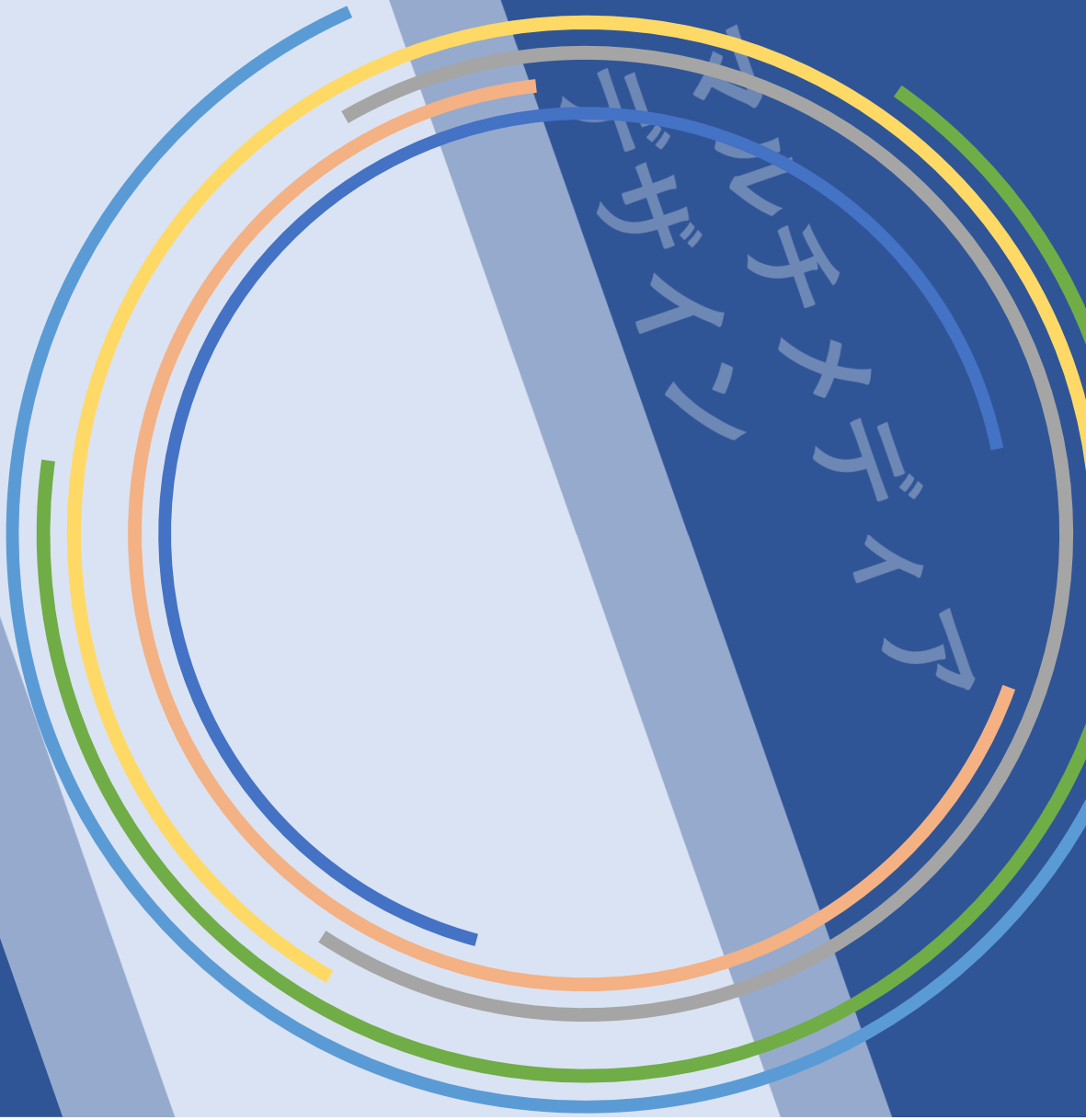
この授業のスライドはLMSで公開されます
実験の質問やレポートに関する質問などはSlackへ
https://keio-st-multimedia.slack.com

