

# Grafika Komputerowa. Biblioteka Libpng

Aleksander Denisiuk

Polsko-Japońska Akademia Technik Komputerowych

Wydział Informatyki w Gdańsku

ul. Brzezi 55

80-045 Gdańsk

denisjuk@pja.edu.pl

Libpng

Case study:  
Wizualizacja

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://users.pja.edu.pl/~denisjuk>

## Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

# Libpng

- Jest oficjalną legalną *wzorcową* biblioteką do obsługi PNG
- Ma zaimplementowane prawie wszystkie opcje formatu PNG
- Zgodna z ANSI C (C89)
- Istnieje od roku 1995
  - ☐ najnowsza stabilna wersja 1.6.21
    - 16 stycznia 2016
- Używa zlib w wersji 1.0.4 i wzwyż (zalecana jest wersja 1.2.5)
- Wolna licencja *libpng license*
  - ☐ dostępna w pliku [libpng.h](#)
  - ☐ certyfikowana przez OSI
  - ☐ można wyświetlić w kodzie

```
printf("%s", png_get_copyright(NULL));
```

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:  
Wizualizacja

- W kodzie (C, C++):

```
#include <png.h>
```

- W opcjach konsolidatora (linkera):

```
-lpng
```

## ■ png\_struct

- ☐ struktura wewnętrzna
- ☐ zawsze jest przekazywana jako pierwszy argument do wszystkich funkcji `libpng`

## ■ png\_info

- ☐ zawiera informację o pliku PNG
- nie powinno się mieć (od wersji 1.5.0 nie ma) bezpośredniego dostępu do tych struktur
  - ☐ koresztać z licznych funkcji `png_get_xxx()` oraz `png_set_xxx()`

## ■ Otwieranie pliku

```
FILE *fp = fopen(file_name, "rb");  
if (!fp) return (ERROR);
```

## ■ Sprawdzanie nagłówka PNG

```
png_byte header[8];  
png_size_t number = 8;  
fread(header, 1, number, fp);  
is_png = !png_sig_cmp(header, 0, number);  
if (!is_png) return (NOT_PNG);
```

## ■ png\_struct

```
png_structp png_ptr = png_create_read_struct  
    (PNG_LIBPNG_VER_STRING,  
     (png_voidp)user_error_ptr,  
     user_error_fn,  
     user_warning_fn);  
if (!png_ptr) return (ERROR);
```

## ■ możliwe jest wykorzystanie domyślnych *callbacków*

```
png_structp png_ptr = png_create_read_struct  
    (PNG_LIBPNG_VER_STRING,  
     NULL,  
     NULL,  
     NULL);  
if (!png_ptr) return (ERROR);
```



## ■ png\_info

```
png_infop info_ptr
    = png_create_info_struct(png_ptr);
if (!info_ptr){
    png_destroy_read_struct(&png_ptr,
        (png_infopp)NULL, (png_infopp)NULL);
    return (ERROR);
}
```

```
if (setjmp(png_jmpbuf(png_ptr))) {  
    png_destroy_read_struct(&png_ptr,  
                           &info_ptr, NULL);  
    fclose(fp);  
    return (ERROR);  
}
```

- Trzecim argumentem, zamiast `NULL`, może być `&end_info`

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
png_init_io(png_ptr, fp);  
png_set_sig_bytes(png_ptr, number);  
png_read_info(png_ptr, info_ptr);
```

