

# 情報工学演習I 第4回



TeX

# 講義目標と参考資料

## ▶ 講義目標

TeX を使ってレポートを書けるようになるろう！

## ▶ 参考資料

奥村晴彦・黒木裕介, “[改訂第9版] LaTeX (ラテック) 美文書作成入門”, 技術評論社, 2023 年 12 月 9 日, ISBN-13: 978-4-297-13889-9

1冊もっていると論文執筆に便利！  
購入をおすすめします！  
電子版もあるよ！



# 内容

---

- ▶ TeX とは？
- ▶ Overleaf
- ▶ 数式，表，図

# TeX とは？

# TeXとは？

---

## ▶ フリーの組版ソフト

- ▶ 文字や図表を配置し，紙面を構成する.
- ▶ 「テフ」，「テック」と読む

TeX

## ▶ 特徴

- ▶ OS に依存せず利用可能
- ▶ TeXファイルの実態はテキストファイルなので，テキストエディタで編集可能
- ▶ 自動ハイフネーション，ペアカーニング，リガチャ，孤立行処理などの高度な組版技術
- ▶ 数式の組版で定評あり

# TeX で書いた論文例

## Performance Evaluation of Face Recognition in the Wavelet Domain

Yuzuko Utsumi

Graduate School of Engineering Science  
Osaka University  
yuzuko@yachi-lab.sys.cs.osaka-u.ac.jp

Yoshio Iwai

Graduate School of Engineering Science  
Osaka University  
iwai@sys.cs.osaka-u.ac.jp

Masahiko Yachida

Graduate School of Engineering Science  
Osaka University  
yachida@sys.cs.osaka-u.ac.jp

**Abstract**—In recent years, many image features have been used for face recognition. The Gabor wavelet feature is a well known image feature used in face recognition systems because its recognition rate is better than that of the systems using Eigenface. However, it remains unclear whether Gabor wavelet features are indeed the best wavelet features for face recognition.

In this paper, we extract image features of facial images from various wavelet transforms (e.g., Haar, French hat, Mexican hat, Daubechies, Coiflet, Symlet, and O-spline), and evaluate face recognition performances. We also compare the recognition performance of fixed scale wavelet features with that of adaptive scale wavelet features. The results show that the performance of the wavelets we used is almost the same as that of the Gabor wavelet, and the performance of adaptive scale wavelet features is better than that of fixed scale ones.

### I. INTRODUCTION

In recent years, demands have increased for security systems to be more effective in various high-security situations. Typical security systems involve keys, cards, and pin numbers. A key is a classical method, but less secure because of the risk of lost or counterfeit keys, and pin numbers have commonly become used instead of keys. However, pin numbers can easily be forgotten, so people tend to set pin numbers that can be easily remembered, e.g. birthdays, phone numbers and so on. Such pin numbers are not really secure. Further, with cards, information is easily stolen by using a skimmer and so card counterfeiting is very common.

Therefore, researchers have developed security systems using personal biometric authentication technologies such as fingerprints, venation patterns, and iris and face recognition[8]. By using biometric information, the risks involved using keys, cards, and pin numbers can be reduced and so security systems made more secure.

There are two general methods for fingerprint recognition[9], [10]; one captures a fingerprint with a camera through a small prism and then recognizes the fingerprint pattern, the other uses a semiconductor sensor to measure the capacitance of a finger from a fingerprint. A semiconductor sensor is very small and is not so expensive; therefore, this method has been applied to mobile phones and notebook computers. Some users, however, are uncomfortable about having their fingerprints checked.

The venation pattern of fingers and the hand is recognized by capturing veins with infra-red illumination[11]. Counter-

feiting this is difficult because a venation pattern is with a particular body, and there is no real need to actually contact a sensor to capture a venation pattern. In recent years, venation pattern recognition has been applied to person authentication in automatic teller machines (ATMs) because the sensor identification without any contact is seen as hygienic and without significant psychological burden for users.

Iris recognition is performed by capturing an iris pattern with a camera[12]. The iris pattern is fixed at about two years old, and is said to be the same throughout a person's life. Thus, once a user's iris pattern is registered into a database, the user can be recognized without updating the iris pattern. A point in common with the above biometric methods is that biometric information must be correctly placed on a sensor to capture the pattern; the registration problem is often not considered. It is therefore important that users must carefully register their biometric information into the sensor.

