データと数理I
第1回/第2回

序章/1章 複素平面で画像処理

武蔵野大学 データサイエンス学部データサイエンス学科
中西 崇文

ローカルルール

- 本授業はGoogle Classroom, Slack, Google Colaboratoryを使用します。
- 講義中なにかあればSlackのチャンネル「#120_データと数理_2023」にこちらからも流しますし、みなさんもこちらに発言してください。
- グループワーク課題資料をGoogle classroomで提出する際は、その資料の表紙、および提出時のメッセージに、各メンバーがどの部分を貢献したかとその貢献度%(全体を100%として)を記述して提出してください

全員入手すること

- 中西崇文, Pythonハンズオンによるはじめての線形代数, 森北出版, 2021.

- https://www.morikita.co.jp/books/mid/085581
- https://www.amazon.co.jp/dp/4627855818/ref=cm_sw_r_tw_dp_RXYHC14YH7BX1C6J8P42?_encoding=UTF8

本授業では、これを「教科書」と呼びます。

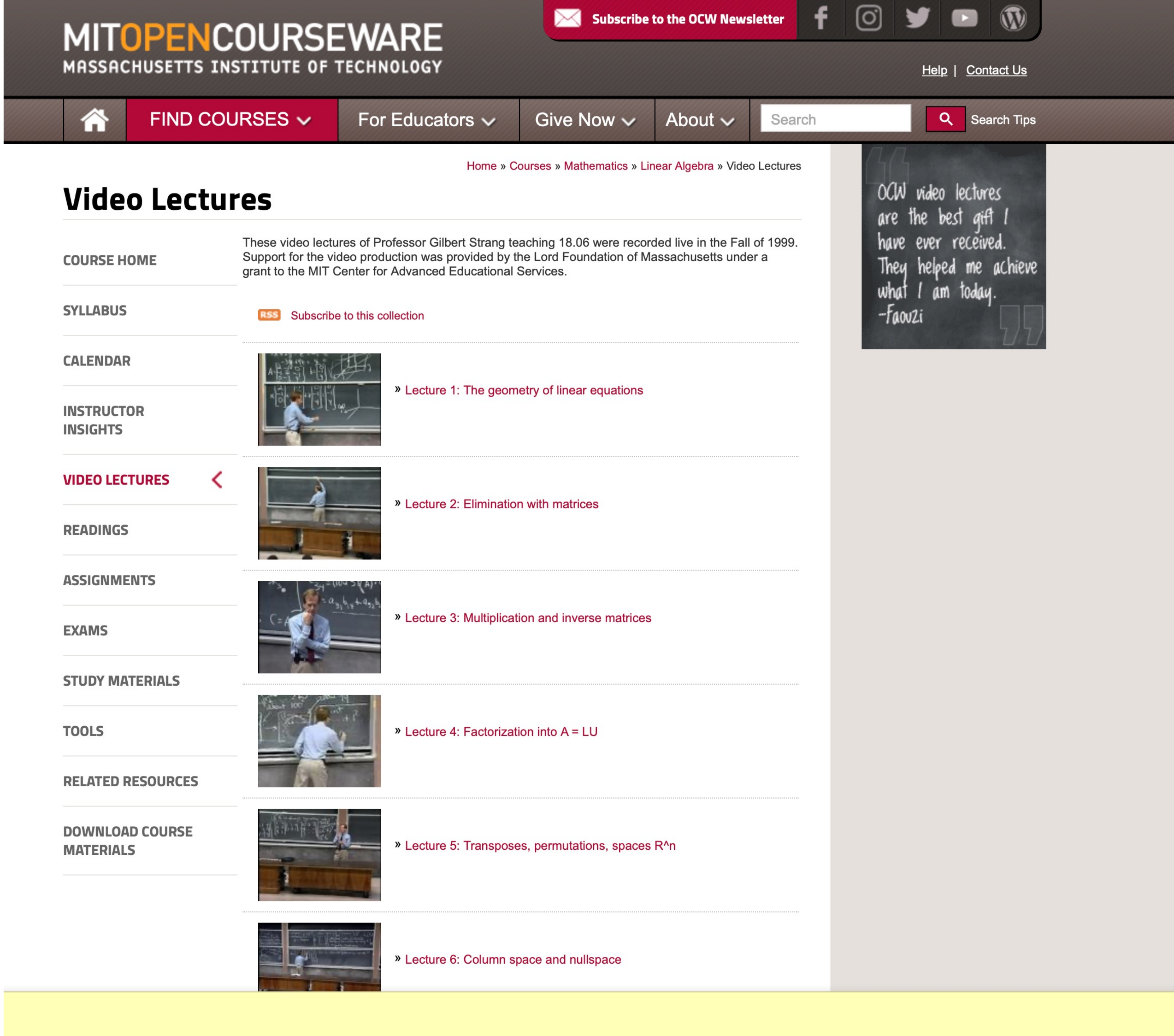
序章

2年生数学系科目

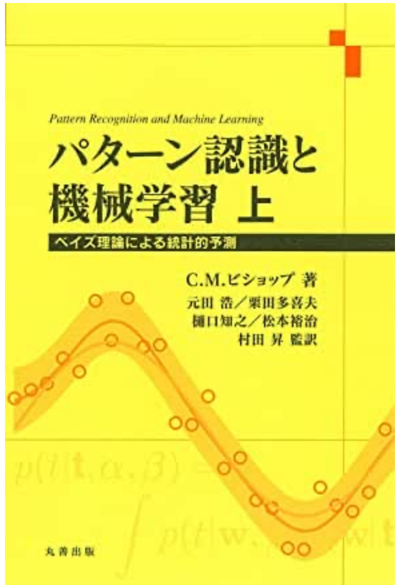
- データと数理I→線形代数
- データと数理II→微分積分
- データと経済統計 →統計学
- データと計量経済学 →ベイズ統計

良い講義

- https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/video_galleries/video-lectures/



データサイエンスの3大教科書
赤本、黄本、緑本



データと数理I	○	○	○
データと数理II	○	○	○
データと経済統計	◎	○	
データと計量経済学		◎	◎

プログラミング、グループワークあり

- 数学をただの理論として終えるのではなく、実際に使える力へ
  - 実際に使える力→数学の概念をプログラムに落とすこと
    - Python→Google Colaboratory

- 詳細な証明は参考書等に譲ることあり

講義の進め方

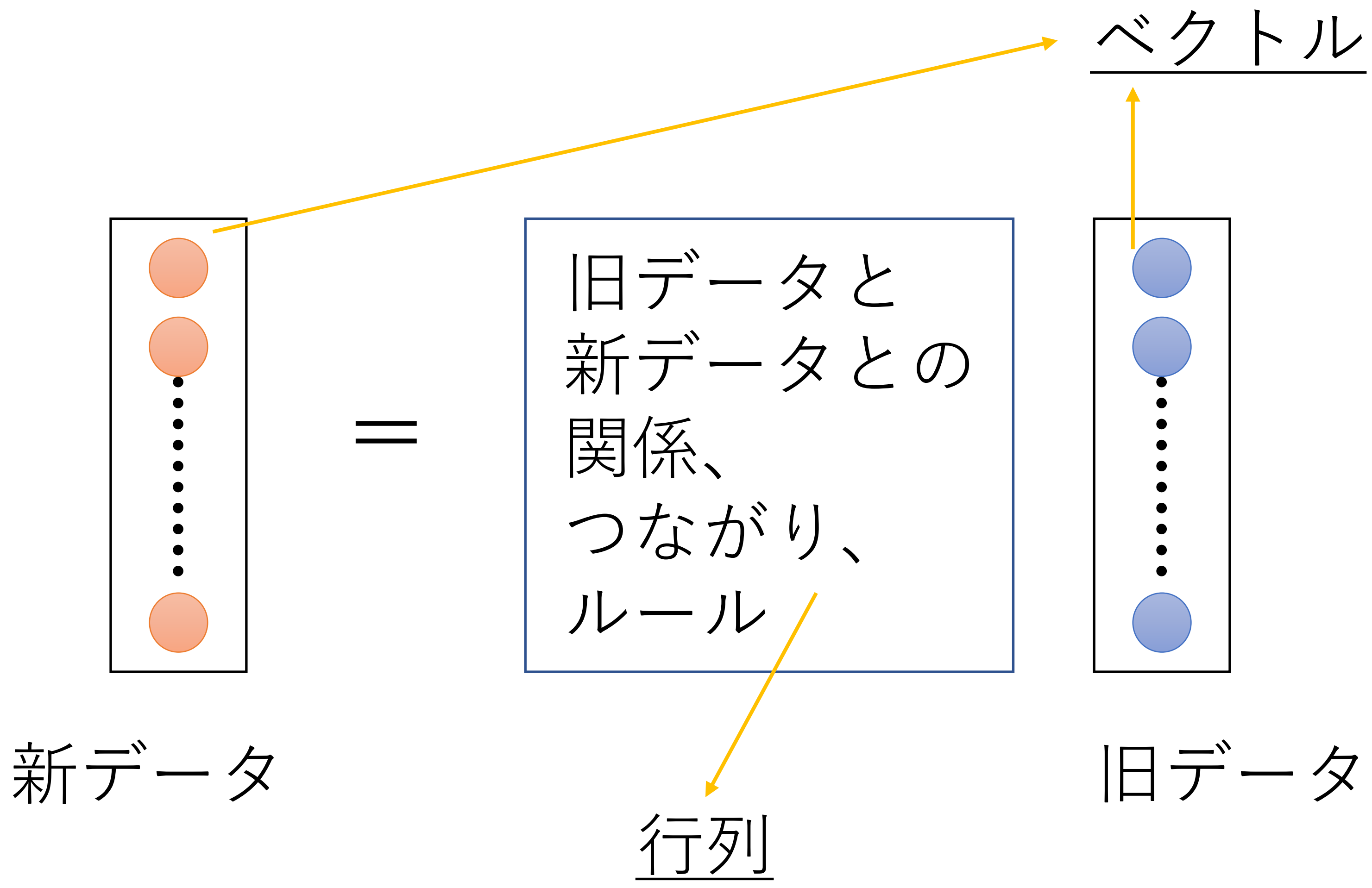
- 1. イントロダクション
- 2. 複素平面をつかった画像の拡大、回転、平行移動
  - 線形代数で実現可能な、画像の拡大、回転、平行移動の操作について、高校までに習得した複素平面を利用して実現する方法を確認し、Pythonでプログラミングしながら実践する。
- 3. ベクトルの基本/ノルム、距離、内積
  - ベクトルの概念、演算について復習した上で、具体的にベクトルをPythonでプログラミングしながら確認する。 ベクトルの「方向」と「大きさ」を抽出したり、2つのベクトル間の関係の「大きさ」を表現する方法として、ノルム、距離、内積の概念を学んだ上で、それらを応用した類似検索の実現例をPythonでプログラミングしながら実践する。
- 4. 行列の基本、行列式、逆行列、連立1次方程式
  - 行列の概念、演算について学んだ上で、具体的にベクトルをPythonでプログラミングしながら確認する。 行列式、逆行列やその周辺の概念を学んだ上で、それらを実際に活用した連立方程式の解法について学修する。
- 5. 線形変換
  - 行列演算で実現される線形変換の世界、概念を学んだ上で、具体的な実践例について議論する。 また、線形変換を使った応用についてグループワークで考え、さらにPythonでプログラミングすることにより、実際のアプリケーションを構築する。また、その成果をプレゼンテーションし、議論する。
- 6. アフィン変換ー画像の平行移動、拡大・縮小、回転、せん断、鏡映
  - 第1回、第2回で扱った画像処理をここまでで扱った線形代数の知識を応用してアフィン変換を学び、再定義する。アフィン変換をPythonでプログラミングすることにより理解する。
- 7. 固有値・固有ベクトル
  - 固有値分解の概念と演算方法について学ぶ。特に演算方法については、演習も行う。 様々な固有値分解の応用を示した上で、どのような応用が考えられるかグループワークで議論する。