```
png_uint_32 width, height;  
int color_type, bit_depth;  
png_uint_32 rowbytes;  
png_get_IHDR(png_ptr, info_ptr,  
             &width, &height,  
             &bit_depth, &color_type,  
             &interlace_type,  
             &compression_type, &filter_method);  
rowbytes=png_get_rowbytes(png_ptr,info_ptr);
```

- Trzy ostatnie argumenty mogą być **NULL**
- **width, height** — szerokość i wysokość obrazu w pikselach (do  $2^{31}$ )
- Są również funkcje do pobrania pojedynczych wartości

# png\_get\_IHDR: głębia i paleta kolorów

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

- `color_type`, `bit_depth` — typ palety kolorów i odpowiednia głębia kolorów dla każdego kanału
  - ☐ `PNG_COLOR_TYPE_GRAY` — (bit depth: 1, 2, 4, 8, 16)
  - ☐ `PNG_COLOR_TYPE_GRAY_ALPHA` — (bit depth: 8, 16)
  - ☐ `PNG_COLOR_TYPE_PALETTE` — (bit depth: 1, 2, 4, 8)
  - ☐ `PNG_COLOR_TYPE_RGB` — (bit depth: 8, 16)
  - ☐ `PNG_COLOR_TYPE_RGB_ALPHA` — (bit depth: 8, 16)
- Dla innej głębii kolorów:
  - ☐ `PNG_COLOR_MASK_COLOR`
  - ☐ `PNG_COLOR_MASK_ALPHA`
  - ☐ `PNG_COLOR_MASK_PALETTE`

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

## ■ `interlace_type`

- ☐ `PNG_INTERLACE_NONE`
- ☐ `PNG_INTERLACE_ADAM7`

## ■ `compression_type`

- ☐ `PNG_COMPRESSION_TYPE_BASE`

## ■ `filter_method`

- ☐ `PNG_FILTER_TYPE_BASE`
- ☐ `PNG_INTRAPIXEL_DIFFERENCING`

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
png_bytep * row_pointers
    = (png_bytep*) malloc(sizeof(png_bytep) * height);
for (y=0; y<height; y++)
    row_pointers[y]
        = (png_byte*) malloc(rowbytes);
png_read_image(png_ptr, row_pointers);
```

# Zwolnienie pamięci, zamykanie pliku

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
png_destroy_read_struct(&png_ptr,  
                        &info_ptr, NULL);  
  
fclose(fp);
```



Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
if (color_type != PNG_COLOR_TYPE_RGBA)
    return (ERROR)

for (y=0; y<height; y++) {
    png_byte* row = row_pointers[y];
    for (x=0; x<width; x++) {
        png_byte* ptr = &(row[x*4]);
        printf("Pixel at position [ %d - %d ] \
            has the following RGBA values: \
            %d - %d - %d - %d\n",
            x, y, ptr[0], ptr[1], ptr[2], ptr[3]);

        ptr[0] = 0;
        ptr[1] = ptr[2];
    }
}
```

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
FILE *fp = fopen(file_name, "wb");  
if (!fp) return (ERROR);
```

## ■ png\_struct

```
png_structp png_ptr = png_create_write_struct  
    (PNG_LIBPNG_VER_STRING,  
     (png_voidp)user_error_ptr,  
     user_error_fn,  
     user_warning_fn);  
if (!png_ptr) return (ERROR);
```

## ■ możliwe jest wykorzystanie domyślnych *callbacków*

```
png_structp png_ptr = png_create_write_struct  
    (PNG_LIBPNG_VER_STRING,  
     NULL,  
     NULL,  
     NULL);  
if (!png_ptr) return (ERROR);
```

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:  
Wizualizacja

## ■ png\_info

```
png_infop info_ptr
    = png_create_info_struct(png_ptr);
if (!info_ptr){
    png_destroy_write_struct(&png_ptr,
                           (png_infopp) NULL);
    return (ERROR);
}
```

---

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

---

Wizualizacja

```
if(setjmp(png_jmpbuf(png_ptr))){\n    png_destroy_write_struct(&png_ptr,\n                            &info_ptr);\n    fclose(fp);\n    return (ERROR);\n}
```

```
png_init_io(png_ptr, fp);  
png_set_IHDR(png_ptr, info_ptr,  
             width, height,  
             bit_depth, color_type,  
             interlace_type,  
             compression_type, filter_method);
```

- Sens parametrów jak w funkcji `png_get_IHDR`, ale
  - `compression_type` powinno być `PNG_COMPRESSION_TYPE_DEFAULT`
  - `filter_method` powinno być `PNG_FILTER_TYPE_DEFAULT`
    - może być `PNG_INTRAPIXEL_DIFFERENCING`

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

**Zapisywanie**

main

Case study:

Wizualizacja