Cameras have usually been utilized as sensors for face recognition because usually they do not make people feel uncomfortable. Many researchers have developed face authentication systems that consist of two modules: facial feature extraction and face recognition modules. PCA[14], ICA[13], and LDA[16] are well known approaches to face recognition that use feature sub-spaces, and HMM[20], SVM[19], and Ensemble learning[21] are similarly well known face recognition methods.

Many features can be extracted and used for face recognition; for example, phase and power of FFT[17], coefficients of DCT[15], [2], and the Gabor wavelet[4]. The Gabor wavelet has a similar characteristic ton that of the human retina, and is said to be robust feature even with illumination changes. The face recognition performance of the Gabor wavelet method has been shown to be better than that of Eigenface[4]. However, in previous works, comparison of the Gabor wavelet with other wavelets has not been performed for face recognition; rather comparisons have been performed against fingerprint recognition[5]. Therefore, in this paper, we compare the recognition performances of the Gabor wavelet with those of various other wavelets; i.e., Haar, French hat, Mexican hat, Daubechies, Coiflet, Symlet, and O-spline. We also compare the recognition performance of fixed scale wavelet features with that of adaptive scale ones, and then evaluate the effect of the accuracy of feature positions on recognition performance.

## 自動ハイクネーション

authentication  
ie sensor **iden-**  
ic and without

## ペアカーニング(文字詰め)

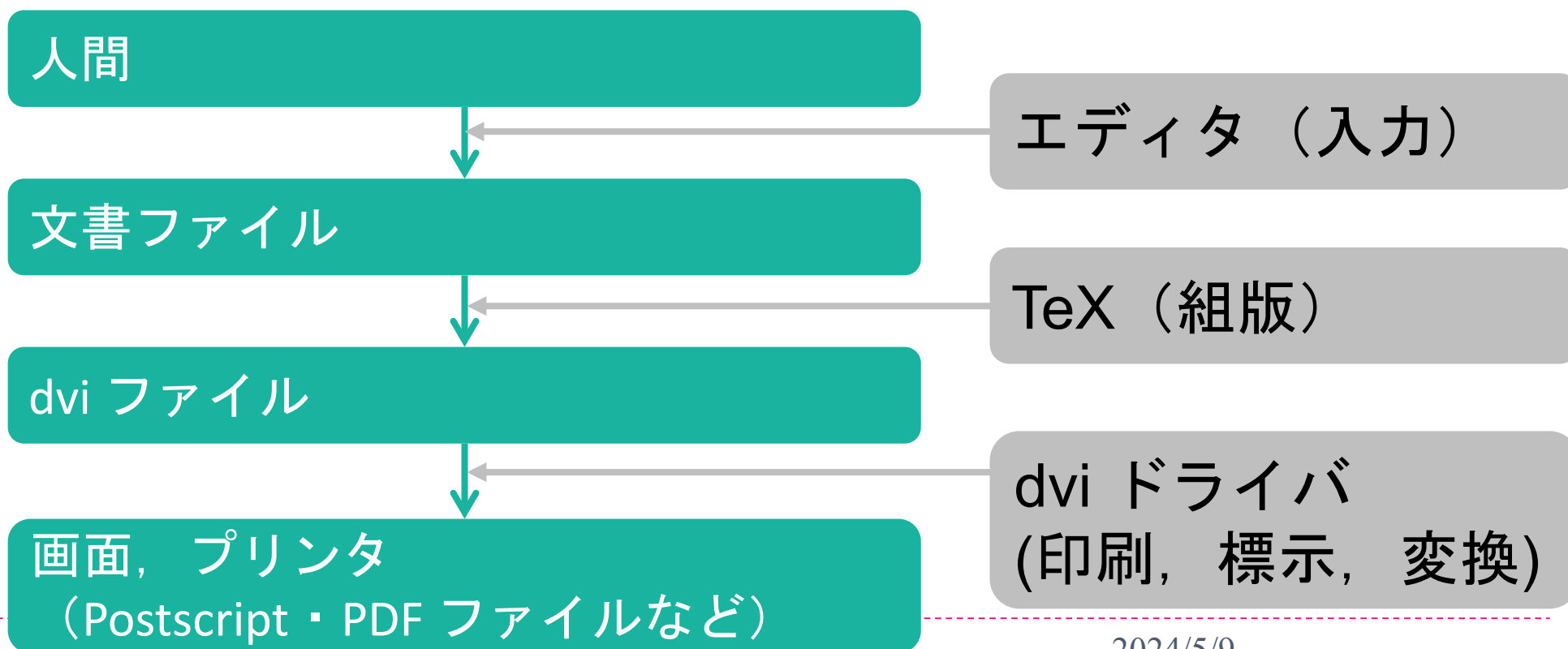
**Transform**

## リガチャ(合字)

**fingerprints**

# TeX の処理方式

- ▶ 一括して全体を処理するバッチ処理
  - ▶ ワードプロソフト（ex. Microsoft word）はキーボード入力が終わるたびに表示
- ▶ 処理の流れ