Primitives for Data Feature Engineering

- <u>Transformation</u>
  - 特徴を変換する
    - T. Kitagawa and Y. Kiyoki, "Fundamental framework for media data retrieval system using media lexico transformation operator," Information Modelling and Knowledge Bases, vol. 12, pp. 316–326, 2001.
    - 岡田 龍太郎, 中西 崇文, 本間 秀典, 北川 高嗣, "メディアコンテンツを対象とした統計的一般化逆作用素構成方式とその楽曲メディアコンテンツ生成への適用,"情報処理学会論文誌, 57(5), pp. 1341 – 1354, 2016
- <u>Measurement</u>
  - 特徴を計量する
    - Y. Kiyoki, T. Kitagawa, and T. Hayama. "A metadatabase system for semantic image search by a mathematical model of meaning," *SIGMOD Rec.* 23, 4 (December 1994), pp. 34–41, 1994.
    - T. Kitagawa and Y. Kiyoki, "A mathematical model of meaning and its application to multidatabase systems," *Proceedings RIDE-IMS `93: Third International Workshop on Research Issues in Data Engineering: Interoperability in Multidatabase Systems*, Vienna, Austria, pp. 130-135, 1993.
- <u>Transition</u>
  - 特徴の変化の過程をみる

線形代数とは

旧データから新データへの変換(Transformation)を追求する学問



どういうものかを知るための例

	桃	みかん	梨
値段(円)	200	100	150
重さ(g)	300	200	250

$$\begin{bmatrix} 850 \\ 1450 \end{bmatrix} = \begin{bmatrix} 200 & 100 & 150 \\ 300 & 200 & 250 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

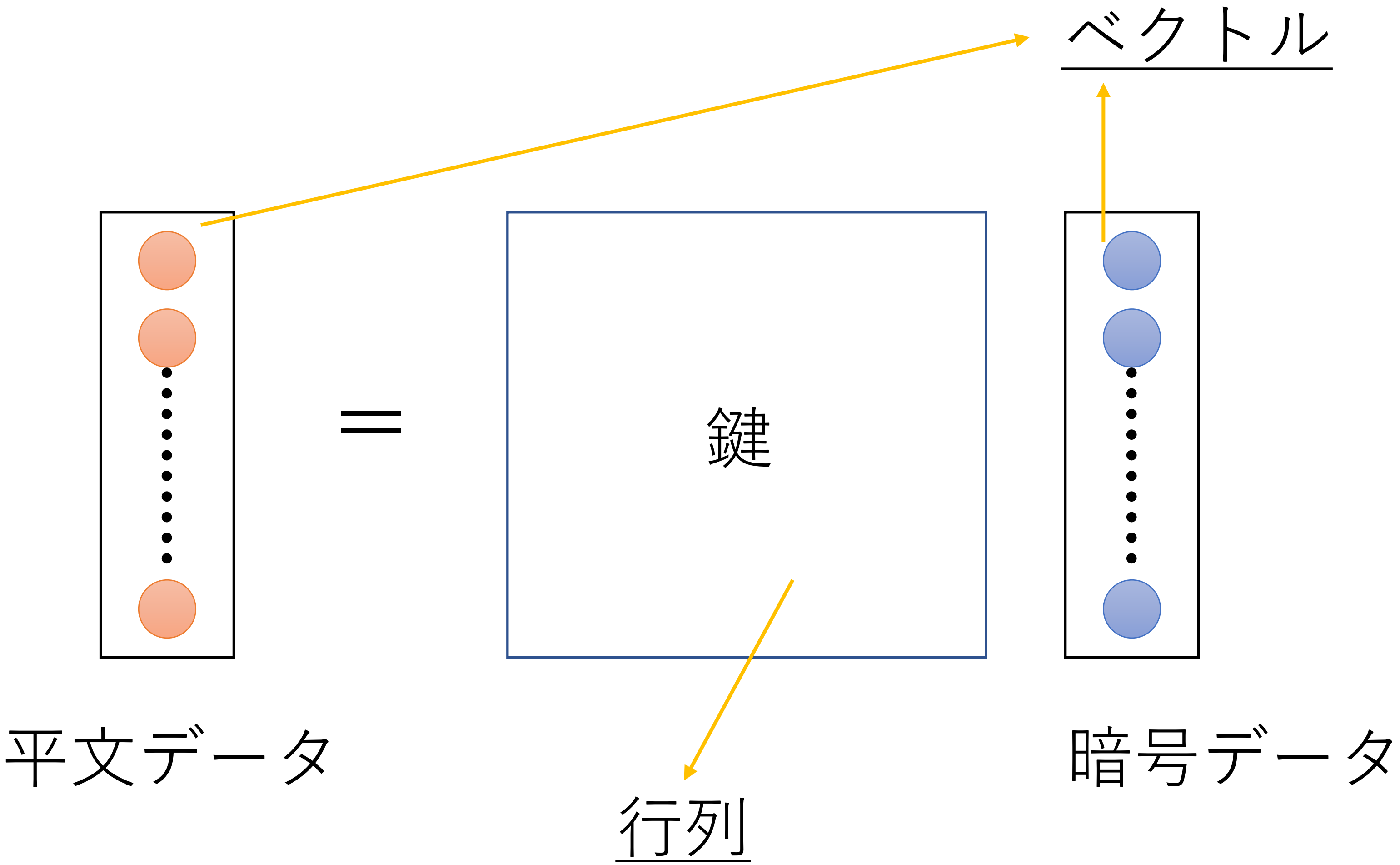
ベクトル(新データ)
値段850円、重さ1450g

行列(ルール・関係)