#10
画像認識
動画・音声圧縮

担当：　西　宏章



画像認識

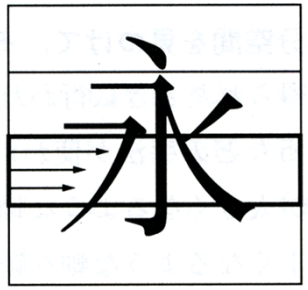
- さまざまな画像認識が提案されているが、ここでは文字認識を取り上げる
  - いずれにしても、Deep Learning全盛期を迎え、画像認識技術は飛躍的に向上した
  - Strategicな方法で、考え方が参考になる手法のみ扱う
  - Deep Learningは、別の授業で
- 文字認識の分類
  - 光学的文字認識方式(OCR)
    - 文字を静止画像として扱い、認識する。カメラやスキャナを利用
  - オンライン文字認識方式(OLCR)
    - 文字の筆記情報を認識する。タブレットやマウスを利用
- 文字認識研究の歴史
  - 1960年には、専用のフォントを用いた英文字認識が行われていた
  - 1970年の後半から手書漢字の認識アルゴリズム研究が盛ん
  - WindowsXP以降、手書き文字認識が標準で備わっている
  - DLの登場で、認識率は人間平均を上回る

文字認識の手順

- 前処理
  - これまで述べた画像処理を利用して、識別しやすいように（識別時にエラーが発生しにくいように）画像を加工する。特に回転や２値化、ノイズ除去、位置大きさの正規化などが行われる
- 特徴抽出
  - 前処理した画像から、識別処理で必要な量や数値を取り出す。一種の情報圧縮
- 識別処理
  - 具体的に、抽出した特徴から辞書を引き、文字を特定する
- 知識処理
  - 認識精度を上げるため、たとえば住所の変換であれば、実在する住所の辞書と照らし合わせるなどして認識率を向上させる

特徴抽出

- メッシュ特徴
  - 画像をメッシュにわけて、方形小領域中の画素数比を求め、特徴値とする(認識は無謀)
- ペリフェラル特徴
  - 横や縦に走査して「最初に黒の画素に当たった時の移動量」を計測する。
    この場合もメッシュ特徴のように小領域に分割して、
    その領域ごとの値を特徴値とする(単独での認識は困難)



- 周辺分布特徴
  - 文字領域部分の各画素を縦や横方向でカウントすることで、周辺分布を得る。
    この分布値を特徴値とする（単独での認識は困難）

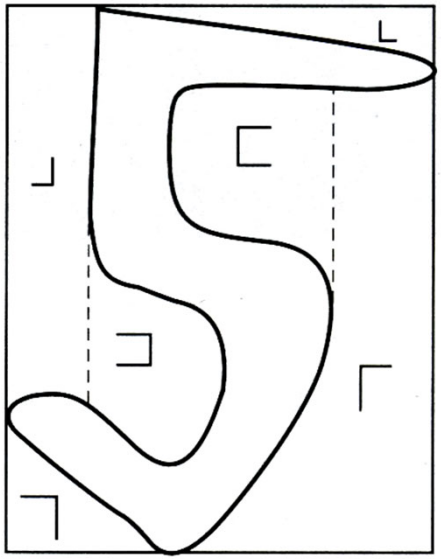
手書き文字認識

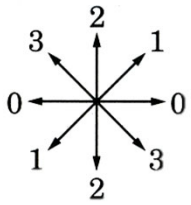
- 文字の構造における特徴を抽出する
  - 輪郭線の構造、芯線の構造、文字領域の構造、背景構造等を利用
  - 輪郭線構造に着目した特徴
    偏や旁に注目するために、まず切り分ける等
  - 芯線構造に着目した特徴

種類	記号	個数
孤立線分	$L(x, y, \theta)$	K_1
屈折点	$R(x, y)$	K_2
分岐点	$B(x, y)$	K_3
第一種交点	$C_1(x, y)$	K_4
第二種交点	$C_2(x, y)$	K_5

手書き文字認識における補助的情報

- 文字の構造における特徴を抽出
  - 背景構造に着目した特徴
  - どの方向に線画存在するかをコの字型の記号で表現
- 方向線寄与度特徴
- 黒い線の長さに基づいて、方向寄与度の大きさを決定、認識に用いる