```
png_write_info(png_ptr, info_ptr);
```

```
png_write_image(png_ptr, row_pointers);
```

- `row_pointers` jest strukturą, utworzoną tak jak przy wczytywaniu obrazka.



# Końcówka, zwolnienie pamięci, zamykanie pliku

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:

Wizualizacja

```
png_write_end(png_ptr, NULL);

for (y=0; y<height; y++)
    free(row_pointers[y]);
free(row_pointers);
png_destroy_write_struct(&png_ptr,
                        &info_ptr);

fclose(fp);
```

Libpng

Wprowadzenie

Struktury

Wczytywanie

Przetwarzanie

Zapisywanie

main

Case study:  
Wizualizacja

```
int main(int argc, char **argv){  
    if (argc != 3) return (ERROR);  
  
    read_png_file(argv[1]);  
    process_file();  
    write_png_file(argv[2]);  
  
    return 0;  
}
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

## Case study: Wizualizacja

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

- W wyniku numerycznych testów powstała tablica danych (liczb rzeczywistych). Trzeba te dane wizualizować
- Rozwiązanie:
  - ☐ utwórzmy plik png: każdy piksel reprezentuje element tablicy poprzez skalę szarości:
    - najmniejsza wartość — piksel czarny
    - największa wartość — piksel biały
  - ☐ Najmniejsza i największa wartości nie muszą się pokrywać z najmniejszą i największą wartościami pośród danych
  - ☐ Napişmy prosty program w C

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

## ■ Plik wejściowy ma strukturę

- ☐ w pierwszej linijce podana jest szerokość
- ☐ w drugiej — wysokość
- ☐ dalej jest tablica danych `double` w postaci binarnej

`%Nx: 128`

`%Ny: 128`

`.....>?sd<,+\nma????`

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
#include <stdio.h>
#include <stdlib.h>
#include <argtable2.h>
#include <png.h>

#include "input.h"
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
#ifndef INPUT
```

```
#define INPUT
```

```
#include <stdio.h>
```

```
int readSize(FILE * in, int * Nx, int * Ny);
```

```
#endif
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
#include <string.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "input.h"
```

```
int readSize(FILE* in, int* Nx, int* Ny){  
    size_t line_size = 32;  
    char* line;  
    line = malloc(line_size * sizeof (char));  
  
    if( line == NULL)    {  
        perror("Unable to allocate buffer");  
        return 4;  
    }
```



# Moduł input. Implementacja, wczytywanie rozmiaru

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
getline(&line, &line_size, in);
if(strncmp(line, "%Nx: ", 4)==0){
    *Nx = atoi(line+5);
}
else {
    fprintf(stderr, "Bad file format?? (first line)\n");
    free(line);
    return 5;
}
```

■ drugi — analogicznie

```
free(line);
return 0;
} //Koniec funkcji
```

# Opcje wiersza poleceń i inne zmienne.

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
double Max;  
double Min;  
const char * in_pixel_file;  
const char * out_png_file;  
  
FILE * in;  
int i, j;  
int Nx, Ny;  
int size;  
double* R;
```

# Rejestracja opcji wiersza poleceń

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
struct arg_file *in_pixel_file_name
    = arg_file0("i", "in-pixel-file",
                "<input>", "Input Pixels File");

struct arg_file *out_png_file_name
    = arg_file0("o", "out-png-file",
                "<output>", "Output PNG File");

struct arg_dbl *min = arg_dbl0("m", "min" ,
                                "<min>", "Minimum data value");

struct arg_dbl *max = arg_dbl0("M", "max",
                                "<max>", "Maximum data value");

struct arg_lit *help = arg_lit0("h", "help",
                                  "print this help and exit");

struct arg_end *end = arg_end(10); // maksymalna liczba
                                   // błędów 10
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
struct arg_int{
    struct arg_hdr hdr;
    int count;
    int *ival;
};
```

```
struct arg_str{
    struct arg_hdr hdr;
    int count;
    const char **sval;
};
```

.....