**Overleaf**



# Overleafとは？

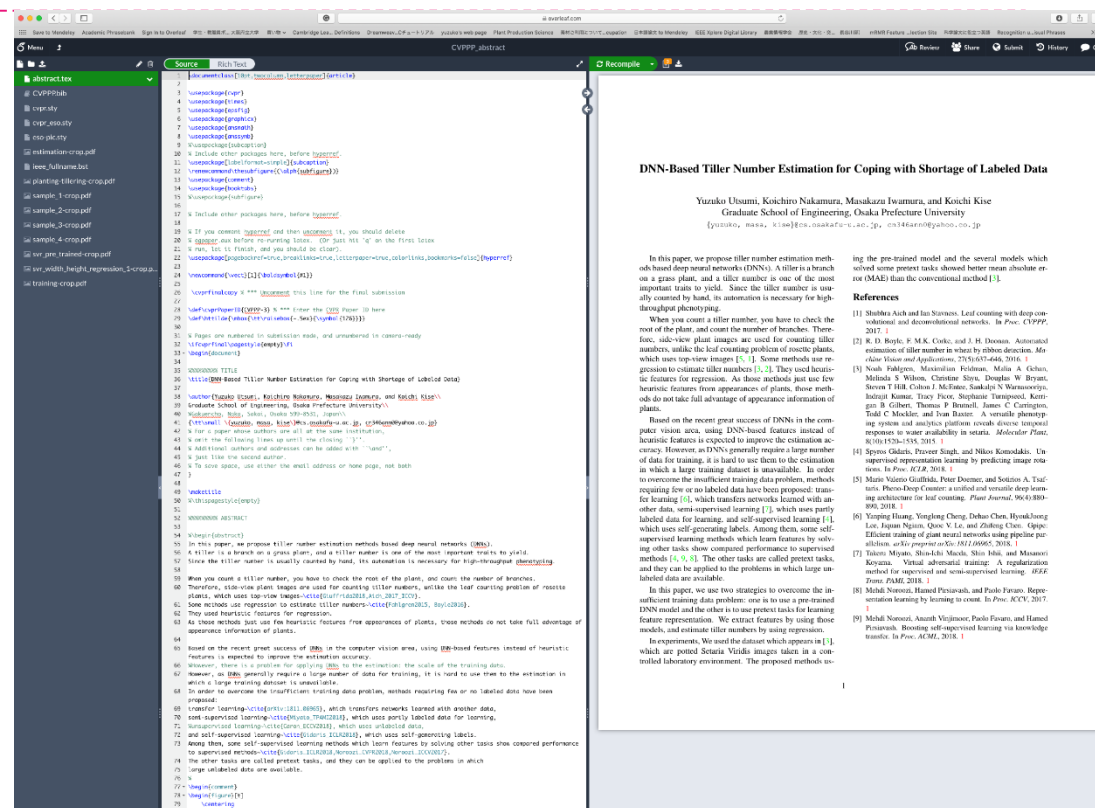
► Web ブラウザ上のLaTeX 執筆環境

## ▶ 利点

- ▶ ローカルにTeX 環境を構築せずにご利用可能
- ▶ リアルタイムでプレビュー可能
- ▶ 共同編集機能
- ▶ 差分管理機能

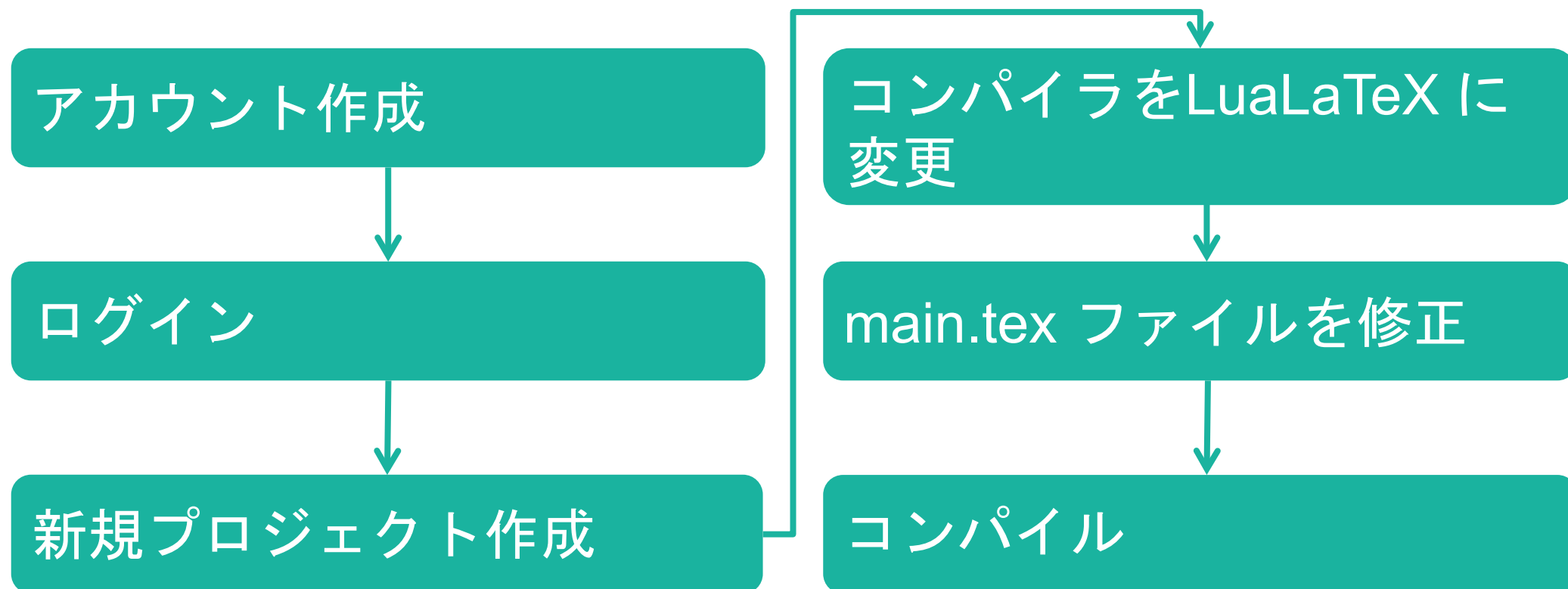
## ▶ 欠点

- ▶ サーバが時々落ちたりメンテナン  
スで利用不可能になる  
(論文投稿締め切り直前に  
使えなくなると悲惨)



# Overleafの使用方法

---



# OverLeaf の利用方法（日本語文書の入力）

---

## ▶ LuaLaTeX

- ▶ pdfTeX をスクリプト言語 lua を使って軽量化したもの.
- ▶ LaTeX ソースから PDF を直接出力する

## ▶ LuaLaTeX で日本語を使いたい

- ▶ `\documentclass{ltjsarticle}` を使う
- ▶ main.tex 編集の際に修正する
- ▶ 自動的に日本語用の luatexjp パッケージが読み込まれる
- ▶ コンパイルして日本語が表示されることを確認する

参考: TeX Wiki LaTeX 入門 / 発展編 LuaLaTeX

