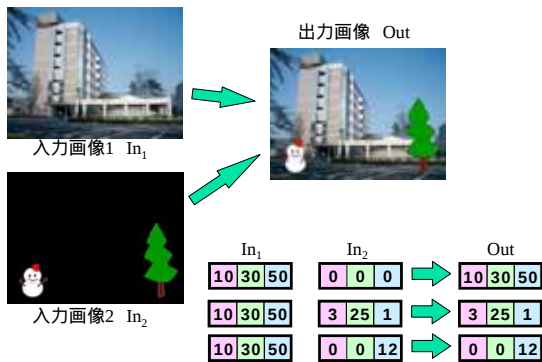
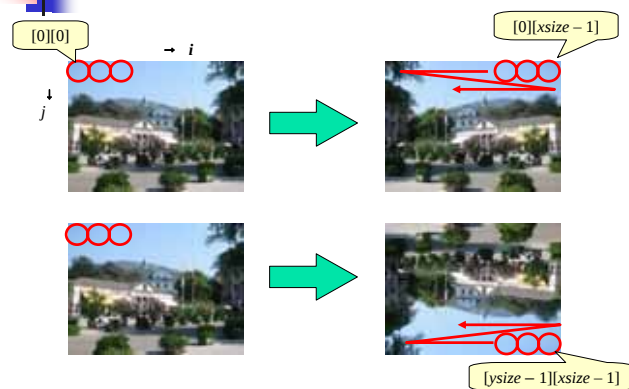


マスク合成(のような処理)



1

左右反転と180度回転



ビットシフトとビット反転

■ 例

1ビット右シフト 11101100 → 01110110

ビット反転 11101100 → 00010011

RGBそれぞれを処理することを忘れないこと



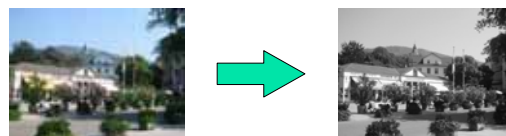
ビットシフト



ビット反転

3

モノクローム画像化



■ 例

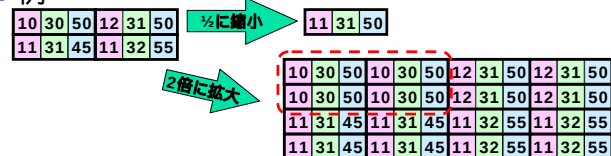
R	100	平均値 125	R	125
G	150		G	125
B	125		B	125

4

縮小と拡大



■ 例



モザイク化



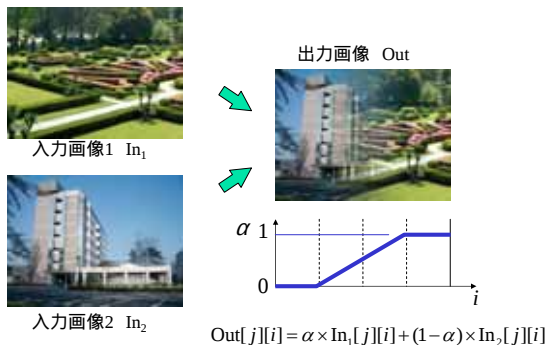
■ 例: 2 × 2画素の平均値で置きかえる



ブロックを大きくするほど粗い画像になる

6

アルファブレンディング



7

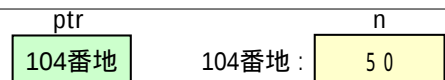
～ポインタの復習(再掲)～

ポインタ型変数宣言の例

```
int n;
int *ptr; ← ptrは int型の変数を指すポインタ変数
```

```
n = 50;
```

```
ptr = &n; ← ptrに n のアドレスを代入
           アドレス演算子 (単項&演算子, &operator)
           データが格納されているアドレスを返す演算子
```



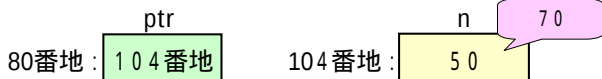
8

間接演算子

```
int n;
int *ptr;

n = 50;
ptr = &n;
*ptr = 70;
```

この“*”は間接演算子 (indirect operator)
ptrが指している変数の中身が70に書き換えられる。



9

PIXEL構造体

```
typedef struct{
    unsigned char r;
    unsigned char g;
    unsigned char b;
} PIXEL, *PPIXEL;
```

?番地 px 200番地

```
int main(void){ //例
    PIXEL x;
    PPIXEL px = &x;
    ...
}
```

200番地 x

r	88
g	B1
b	E7

10

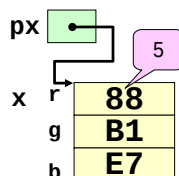
アロー演算子 ->

pxが指しているPIXEL構造体のrの値を変えたい

(*px).r = 5; 正しい

この“*”は間接演算子
ポインタ変数の宣言と混同しない!

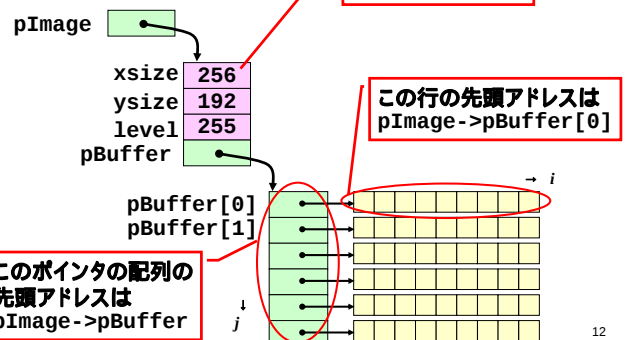
*px.r = 5; 正しくない
(*px.r) = 5と解釈される)



11

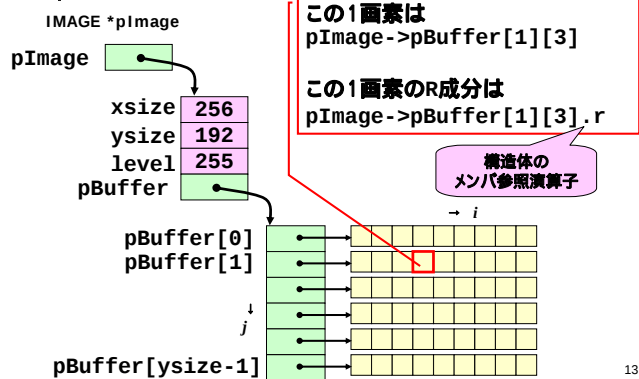
各データの参照方法(1)

```
void ipRotateImage(IMAGE *pImage);
```



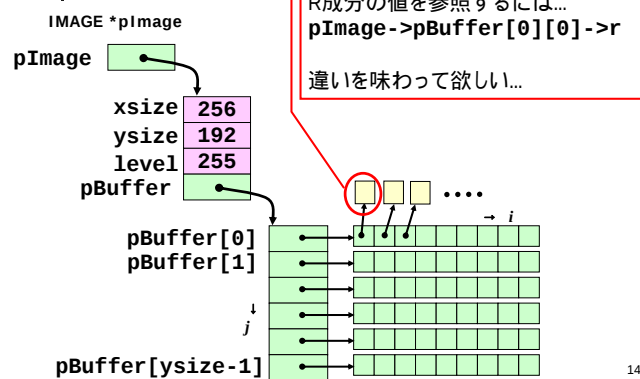
12

各データの参照方法(2)



13

参考



14