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
struct arg_int* arg_int0(const char* shortopts,  
    const char* longopts,  
    const char* datatype,  
    const char* glossary);  
struct arg_int* arg_int1(const char* shortopts,  
    const char* longopts,  
    const char* datatype,  
    const char *glossary);  
struct arg_int* arg_intn(const char* shortopts,  
    const char* longopts,  
    const char *datatype,  
    int mincount,  
    int maxcount,  
    const char *glossary);
```

.....

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
void *argtable[] = {
    in_pixel_file_name, out_png_file_name,
    min, max, help, end};

if (arg_nullcheck(argtable) != 0) {
    printf("error: insufficient memory\n");
    terminate(argtable, 1); // kończy program z kodem 1
}
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
min->dval[0] = -1.5;
max->dval[0] = 2.5;
in_pixel_file_name->filename[0] = "test.pixels";
out_png_file_name->filename[0] = "test.png";
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
int errors;
errors = arg_parse(argc, argv, argtable);

if (errors==0){
    in_pixel_file = in_pixel_file_name->filename[0];
    out_png_file = out_png_file_name->filename[0];
    Min = min->dval[0];
    Max = max->dval[0];
}
else{
    arg_print_errors(stderr, end, "pixels2png");
    arg_print_glossary(stderr, argtable, " %-25s %s\n");
    terminate(argtable, 1);
}
```



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
if (help->count > 0){  
    printf("Usage: pixels2png");  
    arg_print_syntax(stdout, argtable, "\n");  
    arg_print_glossary(stdout, argtable, "  %-25s %s\n");  
    terminate(argtable, 0);  
}
```

# Wczytywanie rozmiaru tablicy

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
in = fopen(in_pixel_file, "rb");
if (!in){
    perror(in_pixel_file);
    terminate(argtable, 2);
}

int err_code = readSize(in, &Nx, &Ny);
if(err_code>0) terminate(argtable, err_code);
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
size = Nx*Ny;  
R = (double *) malloc(size*sizeof(double));  
fread(R, size*sizeof(double), 1, in);  
fclose(in);
```

# Otwieramy plik wyjściowy

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
FILE *fp = fopen(out_png_file, "wb");  
if (!fp){  
    perror(out_png_file);  
    terminate(argtable, 8);  
}
```

# Utworzenie struktury dla zapisu

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
png_structp png_ptr = png_create_write_struct(  
    PNG_LIBPNG_VER_STRING, NULL, NULL, NULL);  
if (!png_ptr) {  
    terminate(argtable, 9);  
}  
png_infop info_ptr  
    = png_create_info_struct(png_ptr);  
if (!info_ptr){  
    png_destroy_write_struct(&png_ptr,  
        (png_infopp) NULL);  
    terminate(argtable, 10);  
}
```

# Zapisywanie nagłówka png

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
png_init_io(png_ptr, fp);

png_set_IHDR(png_ptr, info_ptr,
             Nx, Ny, 8,
             PNG_COLOR_TYPE_GRAY,
             PNG_INTERLACE_NONE,
             PNG_COMPRESSION_TYPE_DEFAULT,
             PNG_FILTER_TYPE_DEFAULT);
printf("\twriting header ...\n");
png_write_info(png_ptr, info_ptr);
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
printf("\twriting image data ...\n");
png_bytep row_pointer=new png_byte[Nx];
for (int j=0;j<Ny;j++){
    for(int i=0;i<Nx;i++){
        int val=(R[Nx*j+i]-Min)/(Max-Min)*256+0.5;
        if (val>255) val=255;
        else if (val<0) val=0;
        row_pointer[i]=(png_byte)val;
    }
    png_write_row(png_ptr, row_pointer);
}
free row_pointer;
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
png_write_end(png_ptr, NULL);  
fclose(fp);  
png_destroy_write_struct(&png_ptr,  
                        &info_ptr);  
  
arg_freetable(argtable,  
              sizeof(argtable)/sizeof(argtable[0]) );  
  
