画像の表現，二値化処理

目的

- 二値化プログラムを実装する

- 疑似階調表現による二値化プログラムを実装する

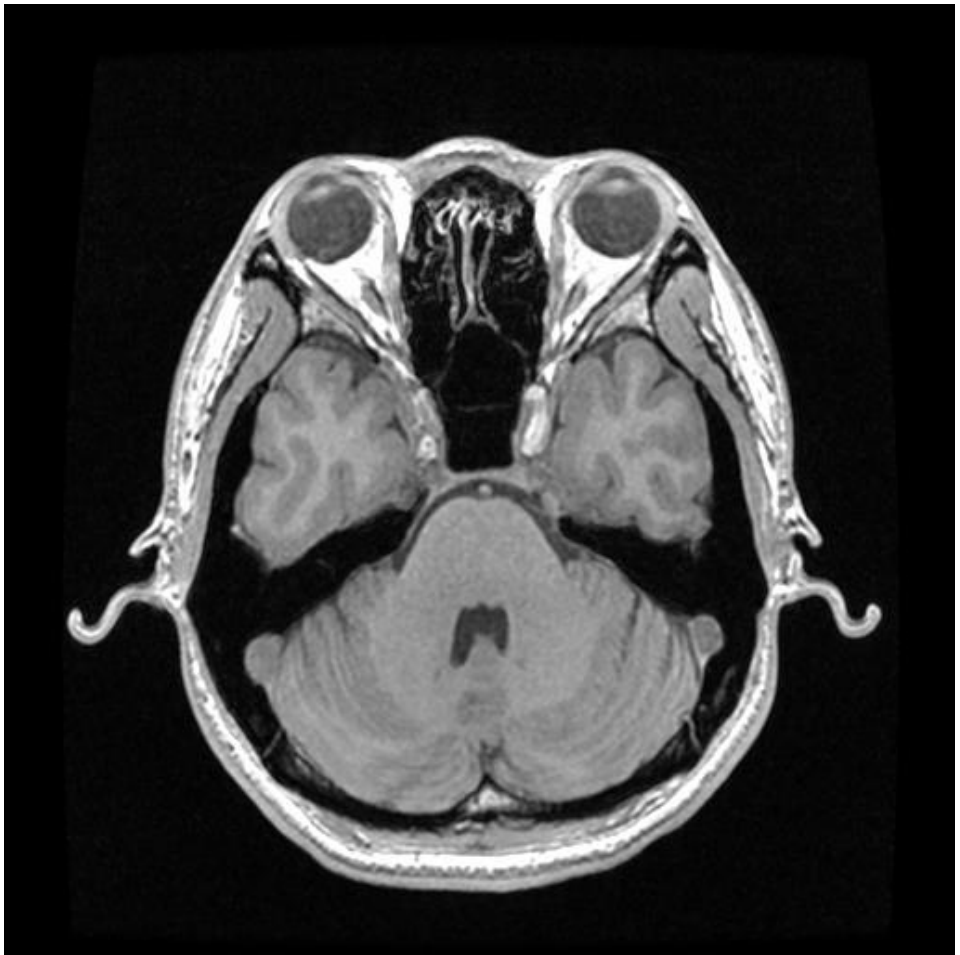
画像情報(階調値，輝度)