<https://texwiki.texjp.org/?LaTeX%E5%85%A5%E9%96%80%2F%E7%99%BA%E5%B1%95%E7%B7%A8#mb665da8>

数式， 表， 図

# 数式を書く

- ▶ \$で挟む
  - ▶ 文章中で数式用の書体として記述
- ▶ ¥begin{equation}, ¥end{equation}を使う
  - ▶ 行替えを行い，数式番号が自動的につく

## ソース

```
\section{数式}
文書中に数式 $a=b+c$ を書きたい場合は，\$で数式を挟みます．
行替えを行って数式を書く場合は，\verb|\begin{equation},
\end{equation}|を使います．
\begin{equation}
a=b+c
\end{equation}
```

## コンパイル結果

### 2 数式

文書中に数式  $a = b + c$  を書きたい場合は，\$で数式を挟みます．行替えを行って数式を書く場合は，`\begin{equation}`，`\end{equation}`を使います．

$$a = b + c \tag{1}$$

# 数式で必要な表現

## ▶ 累乗, 添字

▶  $x^2$  : `$x^2$`

▶  $x^{10}$  : `$x^{10}$`

▶  $A_n$  : `$A_n$`

## ▶ 分数

▶  $\frac{1+x}{2-y}$  : `\frac{1+x}{2-y}`

### ソース

色々な数式表現 :

