

Progetto C: Image Processing Library

I Periferici

Gabriele Cecconello 870751

Edoardo Mazzon 870606

In questo report verranno descritte le operazioni di maggiore importanza: verrà tralasciata la spiegazione di funzioni semplici per non cadere in ovvietà esplicative.

1. Parte 1: gestione della memoria

```
ip_mat * ip_mat_create(unsigned int h, unsigned int w, unsigned int k, float v);
```

Per allocare abbastanza spazio in heap per una `ip_mat`, effettuiamo una prima `malloc` in cui allochiamo memoria per una struct `ip_mat`. In seguito, allochiamo la matrice tridimensionale *data* $h \times w \times k$ e anche spazio sufficiente a contenere un array di k stats.

Si ritorna il puntatore a tale struttura dati.

```
void ip_mat_free(ip_mat *a);
```

Per liberare la memoria occupata da una `ip_mat`, effettuiamo una `free` per ogni dimensione della matrice tridimensionale *data*; successivamente liberiamo l'array *stat* e infine il puntatore alla struct di input *a* stesso.

Nel caso in cui il puntatore a `ip_mat` in input puntasse a NULL, non viene effettuata nessuna `free`.

```
void ip_mat_init_random(ip_mat * t, float mean, float var);
```

Riceviamo in input una media *mean* e una varianza *var* e, dal momento che la funzione preimplementata `get_normal_random` restituisce un valore casuale da una distribuzione gaussiana di media $\mu = 0$ e varianza $\sigma = 1$, otteniamo il nuovo valore casuale dai nuovi parametri *mean* e *var* nel seguente modo:

$$\frac{(x - \mu)}{\sigma} \quad \text{dove } x \text{ è output di } \text{get_normal_random}()$$

2. Parte 2: operazioni semplici sulle immagini

```
ip_mat * ip_mat_to_gray_scale(ip_mat * in);
```

La scala di grigi si ottiene riassegnando ad ogni pixel dei valori R, G e B simili tra loro. Tali valori si calcolano facendo la media tra i canali di ogni pixel.

```
ip_mat * ip_mat_blend(ip_mat * a, ip_mat * b, float alpha);
```

Il valore $\alpha \in [0, 1]$. Con valori vicini a zero sarà più trasparente l'immagine rappresentata da *b* e viceversa.

```
ip_mat * ip_mat_corrupt(ip_mat * a, float amount);
```

Per ogni pixel viene introdotto del rumore bianco gaussiano prodotto da una normale standard e moltiplicato ad *amount*. Vengono effettuati dei controlli per verificare che i nuovi valori dei pixel così risultanti rimangano nell'intervallo $[0, 255]$.

3. Parte 3: convoluzione e filtri

```
ip_mat * ip_mat_convolve(ip_mat * a, ip_mat * f);  
float get_convolved_value(ip_mat *filtro, ip_mat *sottomatrice, int canale);
```

Per effettuare la convoluzione tra una *ip_mat* immagine *a* e una *ip_mat* filtro *f*, ci appoggiamo alla funzione ausiliare *get_convolved_value*. Questa prende in input due *ip_mat* di uguali dimensioni e un canale e restituisce il valore risultante della somma dei prodotti, del canale passato in input, tra le celle corrispondenti delle due *ip_mat*.

La funzione *ip_mat_convolve* prevede uno scorrimento della *ip_mat* immagine *a* su ogni canale di ogni pixel, e nel ciclo più interno dello scorrimento richiama l'ausiliare *get_convolved_value* passando come parametri il filtro *f*, una sottomatrice di *a* delle stesse dimensioni del filtro e il canale corrente.

```
ip_mat * ip_mat_padding(ip_mat * a, int pad_h, int pad_w);
```

La funzione *ip_mat_padding* prende in input una *ip_mat* *a*, due interi *pad_h* e *pad_w* e restituisce una nuova *ip_mat* le cui dimensioni risultano essere quelle della matrice in input *a* sommate, per lato, a *pad_h* e *pad_w*. I valori R, G, B delle nuove celle che incorniciano la vecchia matrice *a* sono inizializzati a 0.

```
ip_mat * create_gaussian_filter(int w, int h, int k, float sigma);
```

La funzione *create_gaussian_filter* crea una *ip_mat* di dimensioni *w*, *h* e *k* e ne inizializza ogni valore secondo la funzione *G* così definita:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Dove *x* e *y* sono le distanze di una certa cella *i, j* dal centro del filtro generato.