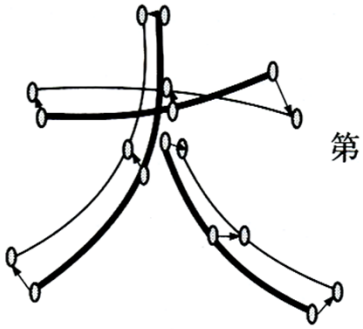




いずれにしても、認識の補助にはなるが単体での認識は困難

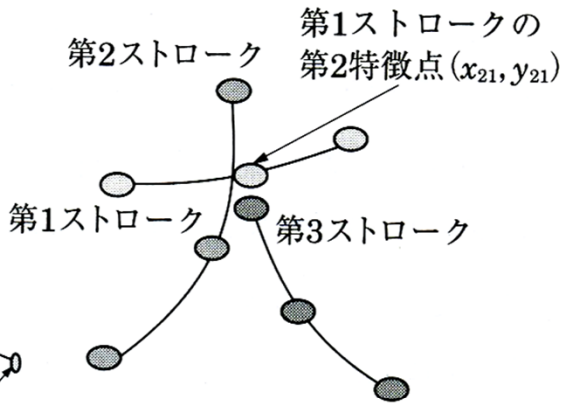
手書き文字認識（OLCRの例）

- 点近似特徴
  - 文字線上の特徴的な点を抽出し、これらの座標値を直接用いて特徴ベクトルを構成する方法（ひらがなはストロークを5等分割(6点)、漢字は2等分割(3点)がよいといわれている）
- 線近似特徴



(a) 変形の程度

＝対応点間の距離（矢印）の総和

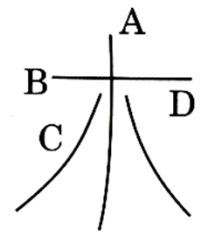


(b) 特徴ベクトル

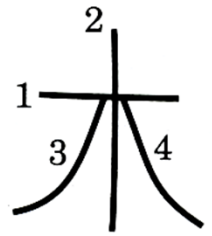
$$f=(x_{11}, x_{21}, x_{31}, x_{12}, x_{22}, x_{32}, x_{13}, x_{23}, x_{33}, y_{11}, y_{21}, y_{31}, y_{12}, y_{22}, y_{32}, y_{13}, y_{23}, y_{33})$$

x_{ij}：第jストロークの第i特徴点のx座標

y_{ij}：第jストロークの第i特徴点のx座標



入力パターン



標準パターン

	A	B	C	D
1	d_{1A}	d_{1B}	d_{1C}	d_{1D}
2	d_{2A}	d_{2B}	d_{2C}	d_{2D}
3	d_{3A}	d_{3B}	d_{3C}	d_{3D}
4	d_{4A}	d_{4B}	d_{4C}	d_{4D}