return 0;  
}
```



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
void terminate(void* *argtable, int exitcode){  
    if(argtable == NULL) printf("NULL\n");  
    arg_freetable(argtable,  
                  sizeof(argtable)/sizeof(argtable[0]));  
    exit(exitcode);  
}
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
GCC = gcc
LIBS = -lpng -lm -lopt
SOURCES = $(wildcard *.c)
OBJECTS := $(SOURCES:.c=.o)
EXECUTABLE = pixels2png

all: $(SOURCES) $(EXECUTABLE)

$(EXECUTABLE): $(OBJECTS)
    $(GCC) $(OBJECTS) -o $@ $(LIBS)

clean:
    rm -f $(EXECUTABLE) $(OBJECTS)
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

**Makefile**

Użycie

ImageMagick

Makefile dwa

\$ make

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
$ ./pixels2png -i test2.pixels -o test2.png \  
                                -m -0.5 -M 1.5
```

```
Max: 1.5
```

```
Min: -0.5
```

```
Nx: 128
```

```
Ny: 128
```

```
in pixels file: test2.pixels
```

```
out PNG file: test2.png
```

```
reading pixels data from test2.pixels...
```

```
outputting PNG data to test2.png...
```

```
    writing header ...
```

```
    writing image data ...
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
$ ./pixels2png --help
```

```
Usage: pixels2png [-h] [-i <input>] [-o <output>] [-m <m
```

```
-i, --in-pixel-file=<input> Input Pixels File
```

```
-o, --out-png-file=<output> Output PNG File
```

```
-m, --min=<min> Minimum data value
```

```
-M, --max=<max> Maximum data value
```

```
-h, --help print this help and exit
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

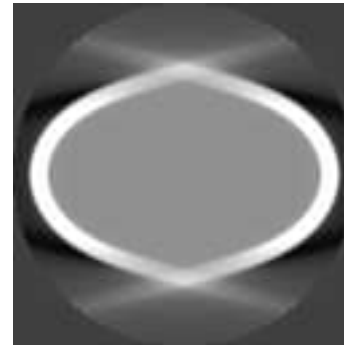
Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

## ■ ImageMagick

- ☐ legalny, wieloplatformowy, ma API dla wielu języków programowania (w tym PHP)
- ☐ [www.imagemagick.org/](http://www.imagemagick.org/)



```
convert -scale 1600% test2.png test2x1600.png
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

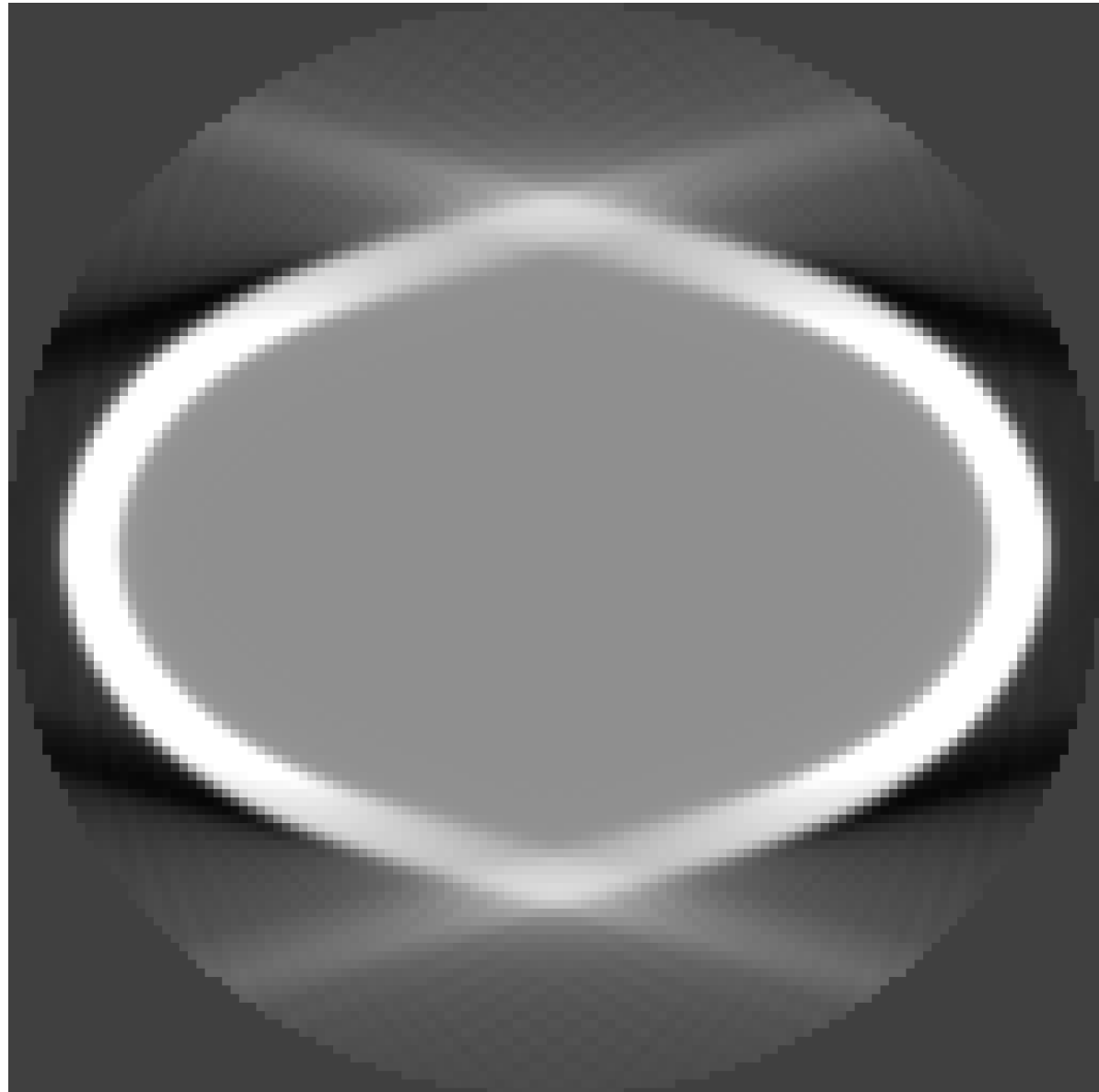
Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa





Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
montage -pointsize 72 -title 'test 2 results' \  
-geometry +16+16 test2x1600.png \  
test2x1600title.png
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

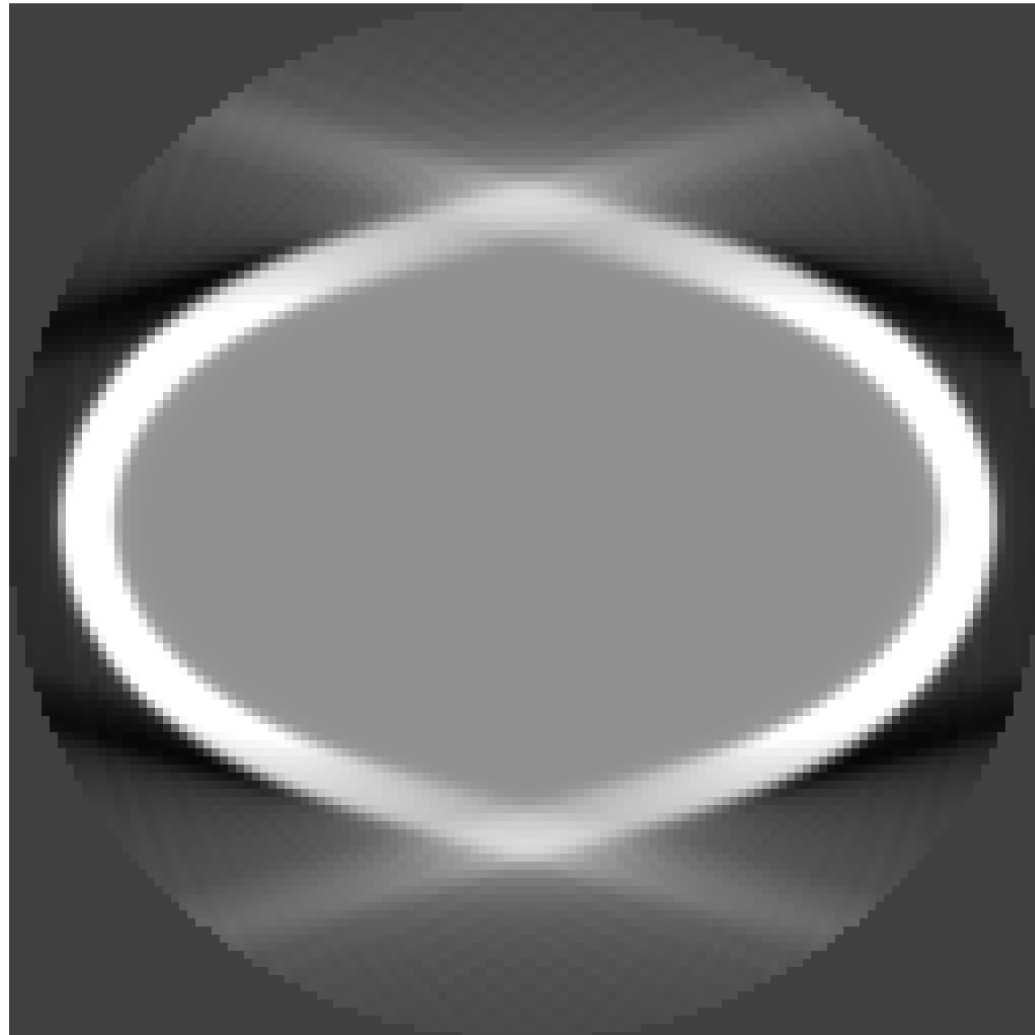
Makefile

Użycie

ImageMagick

Makefile dwa

test 2 results



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

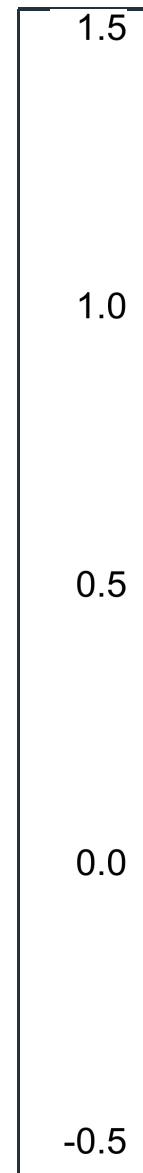
Użycie

ImageMagick

Makefile dwa

```
convert -size 150x2272 -background white \  
-fill black -interline-spacing 470 \  
-gravity east -pointsize 72 \  
label:'1.5\n1.0\n0.5\n0.0\n-0.5' scale.png
```

Libpng
Case study: Wizualizacja
Zagadnienie
Implementacja
Makefile
Użycie
ImageMagick
Makefile dwa



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
convert -size 160x2272 gradient:white \  
gradient.png
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa



Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
montage -tile 3x1 \  
test2x1600title.png \  
-label ' ' scale.png \  
gradient.png \  
-geometry +2+0 \  
wynik.png
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