4. Prova di compilazione

Per testare la nostra libreria è stata apportata una modifica a riga 63 del main "*main_iplib.c*"; tale modifica è la seguente:

Prima:

```
concat_images = atoi(argv[5]);
```

Modifica:

```
if(argc>5){  
    concat_images = atoi(argv[5]);  
}
```

Tale modifica si è resa necessaria in quanto il programma, che poteva essere legalmente invocato con 5 parametri, e quindi *argc* = 5, provocava un accesso illegale e incontrollato all'array *argv* in posizione 5, generando *segmentation fault*.

Suddetto file main è stato prima compilato ed eseguito con le seguenti righe di comando:

```
$gcc -Wall main_iplib.c bmp.o -lm -g -O1  
$valgrind -v --leak-check=full ./eseguibile
```

Con seguente risultato dopo l'esecuzione (il secondo warning generato da *main_iplib.c* è dovuto alla modifica sopra descritta, ma è risolvibile inizializzando *concat_images* a 0):

```
mag 8 14:16
gabriele@gabriele-Nitro-ANS15-52: ~/Desktop/Nuova(funziona)

gabriele@gabriele-Nitro-ANS15-52:~/Desktop/Nuova(funziona)$ gcc -Wall -c bmp.c -lm -g -O1
bmp.c: In function 'bm_load_palette':
bmp.c:5543:5: warning: ignoring return value of 'fgets', declared with attribute warn_unused_result [-Wunused-result]
 5543 |     fgets(buf, sizeof buf, f);
      |     ^~~~~~

gabriele@gabriele-Nitro-ANS15-52:~/Desktop/Nuova(funziona)$ gcc -Wall main_iplib.c bmp.o -lm -g -O1
main_iplib.c: In function 'main':
main_iplib.c:124:7: warning: 'concat_images' may be used uninitialized in this function [-Wmaybe-uninitialized]
   124 |     if(concat_images) {
       |     ^

gabriele@gabriele-Nitro-ANS15-52:~/Desktop/Nuova(funziona)$ valgrind -v --leak-check=full ./a.out flower.bmp flower.bmp gauss outflower.bmp 1 5 5
==3602== Memcheck, a memory error detector
==3602== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==3602== Using Valgrind-3.15.0-608c611914-20190413 and LibVEX; rerun with -h for copyright info
==3602== Command: ./a.out flower.bmp flower.bmp gauss outflower.bmp 1 5 5
==3602==
--3602-- Valgrind options:
--3602--   -v
--3602--   --leak-check=full
--3602-- Contents of /proc/version:
--3602--   Linux version 5.3.0-51-generic (buildd@lgw01-amd64-034) (gcc version 9.2.1 20191008 (Ubuntu 9.2.1-9ubuntu2)) #44-Ubuntu SMP Wed Apr 22
21:09:44 UTC 2020
--3602--
--3602-- Arch and hwcaps: AMD64, LittleEndian, amd64-cx16-lzcnt-rdtscp-sse3-ssse3-avx-avx2-bmi-f16c-rdrand
--3602-- Page sizes: currently 4096, max supported 4096
--3602-- Valgrind library directory: /usr/lib/x86_64-linux-gnu/valgrind
--3602-- Reading syms from /home/gabriele/Desktop/Nuova(funziona)/a.out
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/ld-2.30.so
--3602--   Considering /usr/lib/x86_64-linux-gnu/ld-2.30.so ..
--3602--   .. CRC mismatch (computed ff246dee wanted d35d27d8)
--3602--   Considering /lib/x86_64-linux-gnu/ld-2.30.so ..
--3602--   .. CRC mismatch (computed ff246dee wanted d35d27d8)
--3602--   Considering /usr/lib/debug/lib/x86_64-linux-gnu/ld-2.30.so ..
--3602--   .. CRC is valid
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/valgrind/memcheck-amd64-linux
--3602--   object doesn't have a symbol table
--3602--   object doesn't have a dynamic symbol table
--3602-- Scheduler: using generic scheduler lock implementation.
--3602-- Reading suppressions file: /usr/lib/x86_64-linux-gnu/valgrind/default.supp
==3602== embedded gdbserver: reading from /tmp/vgdb-pipe-from-vgdb-to-3602-by-gabriele-on-???
==3602== embedded gdbserver: writing to /tmp/vgdb-pipe-to-vgdb-from-3602-by-gabriele-on-???
==3602== embedded gdbserver: shared mem /tmp/vgdb-pipe-shared-mem-vgdb-3602-by-gabriele-on-???
==3602==
==3602== TO CONTROL THIS PROCESS USING vgdb (which you probably
==3602== don't want to do, unless you know exactly what you're doing,
==3602== or are doing some strange experiment):
==3602== /usr/lib/x86_64-linux-gnu/valgrind/../../bin/vgdb --pid=3602 ...command...
==3602==
==3602== TO DEBUG THIS PROCESS USING GDB: start GDB like this
==3602== /path/to/gdb ./a.out
==3602== and then give GDB the following command
==3602== target remote | /usr/lib/x86_64-linux-gnu/valgrind/../../bin/vgdb --pid=3602
==3602== --pid is optional if only one valgrind process is running
==3602==
--3602-- REDIR: 0x4021b90 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c9c82 (???)
--3602-- REDIR: 0x4021b90 (ld-linux-x86-64.so.2:strlen) redirected to 0x580c9c82 (???)
--3602-- REDIR: 0x4021960 (ld-linux-x86-64.so.2:index) redirected to 0x580c9c9c (???)
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_core-amd64-linux.so
--3602--   object doesn't have a symbol table
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so
--3602--   object doesn't have a symbol table
==3602== WARNING: new redirection conflicts with existing -- ignoring it
--3602--   old: 0x4021b90 (strlen) R-> (0000.0) 0x580c9c82 ???
--3602--   new: 0x4021b90 (strlen) R-> (2007.0) 0x0483e060 strlen
--3602-- REDIR: 0x401e370 (ld-linux-x86-64.so.2:strncmp) redirected to 0x483efd0 (strncmp)