数字，英字は筆順を示す

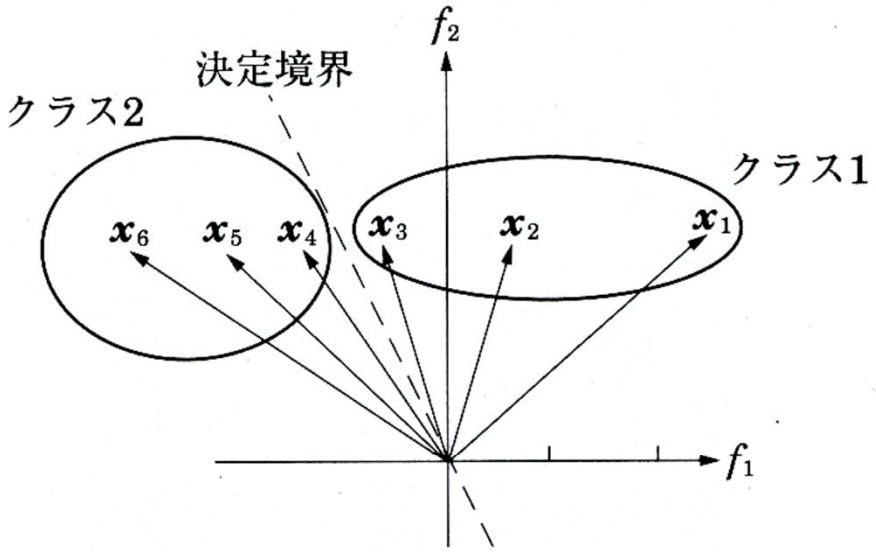
d_{1A}：ストローク1とAの間のストローク間距離

標準パターンの各ストロークから見て距離最小（太字）となる入力パターンのストロークを対応づける

■は対応のついたストロークのストローク間距離を示す

知識処理

- それこそ、数多の方法が存在する
  - サポートベクトルマシン
  - HMM
  - DNN（DL）



$(f_1 - f_2)$：パターン空間または特徴空間

x_i：パターンベクトル（または特徴ベクトル）

f_1, f_2：パターン（または特徴）を意味し，パターン（または特徴）空間を形成する軸になる．
これらの軸をこの空間の基底ともいう．

点線：クラス1と2を分ける決定境界

電子透かし

- 見た目には分からないが、検出ソフトを使用することで埋め込まれた情報を取り出すことができる仕組み
- 不正コピーやデータの改竄を見破ることができる
- 本物である証明に利用
- 紙媒体の場合、紙繊維の情報を利用するなど様々なレベル・方法がある

可視性	分類	特長	用途
見える	重畳印刷 （常時視認）	ヘッダ、フッタ、背景などに見えるよう情報を埋め込む	不正持ち出し抑止
見える	地紋印刷 （隠し文字印刷）	コピーするとあぶり出される、警告情報を埋め込む	不正コピー抑止 原本との識別
見えない	地紋透かし （見えない透かし）	見た目では分からないように背景地紋に埋め込む	情報漏洩 不正流通元の追跡
見えない	フォント透かし （見えない透かし）	見た目ではわからないようにフォントに埋め込む	情報漏洩 不正流通元の追跡

動画像圧縮

- 動画像の情報量
  - NTSCモノクロの場合、
    640(横)x480(縦)x8(諧調)x30(おおよそのフレーム/s)=約75Mbps
  - カラーの場合、その3倍
  - ハイビジョン1080iの場合、
    1920x1024x24x30=約1.5Gbps

- 動画像圧縮技術の基本
  - 差分符号化、フレーム間予測符号化
  - 動き補償予測
  - DCTの応用

MPEG

- MPEG-1
  - 転送レート 1.5Mbps
  - 圧縮率 35:1 家庭用VTRより多少劣る品質(CDV)
- MPEG-2
  - 転送レート 1.65～60Mbps
  - 圧縮率 40:1 アナログTV並みかそれ以上(DVD)
- MPEG-4
  - 転送レート 48～63Kbps
  - リアルタイムおよび双方向利用向け

- 動き補償予測と 2 次元DCTの組み合わせ
  - 動画像情報は次の 3 種類のピクチャに分けられる
  - I (Intra-coded)
    元の信号のまま、JPEGと同等の方式で圧縮
    10～15フレームごとに配置
    Iフレームが届けば、完全な画像がまずは構築できる
  - P (Predictive-coded)
    一方向動き補償予測に利用
    前の画像（IもしくはP）との差分情報
  - B (Bidirectionally predictive-coded)
    前後両方向の動き補償予測を与える

動画像圧縮の基本原理

- 動画像は複数のフレームと呼ばれる画像の集合により構成
- あるフレームと、その1/30秒前の直前のフレームでは、多くの場合似た画像となる
  - 人間は激しい動きに苦手＝情報量が多いから
  - だから、松陰寺 太勇は苦手（動画圧縮の敵）
- 直前のフレームをもとに、現フレームとの差分のみを抽出して符号化
  - フレーム間予測
  - 予測という名前がついているが、実際に予測しているわけではない
    どちらかというと仮定
- 画面内で動く物体を検出、その動きを予測し、予測結果と現フレームとの差を抽出すれば予測の精度が向上する
  - 動き補償

フレーム間予測符号化

- 連続するフレームは同じような内容
  - 静止領域が多い画像に対して効率的
- フレームメモリ
  - 1フレーム前の画像を記憶するメモリ

50	50	50	50	50	50
50	50	50	55	50	50
50	50	55	56	55	50
50	50	55	56	55	50
50	50	55	56	55	50
50	50	55	56	55	50

現在のフレーム画像

50	50	50	50	50	50
50	50	55	50	50	50
50	55	56	55	50	50
50	55	56	55	50	50
50	55	56	55	50	50
50	55	56	55	50	50