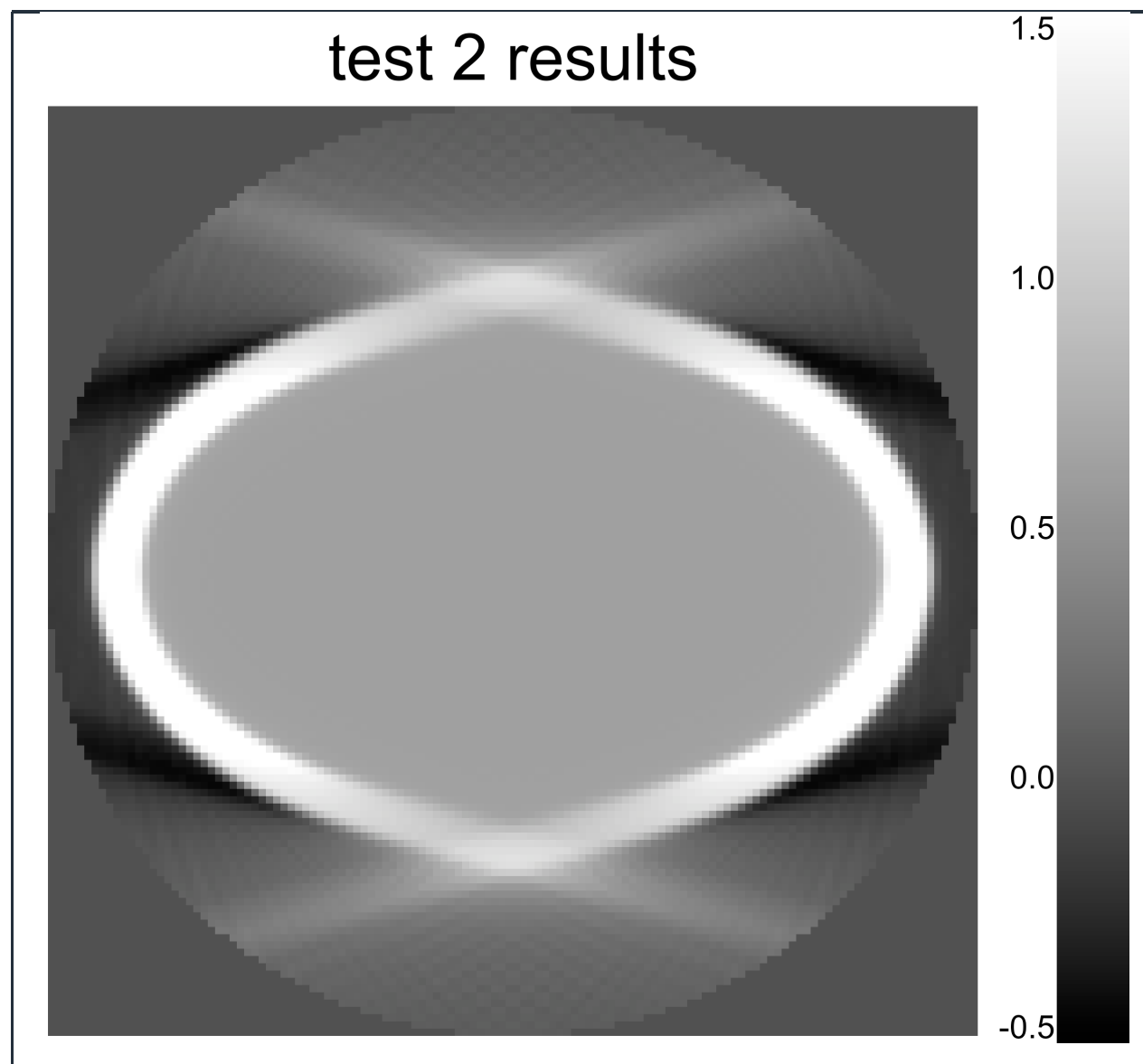
Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa





Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

## ■ Zmienne

```
PNGDIR = Output
```

```
MAX = 1.50
```

```
MIN = -0.50
```

```
MED = $(shell echo "scale=2;\n\n($${MAX}+${MIN})*0.5" | bc )
```

```
Q1 = $(shell echo "scale=2;\n\n($${MAX}+${MIN})*0.25" | bc )
```

```
Q3 = $(shell echo "scale=2;\n\n($${MAX}+${MIN})*0.75" | bc )
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
all: $(SOURCES) $(EXECUTABLE) images
images: $(patsubst %.pixels, %.png,\
    $(addprefix $(PNGDIR)/, $(wildcard *.pixels)))
$(PNGDIR)/%.png: %x1600title.png gradient.png scale.png
    montage -tile 3x1 $< -label ' '\
    scale.png gradient.png -geometry +2+0 $@
gradient.png:
    convert -size 160x2272 \
    gradient:white gradient.png
scale.png:
    convert -size 150x2272 -background white\
    -fill black -interline-spacing 470\
    -gravity east -pointsize 72 \
    label: '$(MAX)\n$(Q3)\n$(MED)\n$(Q3)\n$(MIN) '\
    scale.png
```

Libpng

Case study:  
Wizualizacja

Zagadnienie

Implementacja

Makefile

Użycie

ImageMagick

Makefile dwa

```
%x1600title.png: %x1600.png
montage -pointsize 72 -title \
'test 2 results' -geometry +16+16 $< $@
```

```
%x1600.png: %.png
convert -scale 1600% $< $@
```

```
%.png: %.pixels
./$(EXECUTABLE) -i $< -o $@ \
-M $(MAX) -m $(MIN)
```