ベクトル(旧データ)
桃2箱、みかん3箱、梨1箱

柴田正憲, 貴田研司, “情報科学のための線型代数,” コロナ社, 2009

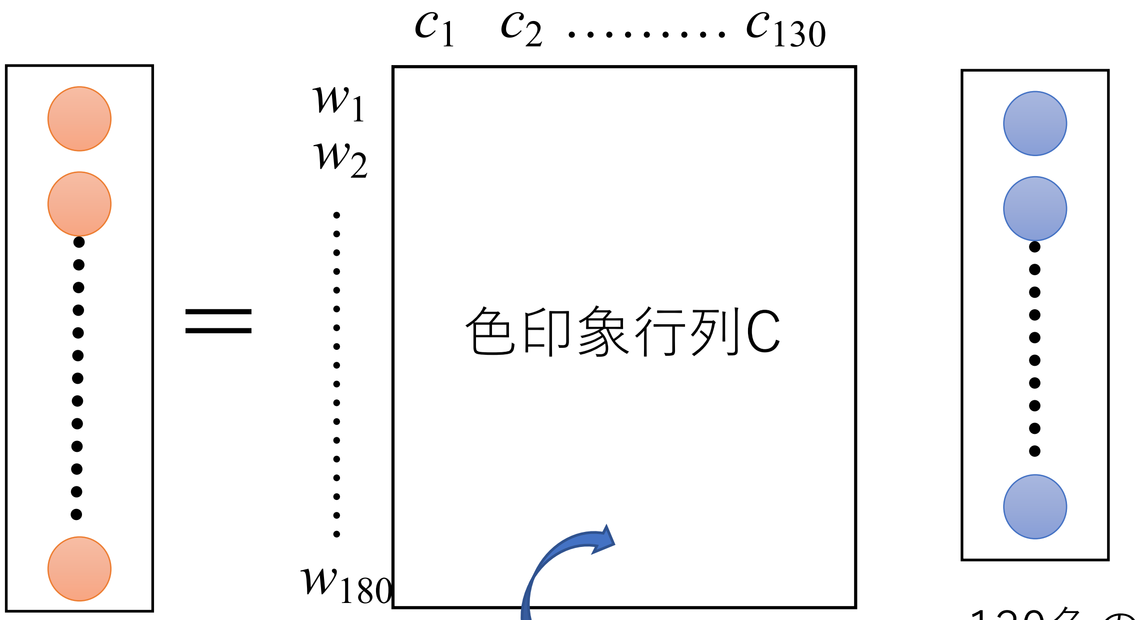
例1)暗号から平文へ



※これは秘密鍵暗号方式の例。暗号方式として大きく「秘密鍵暗号方式」と「公開鍵案号方式」があるためどう違うか調べてみよう

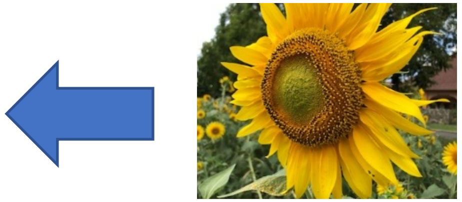
例2) 色情報から印象情報に変換

Media-lexicon Transformation Operator



180語の印象語の割合

130色の色の割合

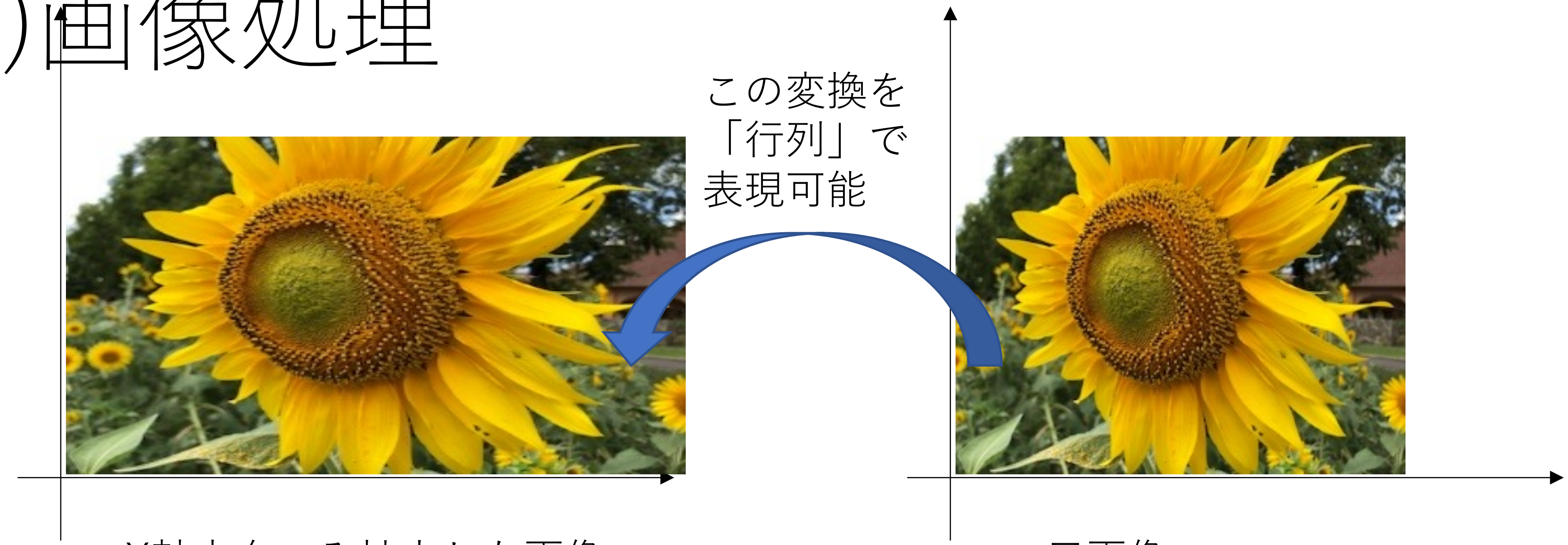


画像メディアコンテンツ

カラーイメージスケール：
130色の基本色と180の印象語の関連性

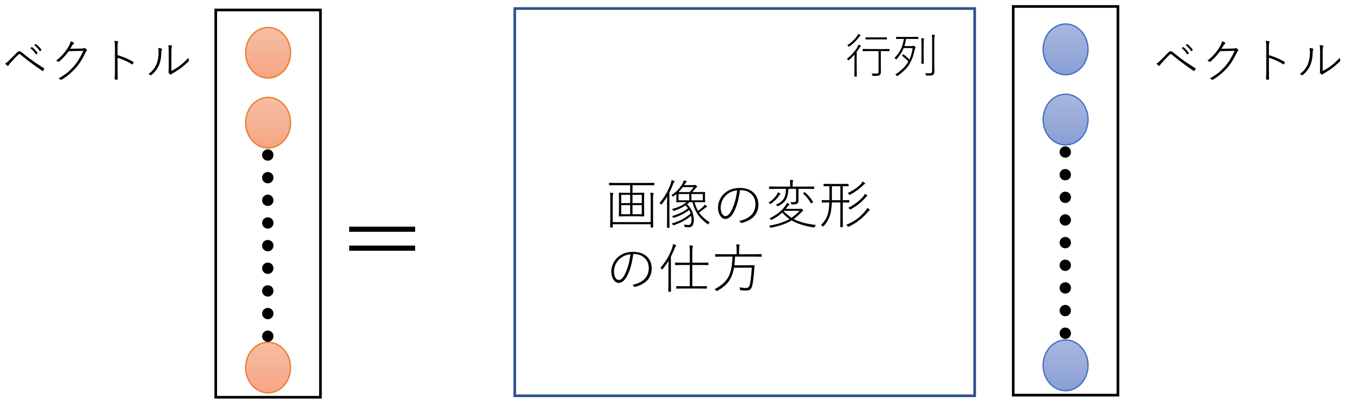
T. Kitagawa and Y. Kiyoki, "Fundamental framework for media data retrieval system using media lexico transformation operator," Information Modelling and Knowledge Bases, vol. 12, pp. 316–326, 2001.
北川高嗣，中西崇文，清木 康:静止画像メディアデータ を対象としたメタデータ自動抽出方式の実現とその意味 的画像検索への適用，情報処理学会論文誌，データベー ス，Vol.43, No.12, pp.38–51 (2002).
木村 侑斗, 岡田 龍太郎, 中西 崇文:画像メディアコンテンツの色彩情報に基づく楽曲メディアコンテンツ印象抽出方式,第12回データ工学と情報マネジメントに関するフォーラム, C1-3,.(2020).

例3)画像処理



X軸方向のみ拡大した画像
(ベクトルで表現可能)

元画像
(ベクトルで表現可能)

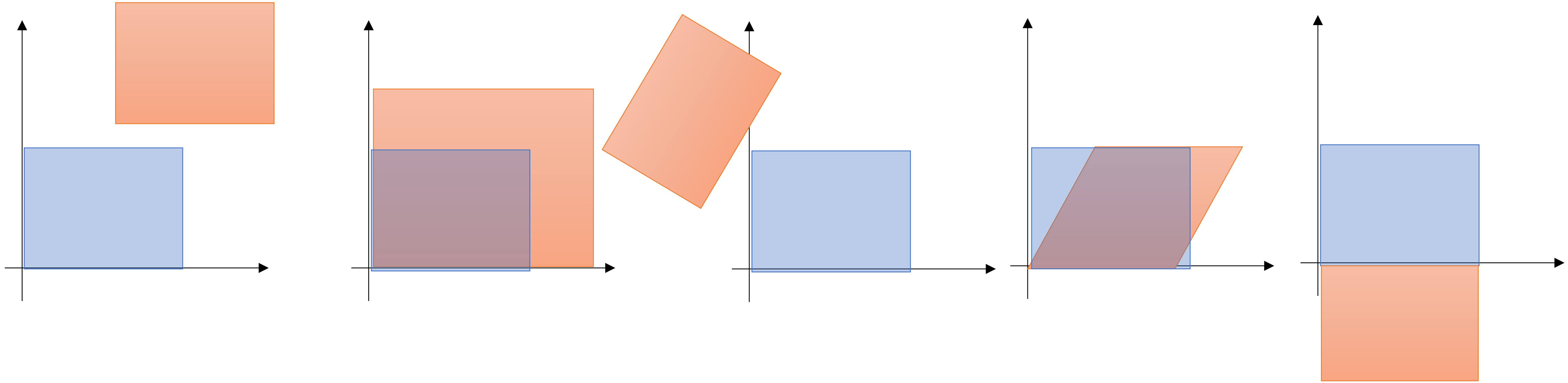


X軸方向のみ拡大した
画像データ

元画像データ

平面画像処理(Transformationの5つの形)

(1)平行移動(Translation)　(2)拡大・縮小(Scaling)　(3)回転(Rotation)　(4) せん断(Skew)　(5) 鏡映(Reflection)



アフィン変換

グループワーク

1. 各グループからリーダーを選出し、リーダーが決まったらSlackのチャンネル「#120_データと数理_2023」に書き込んでください。
2. 教科書p.20～p.27をグループで読み合わせて、線形代数の意味を考察した上で、行列とベクトルを使った変換(Transformation)で実現できそうな応用例を考えてみよう
   - 細かいことは学んでないので間違っていても構いません
   - ベクトル、行列が何次元になるのかは明記
   - このグループ課題の結果は単元「線形変換」で具体的に実装します

- 各グループ発表3分、質疑応答2分
- 発表資料はGoogle Classroomに別途提出してください。

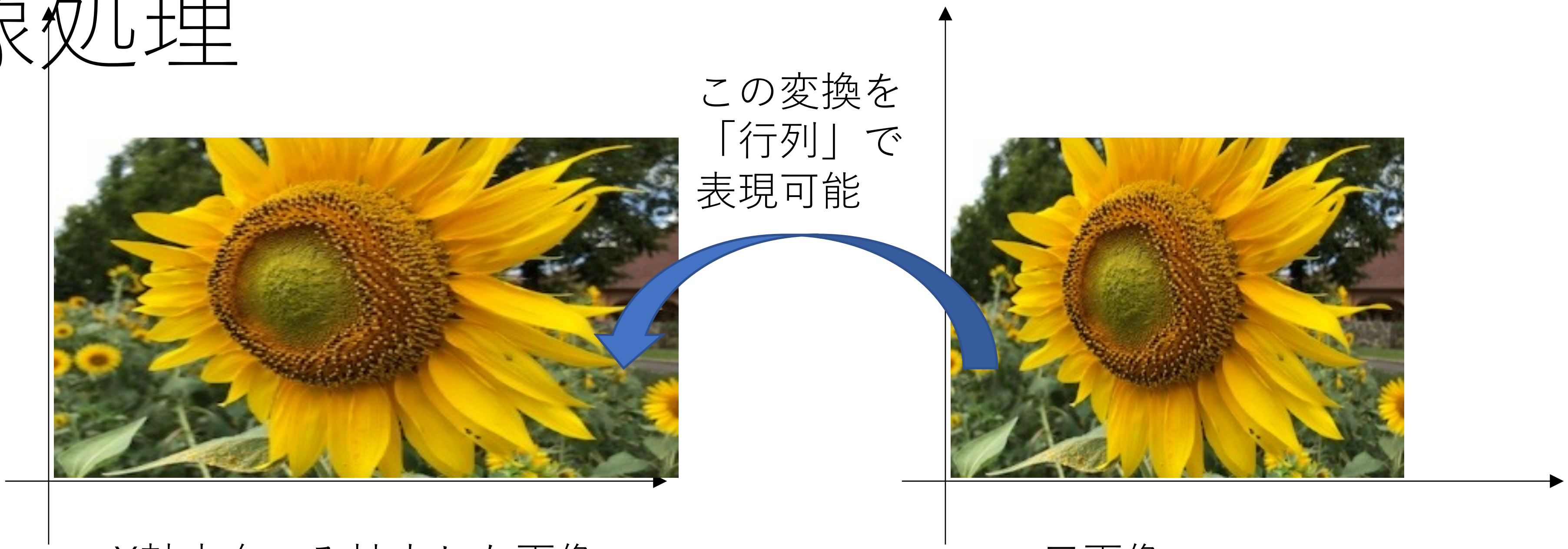
3限目に発表してもらいます！

個人課題

- YouTubeの「鈴木貫太郎,これから数IIIを学ぶ人に贈る。複素数って何だよ？iって何？」
  - https://youtu.be/5KcDC4sG8gU
- を視聴し、複素数について復習をした上で、複素数を平面画像処理にどのように使うのかを考えて要点をA4 1枚にまとめてください。
- [アドバンス課題]さらに、立体画像処理の場合は複素数を拡張した四元数を用いることがある。それについてA4 1枚にまとめて説明してください。
- Google Classroomで提出のこと
(締切はGoogle Classroom参照)