--3602-- REDIR: 0x40220f0 (ld-linux-x86-64.so.2:mempcpy) redirected to 0x4842a20 (mempcpy)
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/libm-2.30.so
--3602--   Considering /usr/lib/x86_64-linux-gnu/libm-2.30.so ..
--3602--   .. CRC mismatch (computed 35f2fc38 wanted 0c2c98dd)
--3602--   Considering /lib/x86_64-linux-gnu/libm-2.30.so ..
--3602--   .. CRC mismatch (computed 35f2fc38 wanted 0c2c98dd)
--3602--   Considering /usr/lib/debug/lib/x86_64-linux-gnu/libm-2.30.so ..
--3602--   .. CRC is valid
--3602-- Reading syms from /usr/lib/x86_64-linux-gnu/libc-2.30.so
--3602--   Considering /usr/lib/x86_64-linux-gnu/libc-2.30.so ..
--3602--   .. CRC mismatch (computed ebee1ba4 wanted 1c32c83a)
--3602--   Considering /lib/x86_64-linux-gnu/libc-2.30.so ..
--3602--   .. CRC mismatch (computed ebee1ba4 wanted 1c32c83a)
--3602--   Considering /usr/lib/debug/lib/x86_64-linux-gnu/libc-2.30.so ..
--3602--   .. CRC is valid
```

```

--3602-- .. CRC is Valid
--3602-- REDIR: 0x4a4d630 (libc.so.6:memmove) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c930 (libc.so.6:strncpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d960 (libc.so.6:strcasestr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c250 (libc.so.6:strcat) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c990 (libc.so.6:rindex) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4ee00 (libc.so.6:rawmemchr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a69e90 (libc.so.6:wmemchr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a699d0 (libc.so.6:wscmp) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d790 (libc.so.6:mempcpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d5c0 (libc.so.6:bcmp) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c8c0 (libc.so.6:strncmp) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c300 (libc.so.6:strcmp) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d6f0 (libc.so.6:memset) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a69990 (libc.so.6:wcschr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c820 (libc.so.6:strnlen) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c3e0 (libc.so.6:strcspn) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d9b0 (libc.so.6:strncasestr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c380 (libc.so.6:strcpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d000 (libc.so.6:mempcpy@GLIBC_2.14) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a6b100 (libc.so.6:wcsnlen) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a69a10 (libc.so.6:wscpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c9d0 (libc.so.6:stpncpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c2b0 (libc.so.6:index) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c7e0 (libc.so.6:strlen) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a55d50 (libc.so.6:memchr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4da00 (libc.so.6:strcasestr_l) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d580 (libc.so.6:memchr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a69ae0 (libc.so.6:wcslen) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4c90 (libc.so.6:strspn) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d900 (libc.so.6:stpncpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d900 (libc.so.6:stpncpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d8a0 (libc.so.6:stpncpy) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4ee40 (libc.so.6:strchrnul) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4da50 (libc.so.6:strncasestr_l) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4a4d4a0 (libc.so.6:strstr) redirected to 0x48301d0 (_vgnU_ifunc_wrapper)
--3602-- REDIR: 0x4b359c0 (libc.so.6:_strchr_avx2) redirected to 0x483da10 (rindex)
--3602-- REDIR: 0x4b35b90 (libc.so.6:_strlen_avx2) redirected to 0x483df40 (strlen)
--3602-- REDIR: 0x4a47290 (libc.so.6:malloc) redirected to 0x483a780 (malloc)
--3602-- REDIR: 0x4a4d150 (libc.so.6:_GI_strstr) redirected to 0x4842ca0 (__strstr_sse2)
--3602-- REDIR: 0x4b38ba0 (libc.so.6:_mempcpy_avx_unaligned_erms) redirected to 0x48419f0 (memmove)
--3602-- REDIR: 0x4b32180 (libc.so.6:_mempcpy_avx2_movbe) redirected to 0x48411e0 (bcmp)
--3602-- REDIR: 0x4a47880 (libc.so.6:free) redirected to 0x483b9d0 (free)
--3602-- REDIR: 0x4b39020 (libc.so.6:_memset_avx2_unaligned_erms) redirected to 0x48418e0 (memset)
--3602-- REDIR: 0x4b38b80 (libc.so.6:_mempcpy_avx_unaligned_erms) redirected to 0x4842660 (mempcpy)
--3602-- REDIR: 0x4b31090 (libc.so.6:_strcmp_avx2) redirected to 0x483eed0 (strcmp)
--3602-- REDIR: 0x4a68590 (libc.so.6:_strstr_sse2_unaligned) redirected to 0x4842c20 (strstr)
==3602==
==3602== HEAP SUMMARY:
==3602==   in use at exit: 0 bytes in 0 blocks
==3602==   total heap usage: 12,186,548 allocs, 12,186,548 frees, 233,890,254 bytes allocated
==3602==
==3602== All heap blocks were freed -- no leaks are possible
==3602==
==3602== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
gabriele@gabriele-Nitro-AN515-52: ~/Desktop/Nuova(funziona)$