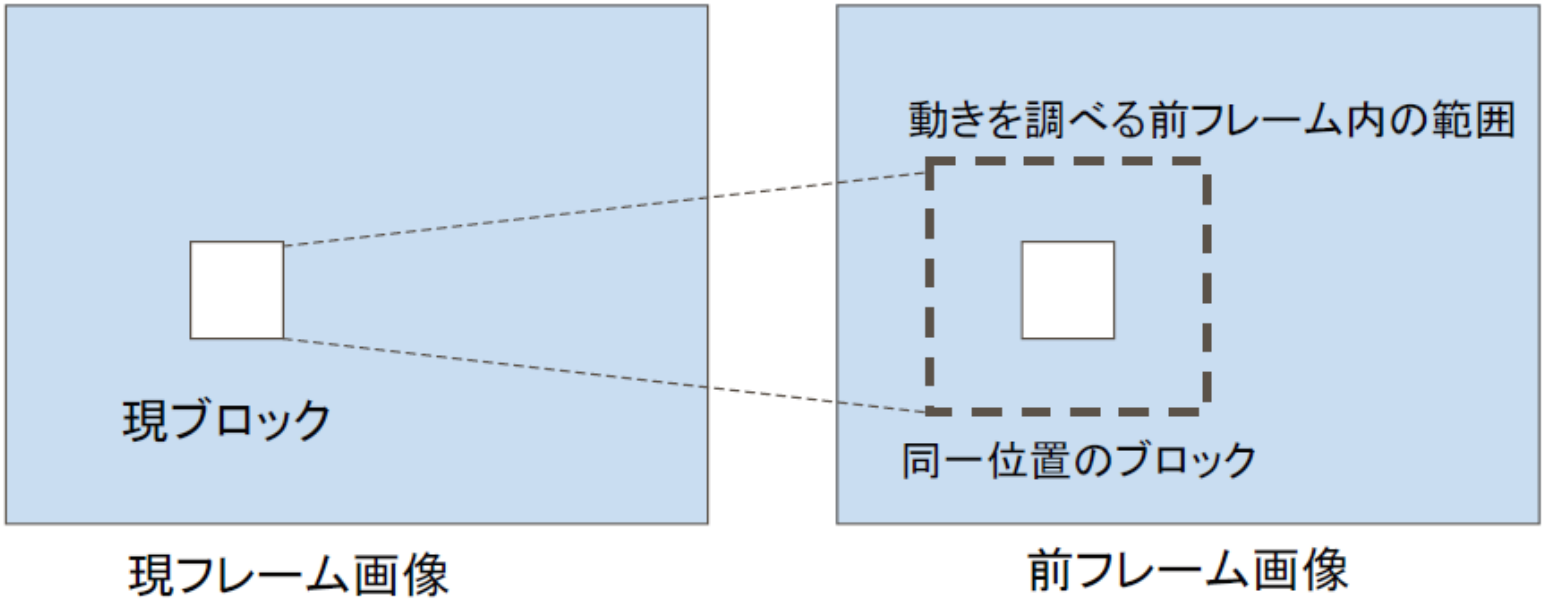
前のフレーム画像

0	0	0	0	0	0
0	0	-5	5	0	0
0	-5	-1	1	5	0
0	-5	-1	1	5	0
0	-5	-1	1	5	0
0	-5	-1	1	5	0

差分

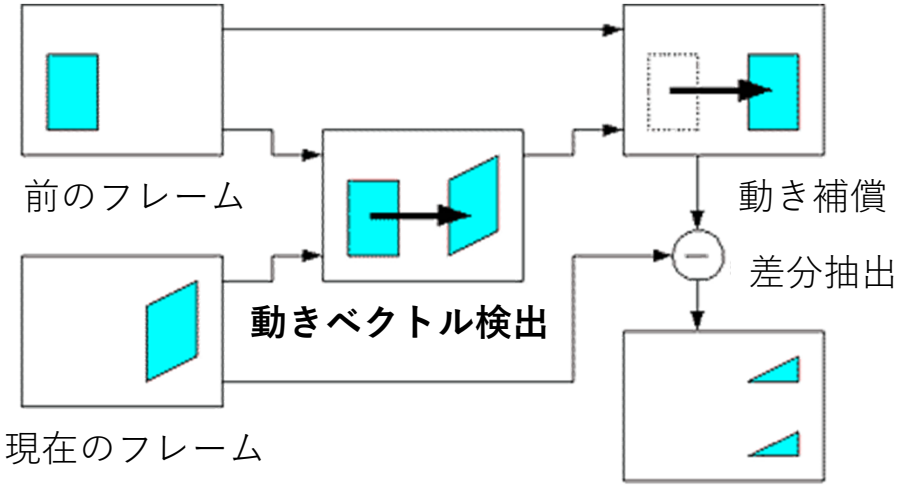
動き位置情報の検出

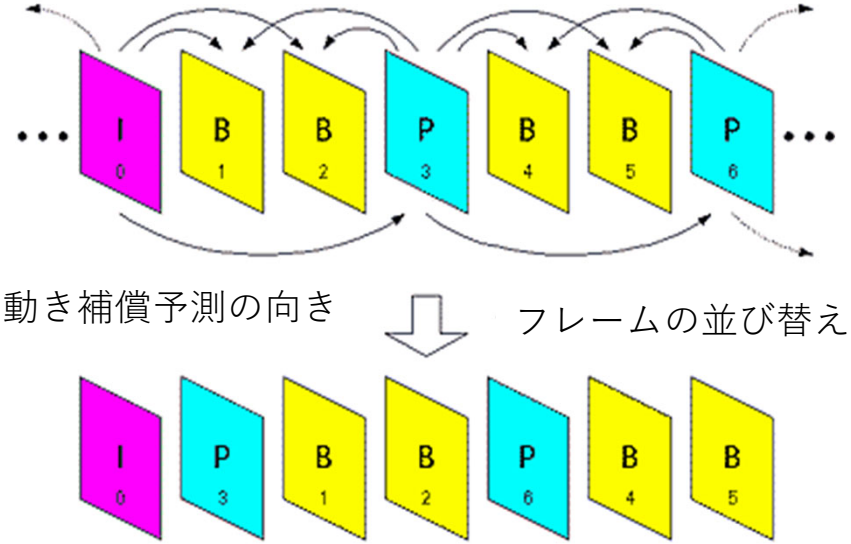
- 現フレームの画像を重なりのない16X16画素のブロックに分割
- 上下左右15画素以内の領域を動き補償範囲とする
- 予測誤差が最小のブロックを最適予測ブロックとして選択



動き補償とフレーム配置

- 動き補償処理の流れ
  - 現在のフレームと前のフレームから動きベクトルを抽出、その結果との差分情報を抽出し符号化する（データに含まれるのは動きベクトルと予測誤差）





- フレームの配置
  - Bピクチャは未来のフレームを予測に用いるため、その元となるPピクチャを先に送っておく

メディアの暗号化

- DVDは暗号化されており、直接データを取得できない
  - CSS（Content Scramble System）
    - 2012 年 10 月 1 日より違法
    - DeCSSが公開されプログラムの使用やソースコードの公表も違法となる
  - DeCSSで利用可能な素数を違法素数