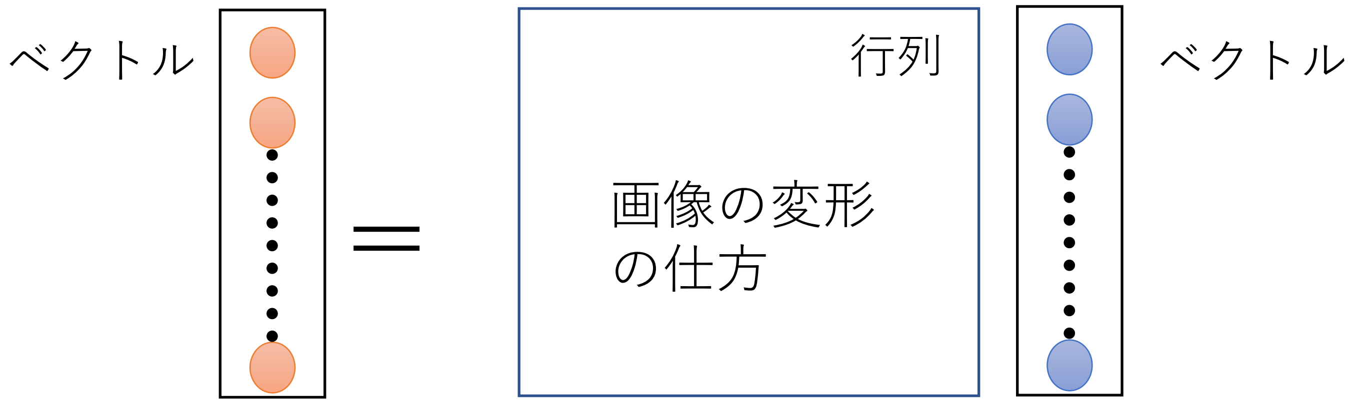
1章 複素平面で画像処理

画像処理



X軸方向のみ拡大した画像
(ベクトルで表現可能)

元画像
(ベクトルで表現可能)

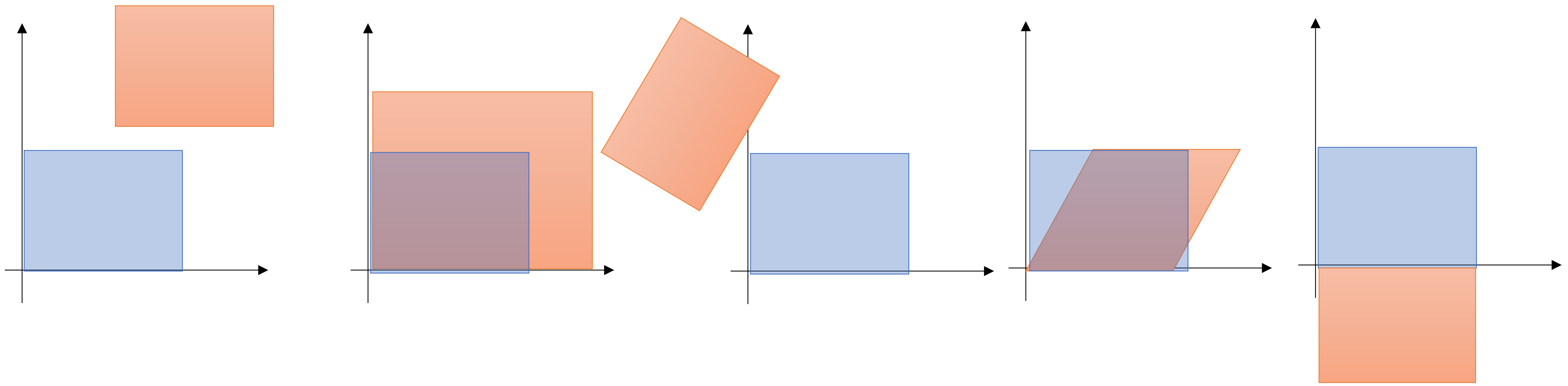


X軸方向のみ拡大した
画像データ

元画像データ

平面画像処理(Transformationの5つの形)

(1)平行移動(Translation) (2)拡大・縮小(Scaling) (3)回転(Rotation) (4) せん断(Skew) (5) 鏡映(Reflection)



アフィン変換

高校で習った数学が役に立つことを実感してもらおう

- 線形代数を使って画像処理ができると言ったが、
  実は高校で習った複素数を使って画像処理(平行移動、拡大・縮小、回転)ができるので、それで感覚をつかんでみよう。

2乗をして負になる数

- $x^2 = 3$ の解は？
- $x = \pm\sqrt{3}$

- では、
- $x^2 = -2$ の解は？

複素数

- 2乗すると負になる数を虚数(imaginary number)と呼ぶ
  - $i^2 = -1$
- $z = a + bi$の形をした数zを複素数(complex number)と呼ぶ
- zのaを実部(real part), zのbを虚部(imaginary part)
  - $a = Re(z)$,
  - $b = Im(z)$
- $z = a + bi$に対して$\bar{z} = a - bi$ をzの共役複素数(complex conjugate number)と呼ぶ

複素数の足し算

- $z_1 = a + bi,\ z_2 = c + di$
  - $z_1 + z_2 = (a + c) + (b + d)i$

- 例)
  - $(4 - 2i) + (3 + 4i) = (4 + 3) + (-2 + 4)i = 7 + 2i$

複素数の掛け算

- $z_1 = a + bi,\ z_2 = c + di$
  - $z_1 z_2 = (a + bi)(c + di)$
    $= ac + a(di) + (bi)c + (bi)(di) = ac + (ad + bc)i + (bd)i^2$
    $= (ac - bd) + (ad + bc)i$

- 例)
  - $(4 - 2i)(3 + 4i) = 12 + 16i - 6i - 8i^2 = 12 + 8 + (16 - 6)i = 20 + 10i$

複素数の割り算

- $\frac{6+2i}{1+i}$
  - $\frac{6+2i}{1+i} = \frac{(6+2i)(1-i)}{(1+i)(1-i)} = \frac{2(3+i)(1-i)}{2} = (3+i)(1-i) = 3-3i+i-i^2 = 4-2i$