- 本講義で扱う画像は8ビットのグレースケールデータを階調値に持つ

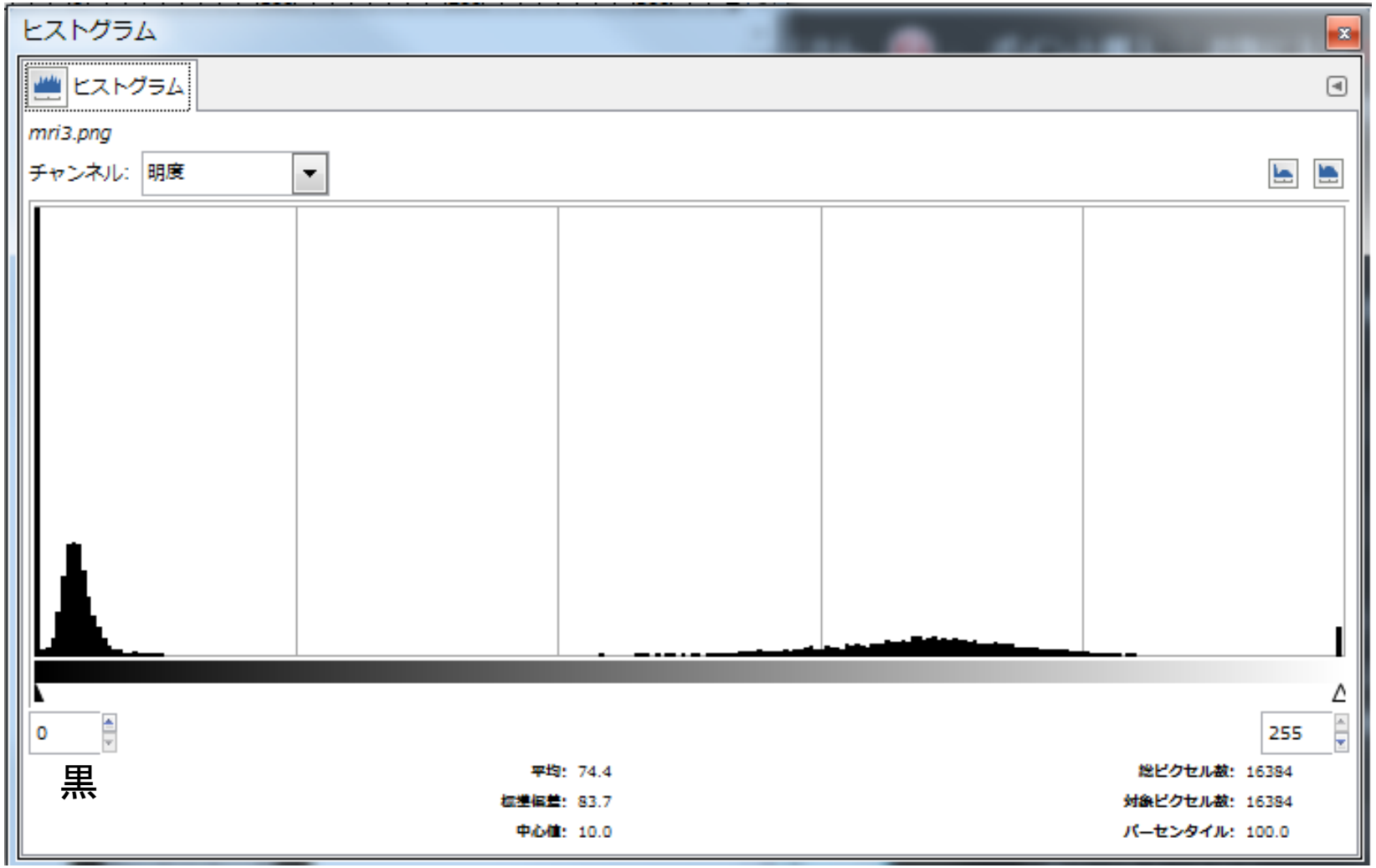
- この階調値を操作することによって，“画像”を“処理“することが出来る

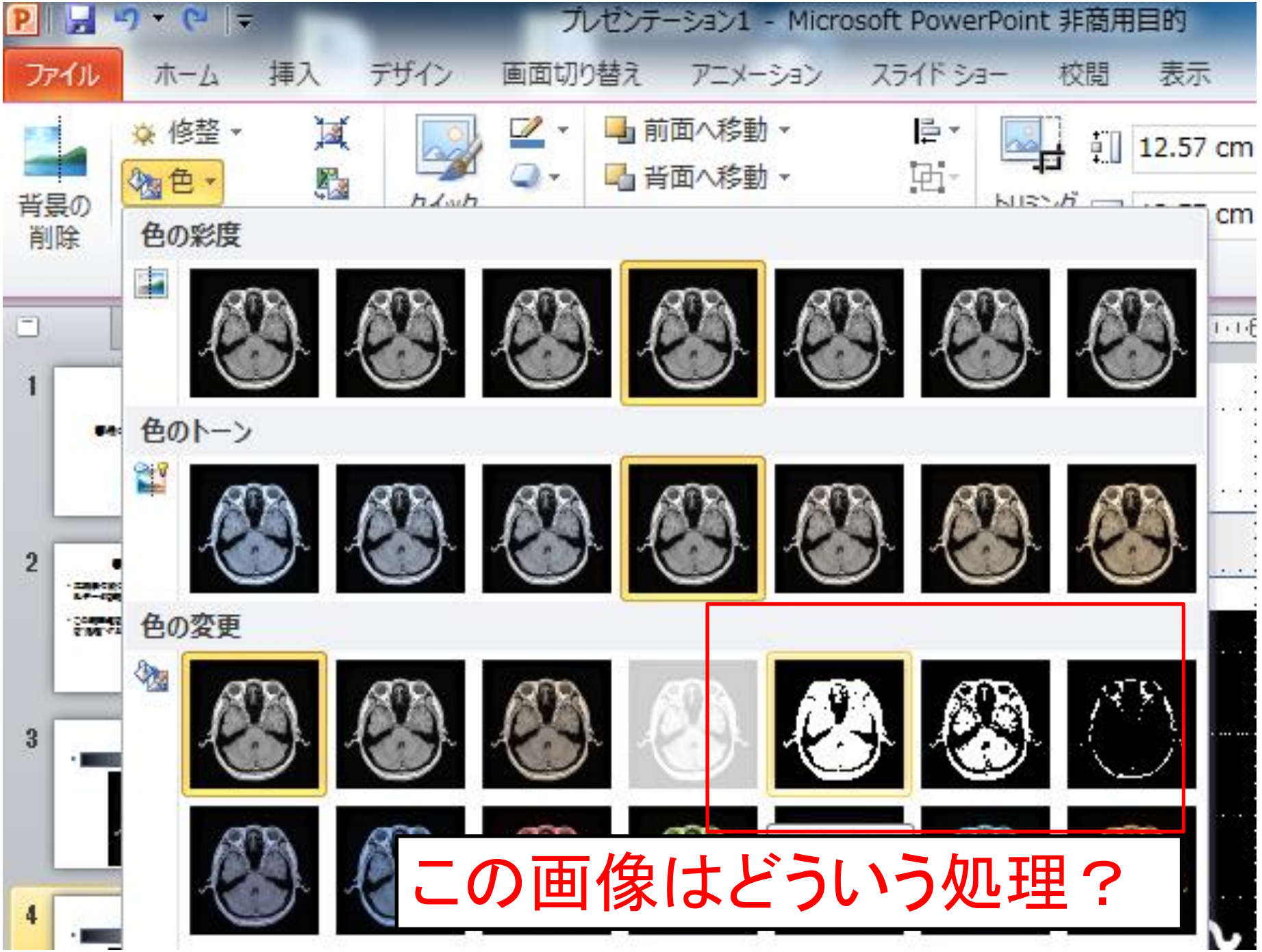
サンプル画像