`$x^2$`, `$x^2$`, `$x^2$`, `\frac{1+x}{2-y}`

### コンパイル結果

色々な数式表現 :  $x^2$ ,  $x^{10}$ ,  $A_n$ ,  $\frac{1+x}{2-y}$

必要に応じてWebで調べながらやってみよう  
キーワード : tex, 数式, など

# 表の作成

## ▶ table, tabular 環境の利用

```
\begin{table}[tb]
\caption{プロ野球セントラルリーグ2016年9月29日の順位}
\centering
\begin{tabular}{|l|l|l|l|l|}
\hline
順位&チーム名&勝率&差\\
\hline
1&広島東洋&.626&-\\
\hline
2&読売&.518&15.0\\
\hline
3&横浜DeNA&.496&18.0\\
\hline
4&東京ヤクルト&.450&24.5\\
\hline
5&阪神&.449&24.5\\
\hline
6&中日&.414&29.5\\
\hline
\end{tabular}
\end{table}
```

### 表の出力位置指定記号

h: その位置  
t: ページ上端  
b: ページ下端  
p: 単独ページ

### 表のタイトル

### 列指定

l: 左寄せ  
c: 中央  
r: 右寄せ  
| は縦に入れる線を表す(Shift+¥)

¥hline: 横線の描画指示

&: 列の区切り

¥¥: 行替え

# 表のあれこれ

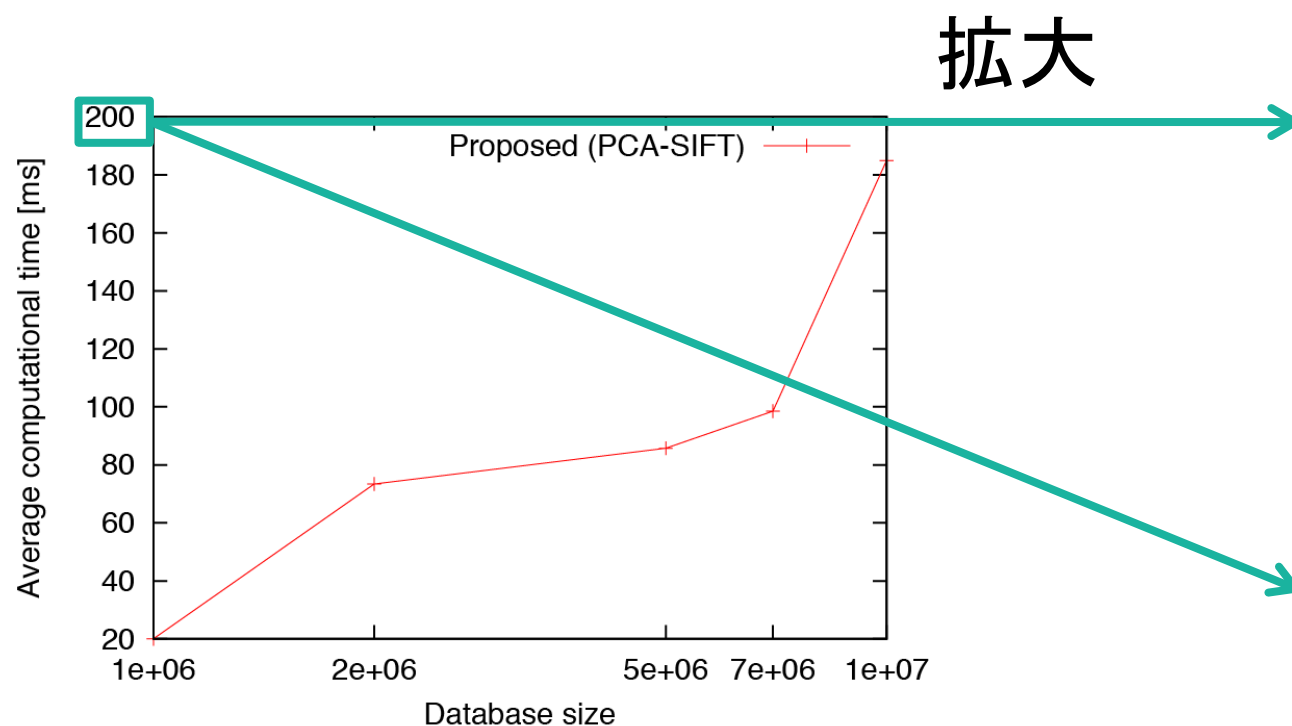
---

- ▶ 表の書き方もいろいろ
- ▶ 自分で参考書やweb で調べて思い通りの表を書けるようになろう



# 図の形式

- ▶ TeX で利用出来る画像形式
  - ▶ ベクトル画像：PDF（おすすめ）
  - ▶ ビットマップ画像：JPEG, PNG, BMP（使って欲しくない）