```

Quindi deallocando con successo tutti i blocchi di memoria allocati dinamicamente.

E in seguito ricompilato con le seguenti righe di comando, riportate nel *makefile*:

```

$gcc -Wall -c bmp.c
$gcc -Wall --ansi --pedantic -lm -g3 -O3 -fsanitize=address -fsanitize=undefined -std=gnu89 -Wextra -o execute main_iplib.c bmp.o

```

Con seguente risultato dopo l'esecuzione (il warning è dovuto alla modifica a *main_iplib.c*, ma è risolvibile iniziando *concat_images* a 0):

```

gabriele@gabriele-Nitro-AN515-52: ~/Desktop/Nuova(funziona)$ gcc -Wall -c bmp.c
gabriele@gabriele-Nitro-AN515-52: ~/Desktop/Nuova(funziona)$ gcc -Wall main_iplib.c bmp.o -lm --ansi --pedantic -lm -g3 -O3 -fsanitize=address -fsanitize=undefined -std=gnu89 -Wextra
main_iplib.c:124:7: warning: 'concat_images' may be used uninitialized in this function [-Wmaybe-uninitialized]
 124 |     if(concat_images) {
      |     ^
gabriele@gabriele-Nitro-AN515-52: ~/Desktop/Nuova(funziona)$ ./a.out flower.bmp flower.bmp gauss outputflower.bmp 1 5 5
gabriele@gabriele-Nitro-AN515-52: ~/Desktop/Nuova(funziona)$

```

Quindi con nessun memory leak.

Per eseguire il programma compilato dal *makefile* è quindi necessaria la riga di comando:

```
./execute paraminput1 paraminput2 operazione paramOutput concat(1, 0) val1 val2
```

Dal momento che il *makefile* crea un eseguibile chiamato "execute".