0  255

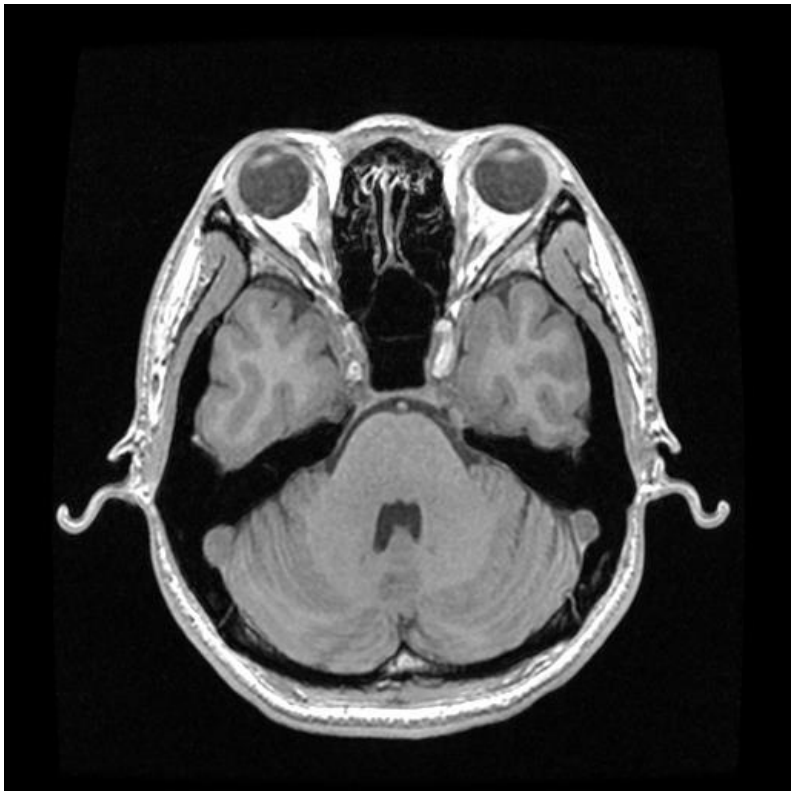


ヒストグラム（輝度分布）


ヒストグラム
ヒストグラム
mri3.png
チャンネル:
明度
0
255
黒
白
平均: 74.4
標準偏差: 83.7
中心値: 10.0
総ピクセル数: 16384
対象ピクセル数: 16384
パーセンタイル: 100.0


プレゼンテーション1 - Microsoft PowerPoint 非商用目的
ファイル
ホーム
挿入
デザイン
画面切り替え
アニメーション
スライド ショー
校閲
表示
背景の削除
修整
色
前面へ移動
背面へ移動
12.57 cm
色の彩度
色のトーン
色の変更
この画像はどういう処理？