    - 法的に利用禁止・公開不可な素数
  - 違法素数：$k \times 256^2 + 2083$や、Linux i385 ELFファイルとして直接実行可能な1811桁素数など、
    - これに便乗し、web上のあらゆる情報から自由にDeCSSコードを生成可能なプログラムを公開するプロジェクトもある
      - 結果あらゆる電子情報の利用が法的に禁じられたとする抗議が行われている
    - **このテキストも違法です**
- BD
  - AACS
  - タイトルキー：各タイトルに充てられ、コンテンツを暗号化した鍵で、漏えいしても当該タイトルしか復号化できない
  - ボリュームキー：タイトルキーを暗号化する鍵
  - メディアキー：ボリュームキーの生成に必要なメディア固有鍵、ディスクに数千個の束（メディアキーブロック）で格納
  - デバイスキー：プレーヤー等に固有の鍵、これを入手すると、全ソフトを復号可能
    - 漏えいしたデバイスキーに対応するメディアキーは一つで、メディアキーブロックの他のメディアキーは影響を受けない。つまり、漏えいしたデバイスキーを二度と使わなければよい
  - 現在も、いたちごっこは続いている

音声圧縮

- 標本化周波数（サンプリング周波数）
  - もとの周波数の 2 倍以上必要（標本化定理）
- 量子化雑音
  - 量子化する結果、原音とは差がどうしても存在する
  - 量子化雑音を低減するには、サンプリングするビット幅を増やす