ベクトル画像

200

ビットマップ画像

200

2024/5/9

# ベクトル画像の作成

---

- ▶ 図の描画（ドロー系ソフトウェア）
  - ▶ OpenOffice.org Draw
  - ▶ Illustrator
  - ▶ OmniGraffle
- ▶ グラフの描画
  - ▶ gnuplot（グラフ作成ソフト）
  - ▶ Microsoft Excel（表計算ソフト）
  - ▶ Python（matplotlib, pandas などのライブラリ）
- ▶ 画像の変換（ビットマップ → ベクトル）
  - ▶ Illustrator (有料ソフト), ImageMagick (Mac)
  - ▶ ウェブ上の変換サービス (使用の際には注意)

# 図のはり方

プロジェクトに画像ファイルをアップロード



main.tex に画像貼り付けの設定を追加

```
¥usepackage{graphicx}
```



main.tex にincludegraphicコマンドを記入

```
\section{図の貼り方}
図はincludegraphicsを使って貼り付けします。
TeXで用いられる画像ファイル形式は主にPDFです。
\begin{figure}[tb]
  \centering
  \includegraphics[width = .8\hsize]{./shakedrop.pdf}
  \caption{Shake Drop 博士}
  \label{fig:my_label}
\end{figure}
```

## 第4回課題 (1/3)

以下をまとめたレポートをTeXを用いて作成し，PDFファイルを提出せよ．

1. つぎの定積分を計算せよ

$$\int_{-\infty}^{\infty} \frac{1}{x^2 + 1} dx$$

- ▶ レポートには導出経過の数式も書くこと
- ▶ **equation** 環境を使うこと
- ▶ 複数の行に渡る場合，**split** 環境を使うと便利  
(ただし，`\documentclass{ltjsarticle}` の下に `\usepackage{amsmath}` を入れる必要あり)

## 第4回課題 (2/3)

2. 以下のページをもとに，工学域の学年ごとの3つの学類と工学域の人数を示す表を作成せよ．表のタイトルも必ずつけること．

<https://www.osakafu-u.ac.jp/info/outline/number/>

学域	学類	1年	2年	3年	4年	5年	6年	合計
工学域	電気電子系学類	0 (0)	0 (0)	211 (7)	167 (5)	-	-	378 (12)
	物質化学系学類	0 (0)	0 (0)	185 (31)	152 (21)	-	-	337 (52)
	機械系学類	0 (0)	0 (0)	179 (12)	133 (8)	-	-	312 (20)
	工学域計	0 (0)	0 (0)	575 (50)	452 (34)	-	-	1,027 (84)

文字揃えがされていればなお良い

## 第4回課題 (3/3)

---

3. 現在興味のあることや気になることを図を交えてレポートに書け。ただし、本文中に図の説明を必ず書くこと。
  - ▶ 図は写真でも絵でもよい
  - ▶ 図は必ずPDF ファイルを貼り付けること
  - ▶ 必ず自分で作文すること

# 課題のフォーマットと注意点

---

## ▶ 課題のフォーマット

授業支援システムにアップロード (**sample.tex**) しているので、適宜利用して下さい

## ▶ 注意点

**Web** サイトや友人のレポート等から文章をコピーしていることが明らかになった場合、第 **4** 回の課題すべて **0** 点とする

### ▶ これまでのコピー例



# 提出に関して

---

- ▶ 提出するもの
  - ▶ TeX で作成した **PDF ファイル**
- ▶ ファイル形式とファイル名
  - ▶ report-4-学籍番号.pdf
- ▶ 提出期限
  - ▶ 2024 年 5 月 15 日 (水) 23:59
- ▶ 提出方法
  - ▶ 授業支援システムから提出
- ▶ レポートのタイトルには以下を含むこと
  - ▶ title: 第 4 回課題
  - ▶ author: 学籍番号と氏名
  - ▶ date: レポートを仕上げた日 (`\date{\today}`とすると出力される)