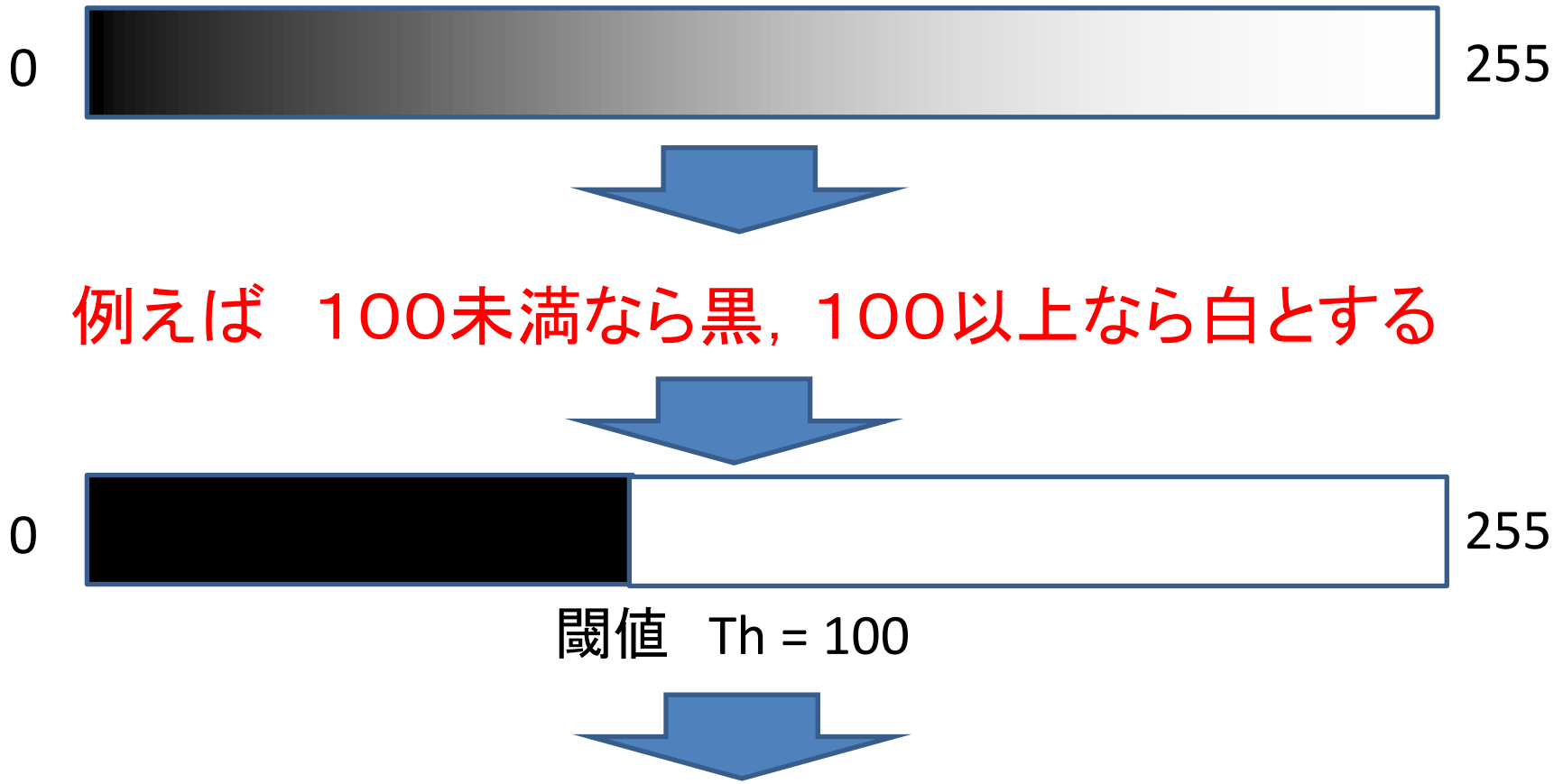
Before



二値化処理

After

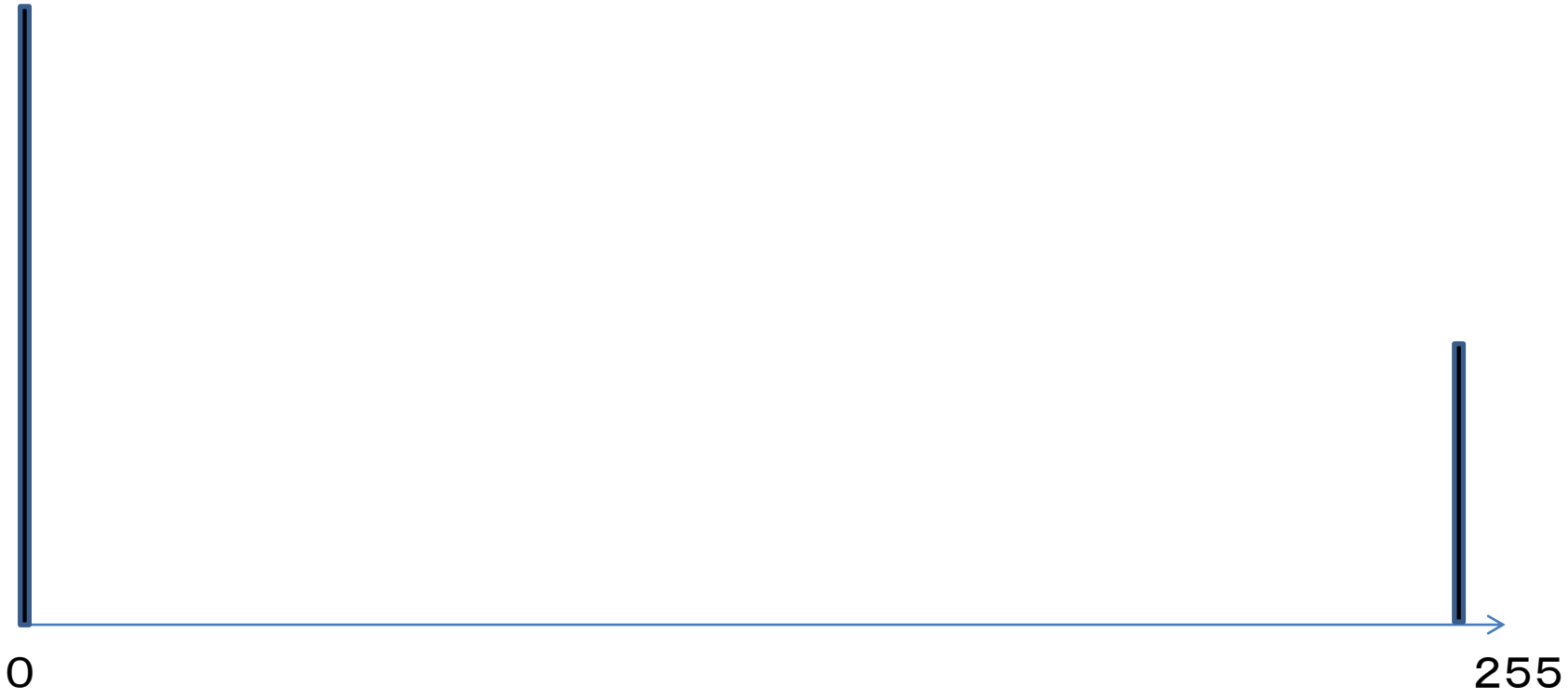
最も簡単な二値化処理



閾値は自由に設定できるようにする

二値化後のヒストグラム（イメージ）

- 黒か白か



課題

- 上述のアルゴリズムを実装し，画像処理前後の画像を比較せよ

- 閾値を変更して試して画像がどのようになったかを比較検討せよ

プログラムの雛型

ヘッダーファイルのインクルード

```
#include <stdio.h>
#include “pgmlib.h”
```

プロトタイプ宣言

```
void binarize(int n);
```

main関数

```
char file[100] = “”;
load_image(0,file);
binarize(0);  //0番目の画像に対して二値化
save_imave(0 ,file);
```

```
void binarize(int n)
{
	int thvalue;
	int x,y;
	do{
		printf(“2値化の閾値(0-255)：”);
		scanf(“%d”, &thvalue);
	}while(thvalue < 0 || thvalue > 255);

	//ここで二値化
	//image[n][x][y]の階調値が閾値以下か？
}
```

目的

- 二値化プログラムを実装する

- 疑似階調表現による二値化プログラムを実装する

二値化

- 二値化処理によって画素の階調値は0または255となる
- 中間的な階調を表現できないが，疑似階調表現により元の階調画像の印象を保つような二値化が可能である
- ドットパターン，ディザ法，誤差拡散法などがある