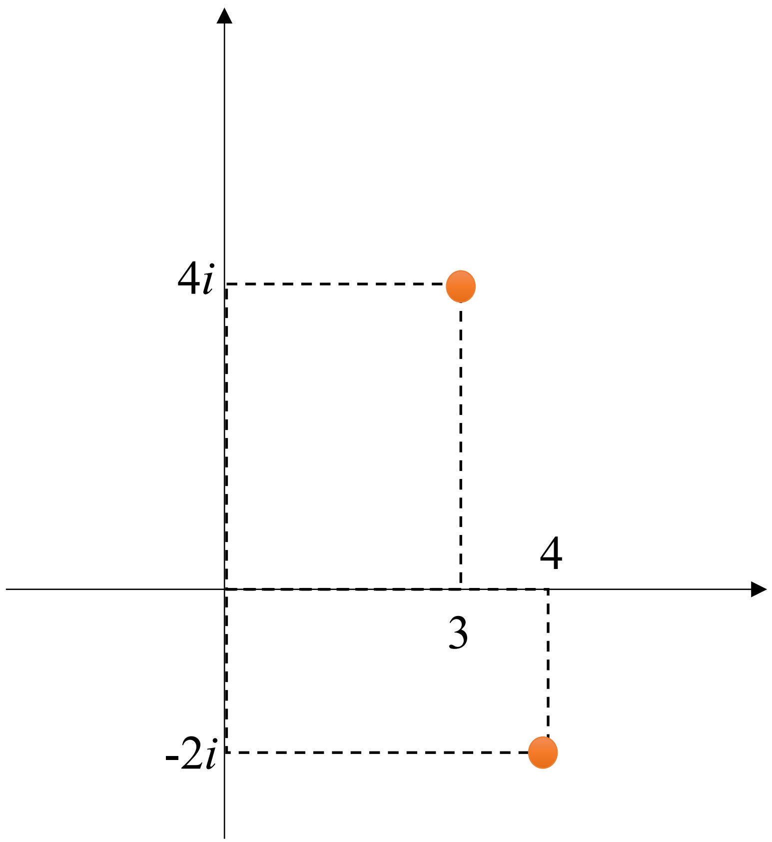
Pythonで計算してみよう

- $(4-2i)+(3+4i)$
- $(4-2i)-(3+4i)$
- $(4-2i)(3+4i)$
- $\frac{4-2i}{3+4i}$

複素平面

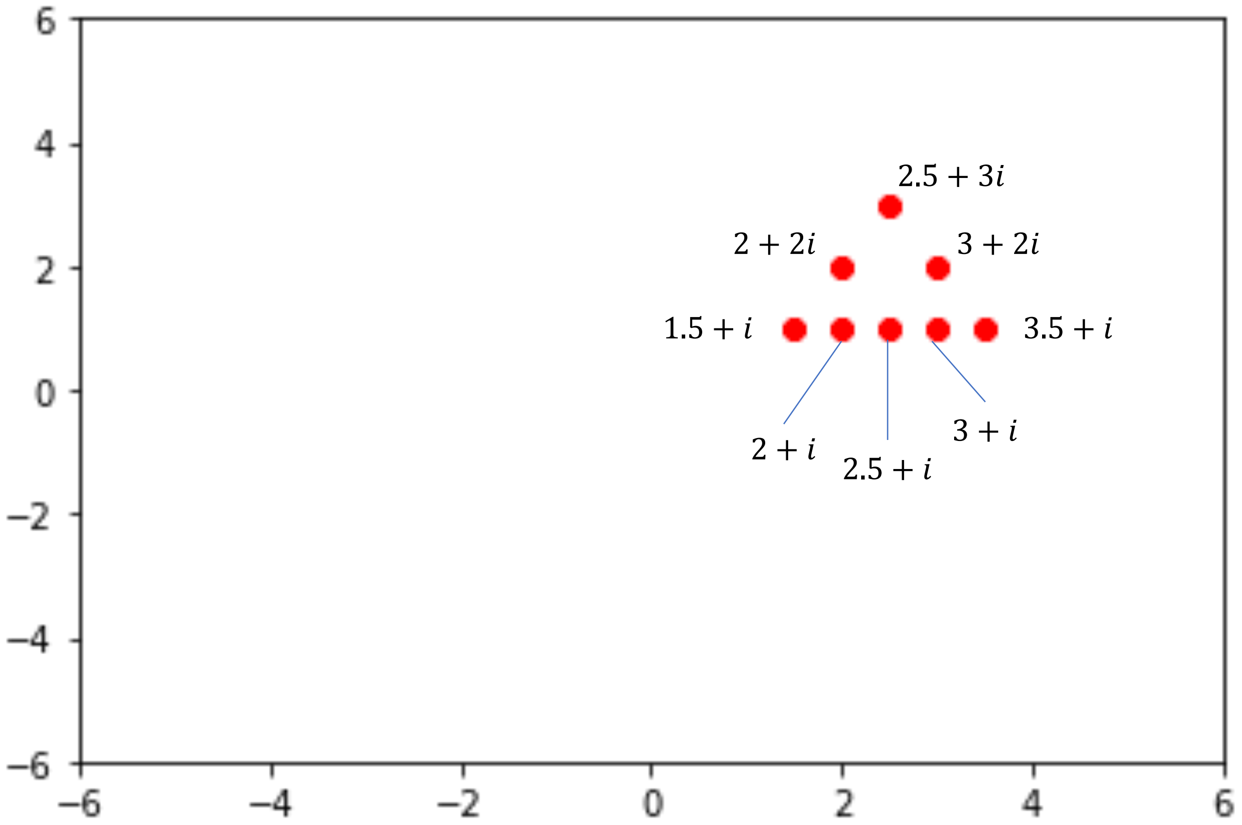
- 複素数を座標で表現する
- 実部は横軸、虚部は縦軸
  - 実部を表す横軸→実軸(real axis)
  - 虚部を表す縦軸→虚軸(imaginary axis)

- 例)
  - $z_1 = 4 - 2i$
  - $z_2 = 3 + 4i$



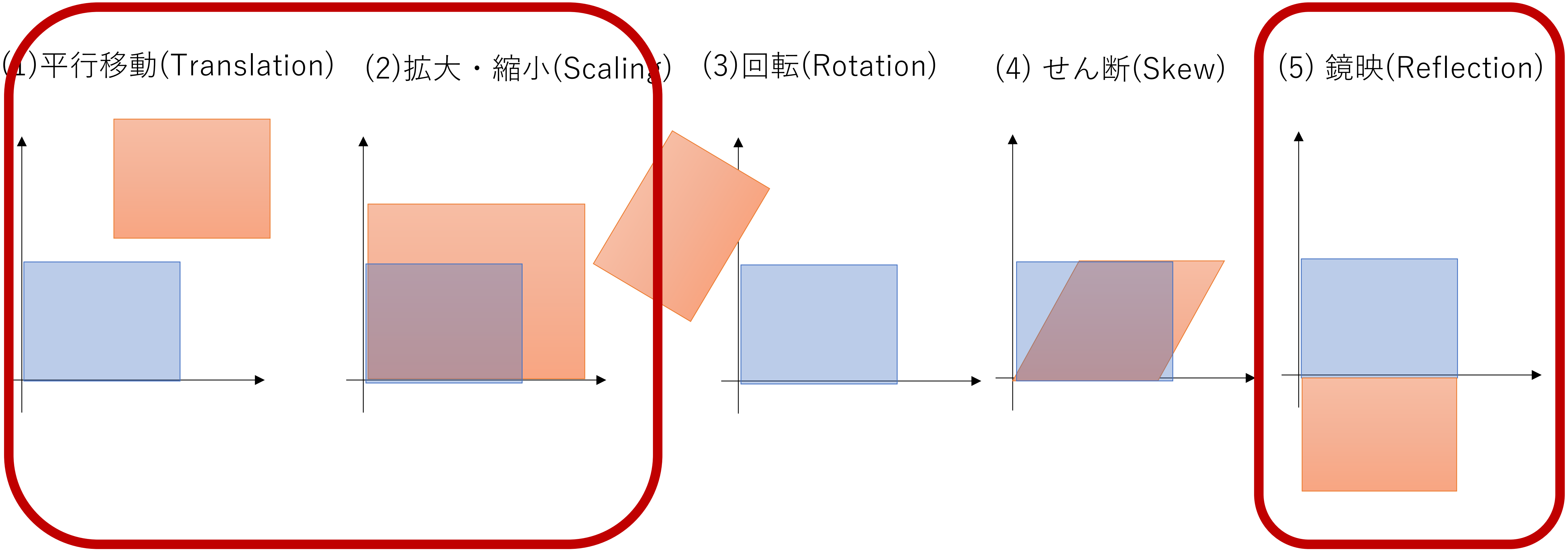
Pythonで複素平面を表現する

- 8個の複素数を複素平面上にプロットする
  - $2.5+3i, 2+2i, 3+2i, 1.5+i, 2+i, 2.5+i, 3+i, 3.5+i$



小さな二等辺三角形にみえる？

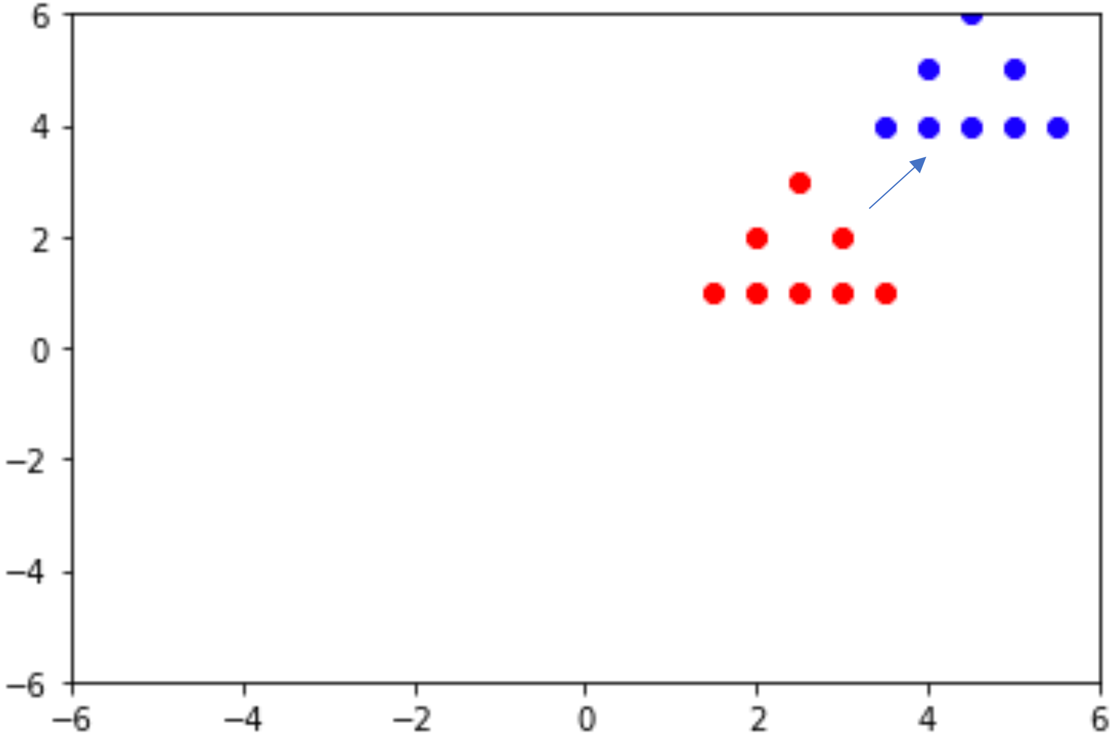
平面画像処理(Transformationの5つの形)



