向きの画像

- ・ パソコンで書いた図画
 - 同じ色の領域が続く
 - 色の境界が明確
 - 色数が少ない
- ・ 損失のない圧縮でも効果的
 - 損失のある圧縮だと
 - ・ 色の境界がぼやける
 - ・ 字が読みにくくなる

・
・
・

JPG向きの画像

- 風景画像
 - 色がなめらかに変化する
 - 明確な色の境界がない
 - 同じ色の領域が続かない
 - 色数が多い
- 画像向きの損失のある圧縮が効果的
 - 細かな部分の情報の保存が不要
 - 人間の認識力とのかねあい

・
・
・
・
・
・
・
・

⋮

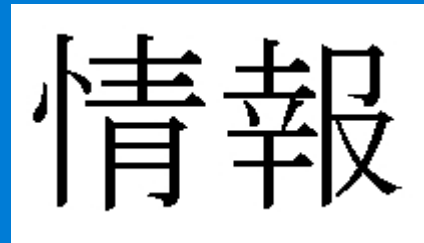
画像圧縮の実例(文字図形)

- 画質とファイルサイズ

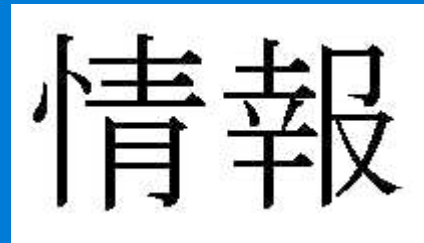
fig100.bmp	3422バイト
fig100.gif	1085
fig100.jpg	13794
fig50.jpg	4614
fig1.jpg	1704

quality

100



50



1



⋮

画像圧縮の実例(風景画像)

- 画質とファイルサイズ
– サイズはバイト

95.bmp 129170

95.gif 29823

95.jpg 25078

55.jpg 7151

15.jpg 3284

quality

95

55

15



•
•
•

動画による通信

- 動画はデータ量が多い
- 細い通信回線の考慮
 - 人間の認識能力をうまく利用
 - ある瞬間の画像の解像度はあまり高くなくてよい
 - 連続する画像の変化で解像度を補う
 - テレビの技術
 - 変化の少ない部分の情報を少なくする
 - 差分データ
 - 通信回線での損失の考慮

• • • • • • • •

・
・
・

損失のある回線での通信

- 変化のある部分の差分情報を送る
 - フレーム間圧縮
 - 変化が激しいとコマ数が減る
- 損失に備えて絶対情報を定期的にする
- 画像が途切れたらそのまま静止画表示
 - 再生側の動作
- 音声は極力損失させない

•
•
•

圧縮と遅延

- 動画を圧縮すると遅延が発生
 - フレーム間圧縮のための遅延
 - 圧縮計算のための遅延
 - 損失補償のための蓄積による遅延
 - 1フレームに対する情報を時間的に前後に冗長に分散させる

⋮

動画転送プロトコル

- MPEG1
 - 1.5Mbps
- MPEG2
 - 4-9Mbps
- Motion JPEG
 - フレーム間圧縮なし
- H.261
 - 64K-1.9Mbps

⋮

•
•
•

参考文献

- データ圧縮ハンドブック
 - M. ネルソン（荻原剛志・山口英 訳）
 - プレンティスホール・トツパン
 - ISBN 4-8101-8578-8