電話並みの音声	5kHz
AMラジオ程度	8kHz
FMラジオ、TV音声	11kHz
中程度の品質	22kHz
Digital Audio (CD)	44.1kHz

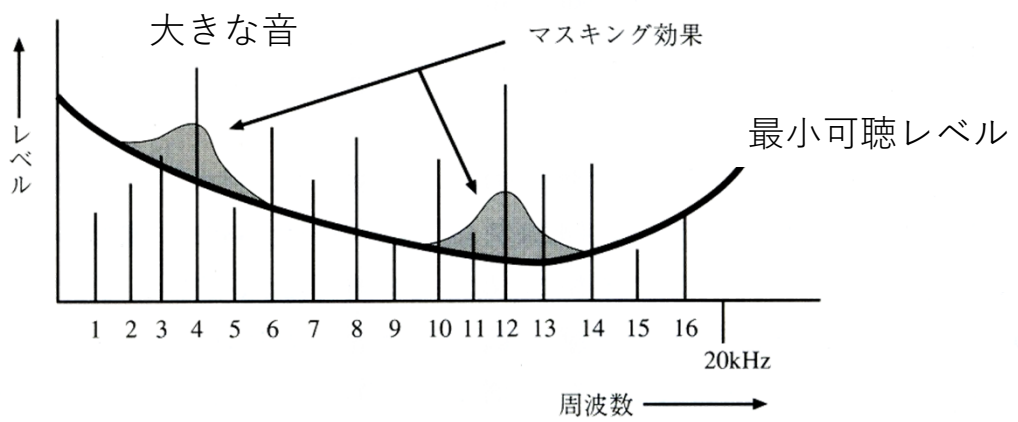
- ADPCM
  - 単純に量子化する方法
- ADCT
  - 離散コサイン変換で表現する方法

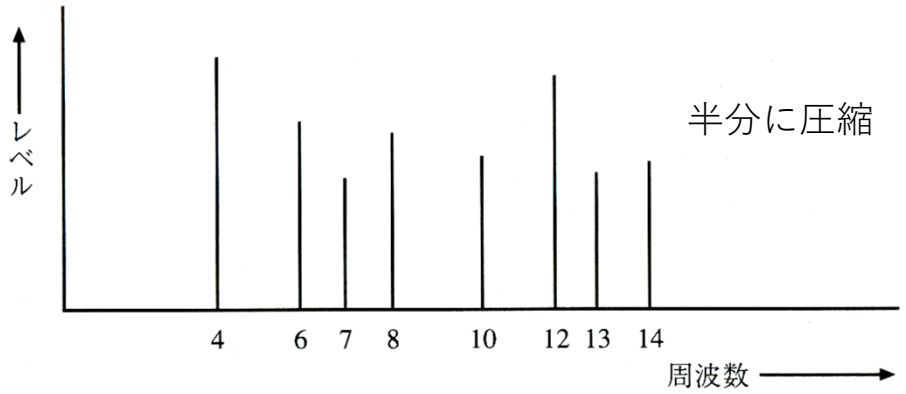
64kbpsPCM	ADPCM	8bit8kHz
32kHzPCM	ADPCM	4bit8kHz
CELP(16kbps),VSELP	DCT	

CD(Compact Disk)	PCM	16bit44.1kHz
MD(Mini Disk)	ATRAC	4bit44.1kHz
DCC(Digital Compact Cassette)	PASC	4bit44.1kHz
DAT(Digital Audio Tape recorder)	PCM	16bit48kHz等

音声圧縮におけるマスキング効果

- 人間の可聴レベルを図示（右図）
  - 周波数によって、ある程度大きな音でないと聞こえないというレベルが存在
  - 大きな音がした場合、その前後の周波数領域の音が聞こえなくなるマスキング効果が発生する（図の網部分）
- 例えば、
  - 可聴レベル以下の周波数情報は除去
    - 図の１，２，５，９，１５，１６
  - マスキング効果も考慮して除去
    - 図の３，１１
  - この図では、結果として情報が8/16つまり半分となる
- 周波数だけでなく時間的にも行う（周波数マスキング・時間マスキング）





mp3

- 音を周波数領域で表現する
- 可聴域を超える高周波領域、低周波領域の成分を削除する
  - ＬＰＦ・ＨＰＦ
- マスキング効果をつかって聴覚心理上聞こえにくい音を削除する
- 周波数領域で表現したままビット列に変換する
- 変換されたビット列をHuffman符号化する