複素平面上でなんとか実現できる

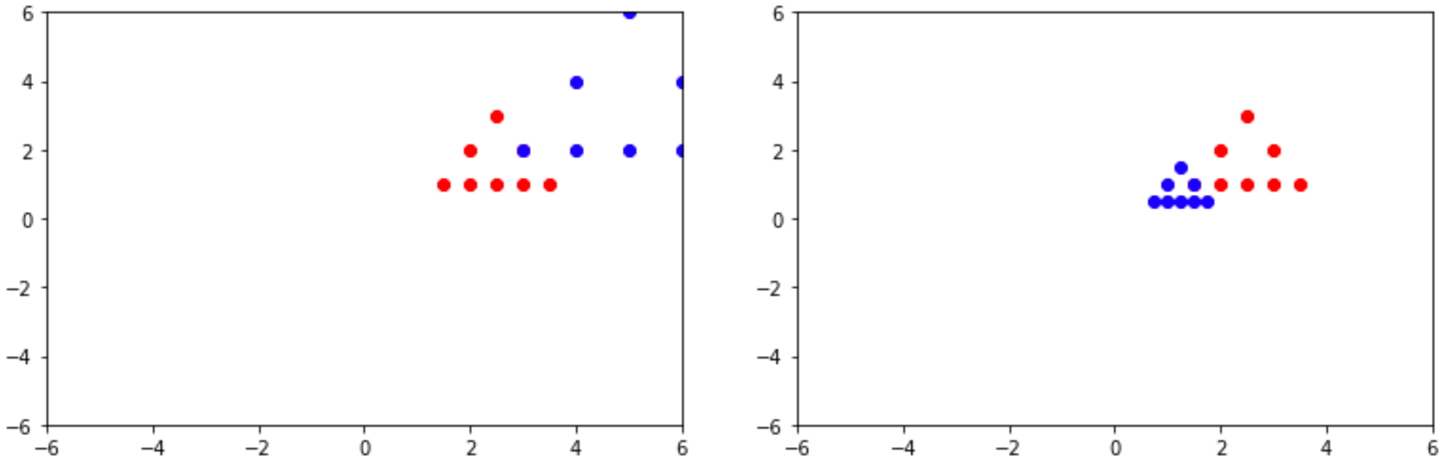
平行移動

- 複素数に複素数を足し算
- 例
  - 横方向に2、縦方向に3を移動したいとき
  - $2+3i$を足せばよい



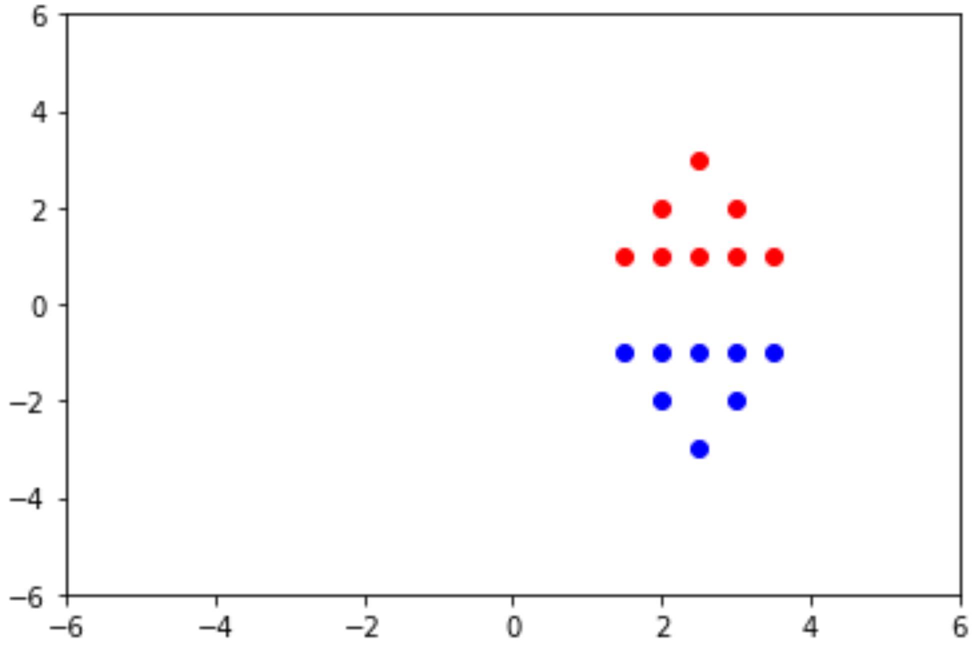
拡大・縮小

- 複素数に実数を掛け算
- 例
  - 2倍拡大したい場合
    - 2を掛ければよい
  - 1/2に縮小場合
    - 1/2を掛ければよい

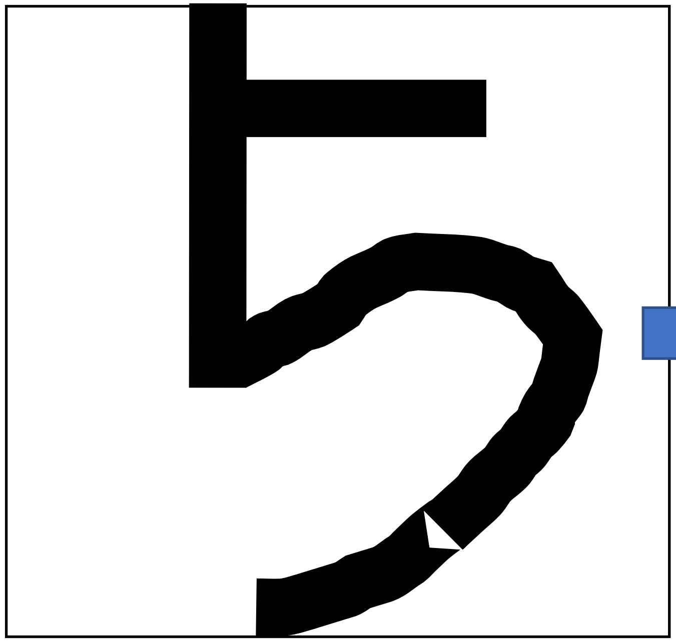


鏡映(Reflection)

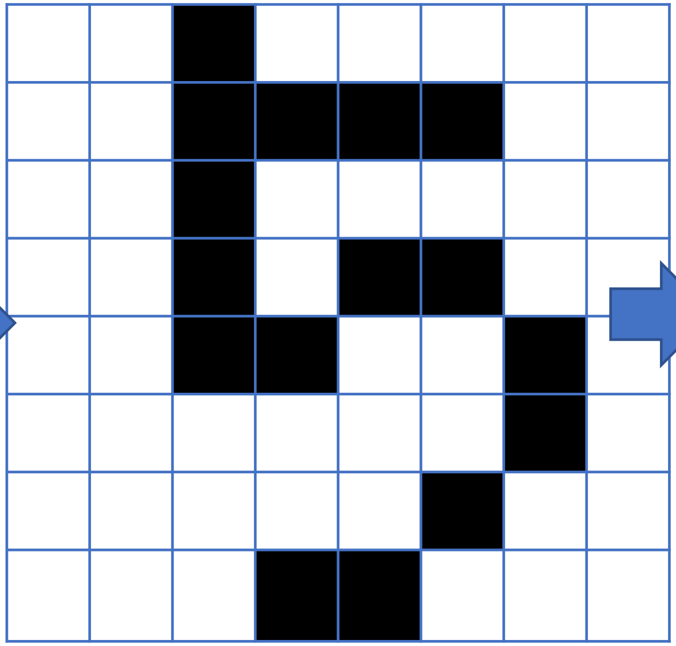
- 共役複素数(complex conjugate number)にする



白黒画像のディジタル化



元画像(白黒)



標本化

0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0

量子化

00100000001
11100001000
00001011000
01100100000
00100000010
000011000

符号化

画像ファイルを読んで白黒(2値化)した上で複素座標にする関数

画像をそのまま読み込むと

左上が(0,0)
img[0][0]

y →

x ↓

0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0

複素平面上の座標解釈 $x + yi$

y ↑

0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	0	0	0	0	0
0	0	1	0	1	1	0	0
0	0	1	1	0	0	1	0
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	1	1	0	0	0

0

左下が(0,0)

x →

```
img = imgfile2bin('tmp.png')
```

imgfile2bin
正方形の画像ファイルを200×200ピクセルにした上で、
複素平面上でプロットする(画像において黒である)座標を取得

```python
import numpy as np
from PIL import Image

def imgfile2bin(filename):
    threshold = 100
    img = np.array(Image.open(filename).convert('L').resize((200, 200)))
    img_bool = img > threshold
    c_img = np.array([])
    for i in range(img_bool.shape[0]):
        for j in range(img_bool.shape[1]):
            if img_bool[i,j]==False:
                c_img=np.append(c_img,complex(j,(img_bool.shape[0]-1)-i))
    return c_img
```