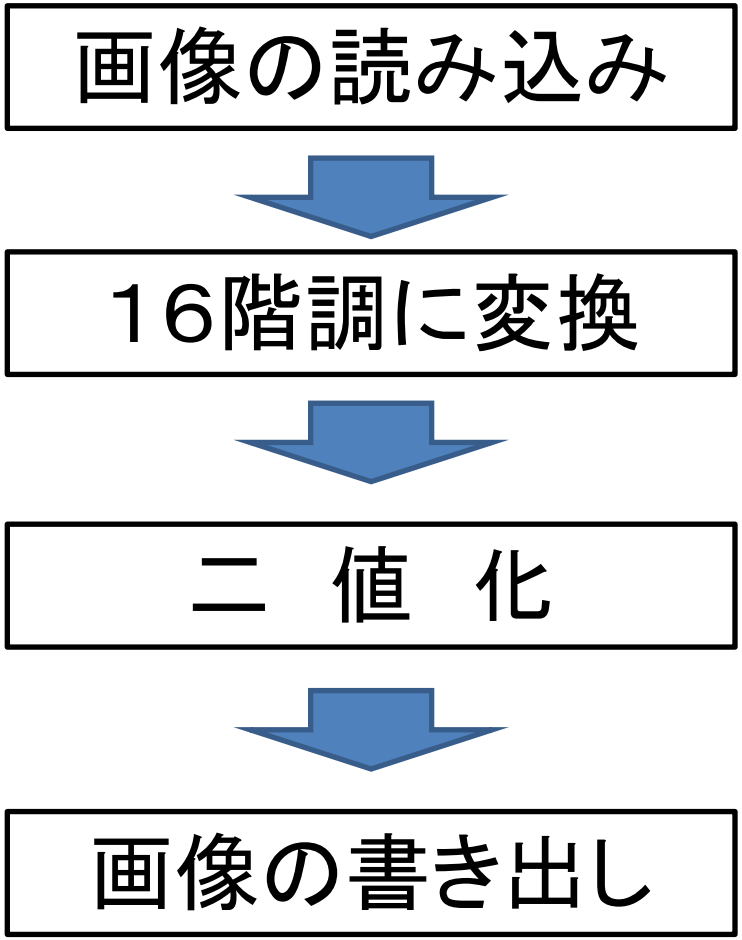
組織的ディザ法

- 図に示すようなディザマトリクスと呼ばれる
  n×nの数字（0～n^2-1)を配置した表を用いる

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

図　Bayer型のディザマトリクス

具体的な処理 (n=4の場合)



サンプル

0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0

１６階調に変換

0	15	0	15	0	15	0	15
15	0	15	0	15	0	15	0
0	15	0	15	0	15	0	15
15	0	15	0	15	0	15	0
0	15	0	15	0	15	0	15
15	0	15	0	15	0	15	0
0	15	0	15	0	15	0	15
15	0	15	0	15	0	15	0

ディザマトリクスと比較

0	15	0	15
15	0	15	0
0	15	0	15
15	0	15	0

ディザマトリクス

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

閾値を用いて二値化

0	255	0	255
255	0	255	0
0	255	0	255
255	0	255	0

ディザマトリクス

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

ディザマトリクスと比較（以下同様）

ディザマトリクス

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

0	15	0	15
15	0	15	0
0	15	0	15
15	0	15	0

二値化後の画素

0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0
0	255	0	255	0	255	0	255
255	0	255	0	255	0	255	0

```
#include <stdio.h>
#include "pgmlib.h"
int dmatrix[4][4] = {
    {  0,  8,  2, 10 },
    { 12,  4, 14,  6 },
    {  3, 11,  1,  9 },
    { 15,  7, 13,  5 } };
void dither(int n);
int main(void)
{
```

```
    load_image(0,"");
    dither(0);
    save_image(0,"");
    return 0;
}
```

```
void dither(int n)
{
        int x,y,i,j,xp,yp;
        int bxsize,bysize;
        if ( ( height[n] □ □ )!=0 || (width[n] □ □ )!=0 ){
          printf("サイズは4の倍数にして下さい¥n");
          exit(1);
        }
        //16階調に変換する
      for(y=0;y<height[n];y++)
                for(x=0;x<width[n];x++)
                    image[n][x][y] = image[n][x][y] / □;
```

```
bxsize = width[n] / 4;
bysize = height[n] / 4;
for(y=0;y<bysize;y++){
    for(x=0;x<bxsize;x++){
        for(i=0;i<4;i++){
            for(j=0;j<4;j++){
                //考えて下さい


            }
        }
    }
}
```