- mp3や様々な動画圧縮など、**人間の感覚特性を利用する圧縮手法**は、エンコード技術が音質・画質を左右する
  - 同じ圧縮手法を用いている、異なるメーカの圧縮プログラムが、同じ性能（聴覚や視覚上の表現力の高さや品質）を示すわけではない

演習問題（９）

- Processingを用いエラー訂正の動作を確認しなさい
  - グラフィックに関係しないため、Processingが適切ではないが…
- エラー訂正はメディア情報の伝達において重要な要素技術である
  - ここでは、基本となるハミング符号を用いたエラー訂正手法を用いている
  - Gは生成行列、Hは検査行列と呼ばれる

演習(9-1) 次のページのコードを入力し動作を確認しなさい

演習(9-2) プログラム先頭2行にあるwordが送りたい数列、pがエラーを入れる場所を示している。pはなぜ0から6までの値であるか？説明しなさい

演習(9-3) wordやpを変更し正しくエラーが訂正されているか確認しなさい

演習(9-4) pを0から7まで変化させたときのsyndromeの値と検査行列がどのような関係にあるかを調べなさい

演習(9-5) コードを変更して無理やり2箇所間違えるように修正し、エラーが訂正できるかどうか調べなさい

```
int word[] = {0, 0, 0, 0}; // input vector (4bits)
int p = 3; // error position (0-6)

int G[][] =
  {{1, 0, 0, 0, 0, 1, 1},
  {0, 1, 0, 0, 1, 0, 1},
  {0, 0, 1, 0, 1, 1, 0},
  {0, 0, 0, 1, 1, 1, 1}};
int H[][] =
  {{0, 0, 0, 1, 1, 1, 1},
  {0, 1, 1, 0, 0, 1, 1},
  {1, 0, 1, 0, 1, 0, 1}};
int cw[] = new int[7]; // codeword
int sy[] = new int[3]; // syndrome
for (int j = 0; j < 7; j++) {
  cw[j] = 0;
  for (int i = 0; i < 4; i++)
    cw[j] += word[i] * G[i][j];
  cw[j] = cw[j] % 2;
}
println("code word is "+cw[0]+cw[1]+cw[2]+cw[3]+cw[4]+cw[5]+cw[6]);
if ((p >= 0) && (p < 7)) cw[p] = (cw[p] + 1) % 2;
println("word with an error is "+cw[0]+cw[1]+cw[2]+cw[3]+cw[4]+cw[5]+cw[6]);
for (int i = 0; i < 3; i++) {
  sy[i] = 0;
  for (int j = 0; j < 7; j++)
    sy[i] += cw[j] * H[i][j];
  sy[i] = sy[i] % 2;
}
println("syndrome is "+sy[0]+sy[1]+sy[2]);
int l = 0;
for (int i = 0; i < 3; i++) {
  l *= 2;
  l = l + sy[i];
}
if (l != 0) {
  println("error found at "+(l-1));
  cw[l-1] = (cw[l-1] + 1) % 2;
  println("the original word is "+cw[0]+cw[1]+cw[2]+cw[3]);
} else {
  println("No error");
}
```

注意事項

- これらの設問に対する回答を、Microsoft Wordファイルで作成
- LMSで提出すること
- A4で作成すること
  - ソースコードの記載はコピペになるため不要で、どのような入力値を使ったかや、実行結果を示せば十分です。なお、常識的な範囲でページ数を超過することは問題ありません
- **ソースコード、動作画面をキャプチャして貼り付けること**
  - キャプチャはキャプチャしたいWindowを選択してAlt+Print Screenでできます。そのまま、Wordファイルに張り付けることができます（＋は押しながらの意味）
  - MacOSは、Control+Command＋Shift＋4とするとカーソルがカメラ型に変わります。特定のウインドウやメニューバーをクリックしてキャプチャ、画像はクリップボードに転送されます
- 最初にタイトルとして「演習問題（９）」と書き、名前と学籍番号を記載すること
  このフォーマットに従っていないレポート答案は受け取らない
- 締め切りなど詳細はLMSを確認すること