[34]
```python
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

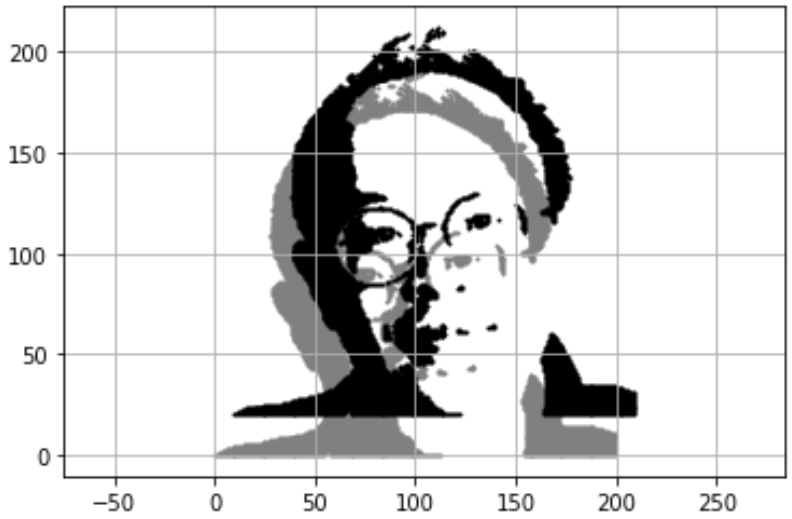
[35]
```python
# nakanishi.pngを置いた場所を指定。この例は/マイドライブ/Colab Notebooks/DataMathematicsI/nakanishi.pngに置いた場合
img = imgfile2bin('/content/drive/MyDrive/Colab Notebooks/DataMathematicsI/Lec01/nakanishi.png')
```

[36]
```python
img
```

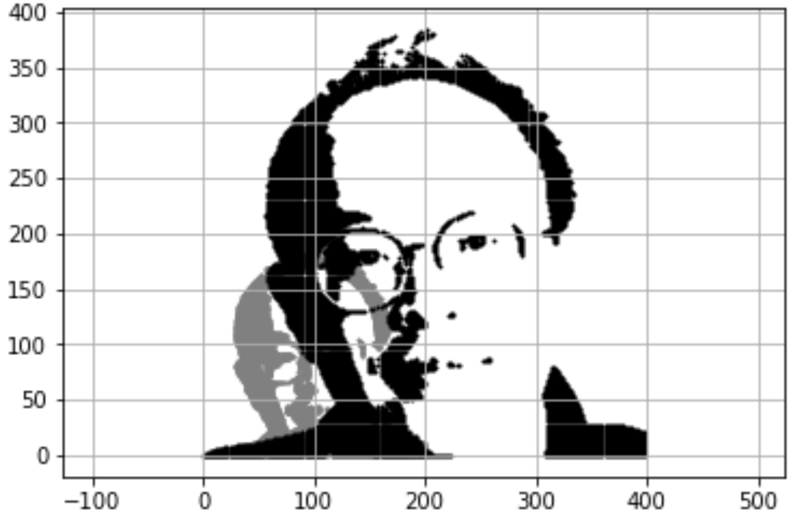
```
array([101.+192.j,  98.+191.j, 100.+191.j, ..., 197.  +0.j, 198.  +0.j,
       199.  +0.j])
```

これじゃつまらないので
実際の画像で複素平面を体験

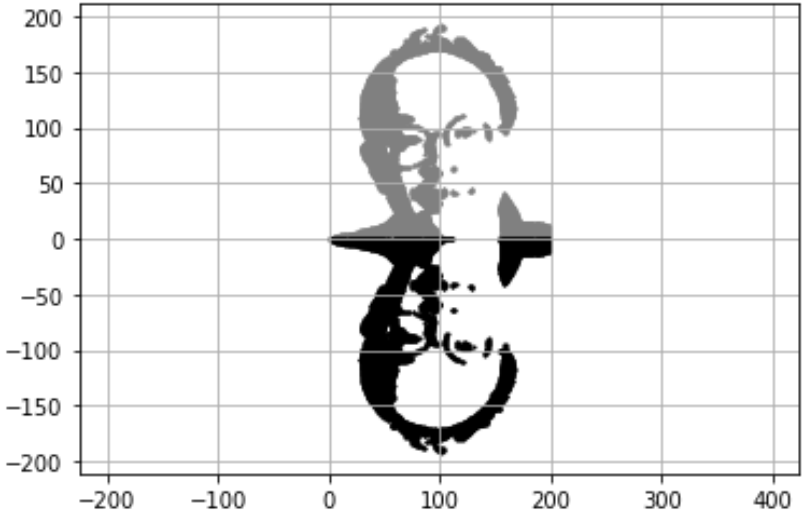
平行移動



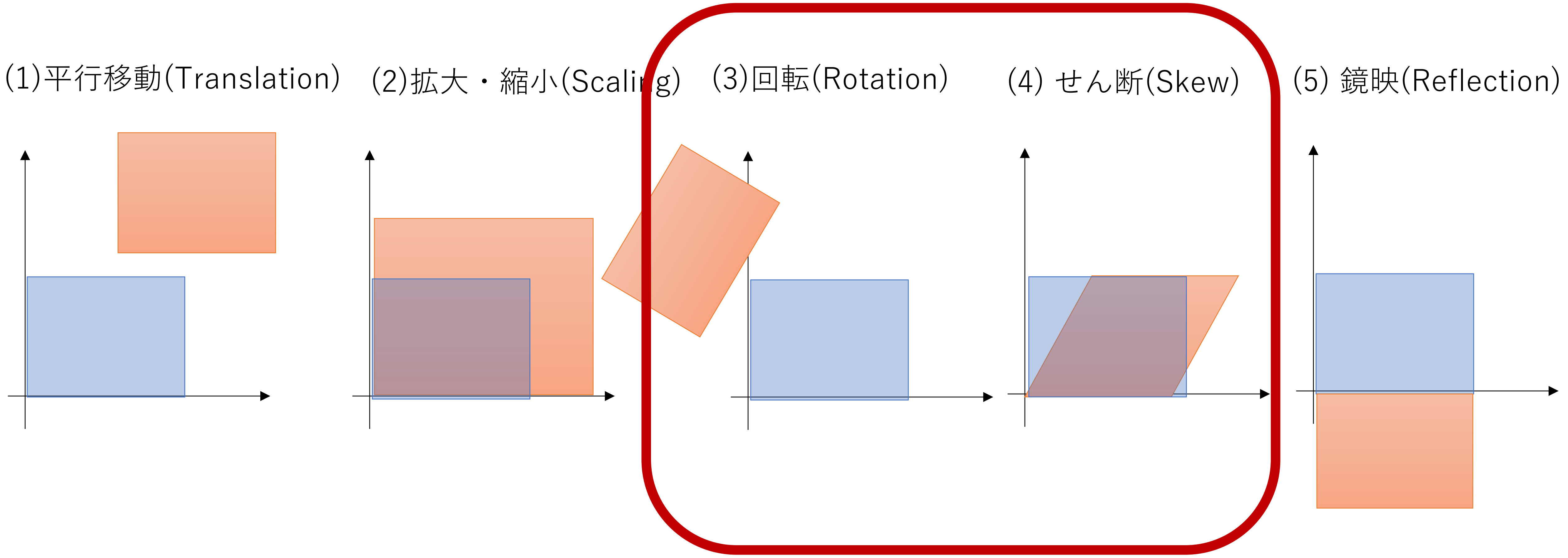
2倍に拡大



鏡映(共役複素数にする)



平面画像処理(Transformationの5つの形)

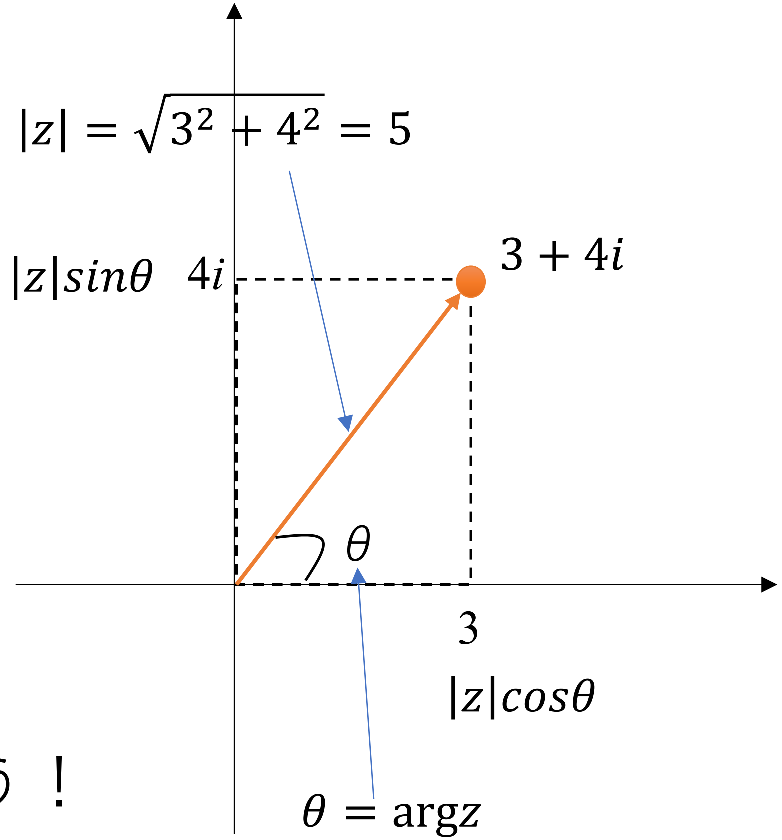


極座標上で実現できる

極座標

- $z = a + bi$を極形式
$$z = r\,(cos\theta + isin\theta)$$
(ただし、$r = |z| = \sqrt{a^2 + b^2}, \theta = \mathrm{arg}z$)
と表現することができる

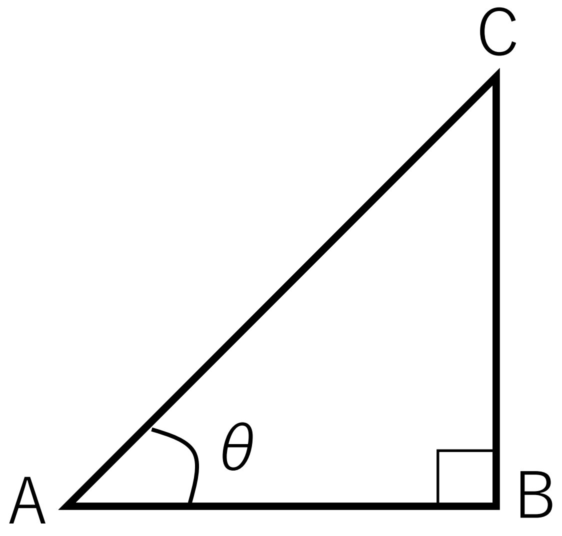
- 上記の複素数の表現の(r, θ)を極座標
(polar coordination)と呼ぶ



極座標で表現できれば回転が可能になるでしょう！

(忘れた人用)三角関数を復習しよう

- $sin\theta = \frac{BC}{AC}$
- $cos\theta = \frac{AB}{AC}$
- $tan\theta = \frac{BC}{AB}$



度数法(度:°)	0°	30°	45°	60°	90°	120°	135°	150°	180°
弧度法(ラジアン:rad)	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	$\frac{2}{3}\pi$	$\frac{3}{4}\pi$	$\frac{5}{6}\pi$	π
$\sin\theta$	0	$\frac{1}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{\sqrt{3}}{2}$	1	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	0
$\cos\theta$	1	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{1}{\sqrt{2}}$	$-\frac{\sqrt{3}}{2}$	-1
$\tan\theta$	0	$\frac{1}{\sqrt{3}}$	1	$\sqrt{3}$	−	$-\sqrt{3}$	-1	$-\frac{1}{\sqrt{3}}$	0

Pythonで複素数から極座標に変換

- $z = 3 + 4j$ → $z = r\,(cos\theta + isin\theta)$
- → $(r, \theta) = $ (5.0, 0.9272952180016122)

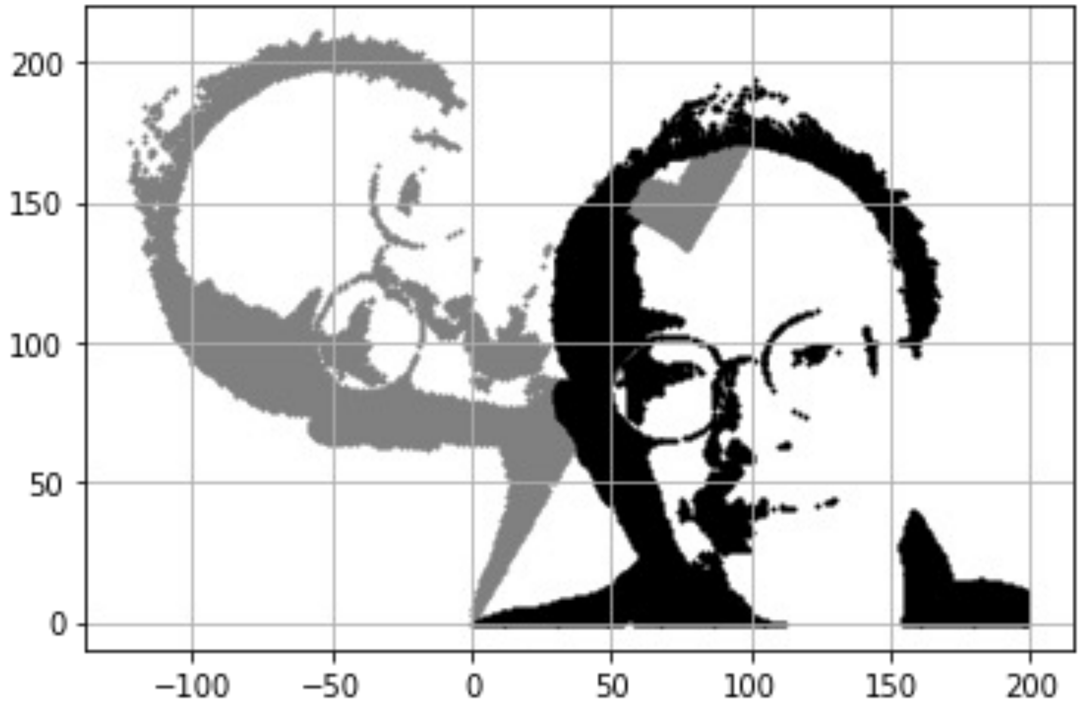
$r = |z| = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$

$\theta = \mathrm{arg} z$
弧度法(ラジアン)であることを注意

cmathのpolar関数を用いれば複素数から極座標に変換できる
cmathのrect関数を用いれば極座標から複素数に変換ができる

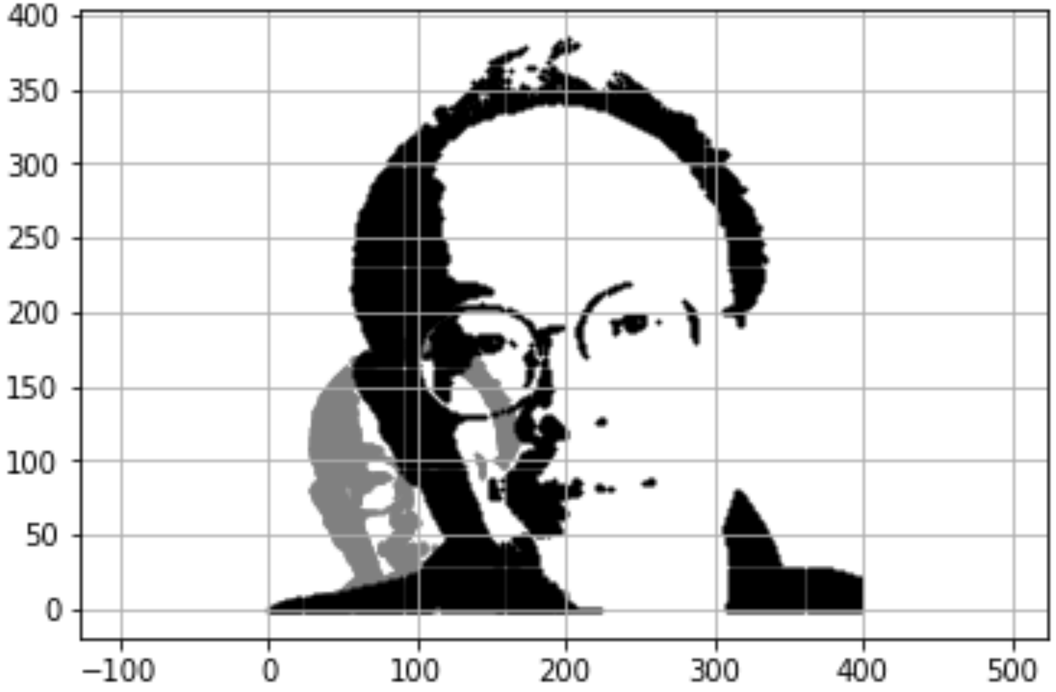
回転

- 極座標表現の θ を度数法に直し、角度を足す



拡大・縮小

- 極座標表現のrに実数を掛ける



オイラーの公式

- $e^{i\theta} = cos\theta + isin\theta$という式がオイラーの公式と呼ばれている
- これを使えば、
  - $z = a + bi$の極形式
    $z = r\ (cos\theta + isin\theta) = re^{i\theta}$と表現可能

- ちなみに
  - $e^{-i\theta} = \cos(-\theta) + isin(-\theta) = cos\theta - isin\theta$となり、 $e^{-i\theta}$は$e^{i\theta}$の共役になっている。これを用いると次の式が導き出される
    - $cos\theta = \frac{e^{i\theta}+e^{-i\theta}}{2}$
    - $sin\theta = \frac{e^{i\theta}-e^{-i\theta}}{2i}$

https://wiis.info/blog/beauty-of-eulers-formula/

こんな時期だから見てみよう

- 博士の愛した数式(小川洋子)
  - $e^{i\pi} + 1 = 0$
    - $e^{i\pi} = cos\pi + isin\pi = -1$

- 中学の知識でオイラーの公式がわかる(鈴木貫太郎)
  - 実はもともとYouTubeの動画コンテンツ
    - 刷新されて公開されている(全20動画)
    - https://www.youtube.com/playlist?list=PLFrlW-Y5LqlZb-YrRcr2ezhhfG4nQ7KSF

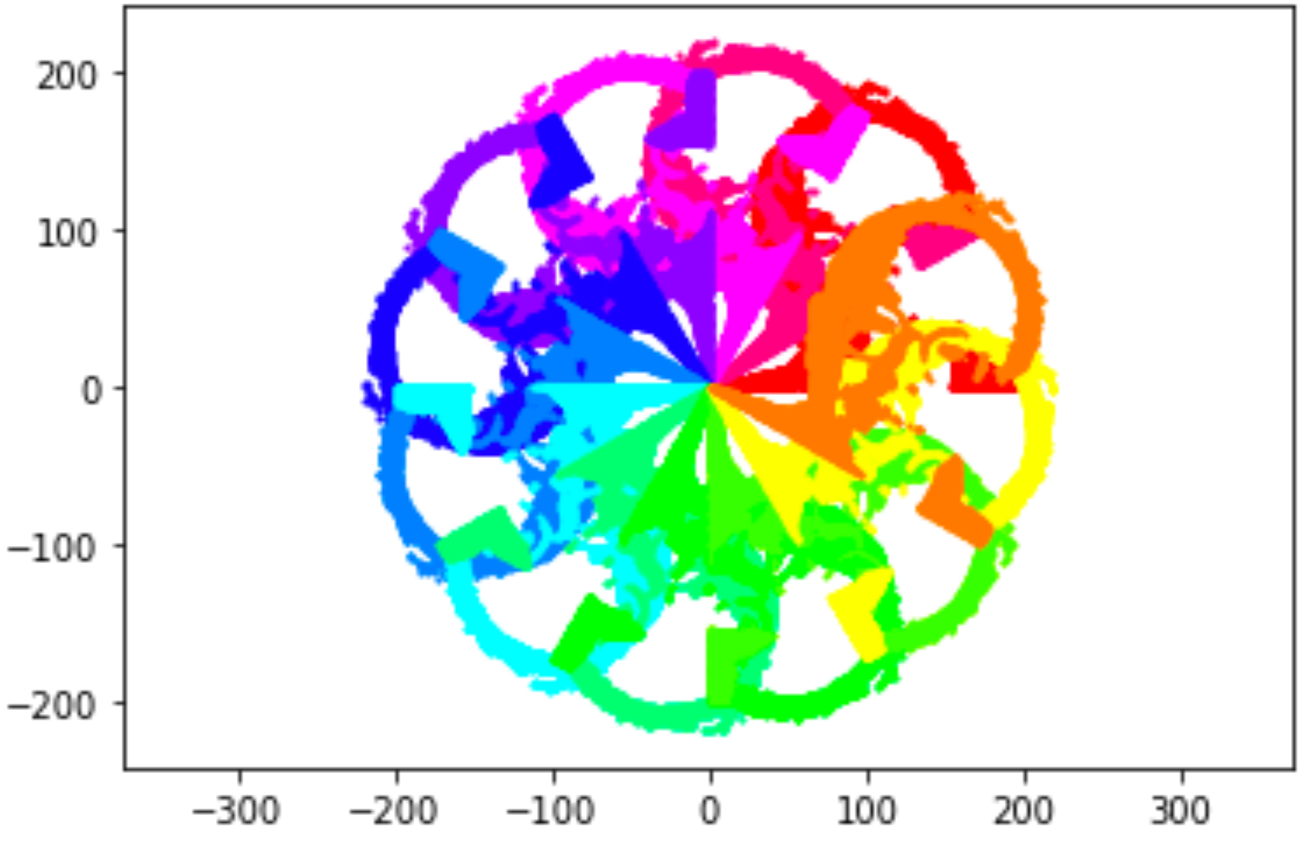
個人課題

1. 次のプログラムを完成させよ
    1. 自分の好きな正方形の写真を読み込み、複素平面上に表示するプログラムを作成
    2. 1の画像を極座標形式に変換し、2倍拡大した上で120度回転させた画像を表示
    3. 次のページのように平行移動、拡大・縮小、回転をさせることで自由な作品を作成
        - これらの課題は全て複素数、複素平面、極座標を用いて行うこと
        - 適宜プログラムにコメントをつけること
        - ipynb形式でGoogle Classroomで提出のこと
(締切はGoogle Classroom参照)
    - [アドバンス課題]立体画像を対象として四元数を用いて平行移動、拡大・縮小、回転等を行なってみよう

2. 教科書p.28〜p.61を読んで理解したことをA4 1枚でまとめよ
その際理解のため「予備校のノリで学ぶ「大学の数学・物理」【大学数学】線形代数入門①(概観&ベクトル)【線形代数】」を視聴することもおすすめする
https://youtu.be/svm8hlhF8PA
    - docx形式およびpdf形式でGoogle Classroomで提出のこと

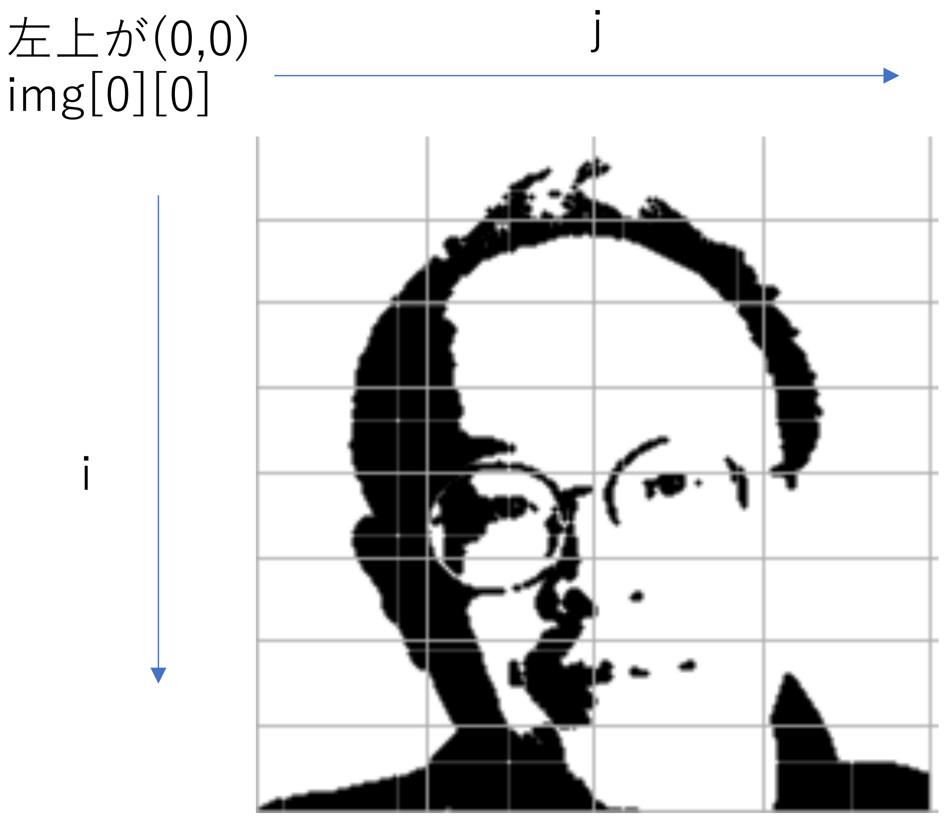
参考)

- 極座標系に直して角度を足し合わせながら色を変えていく



画像ファイルを読んで白黒(2値化)した上で複素座標にする関数

画像をそのまま読み込むと



Matplotlib上の座標解釈