Laboratorium 10

Suwaki

Historia pokazuje, że większość z Was uwielbia interaktywne programy. Konsole, w których 30 razy program pytał użytkownika o kolejne punkty (x,y) środków rysowanych okręgów cieszyły oczy zarówno prowadzącego jak i Waszych kolegów i koleżanek… Moment, w którym nie raz cały ten proces zwieńczany był epickim „wywaleniem się" programu i trzeba było 30 wartości podawać jeszcze raz zapadał w pamięć tak bardzo, że gościł w snach prowadzącego na długo po tym, powodując nocne stany lękowe…

Niemniej jednak to prawda, że interaktywne programy mają w sobie „to coś". Najprostszym przykładem aktywnej interakcji z programem, napisanym z wykorzystaniem OpenCV, jest tworzenie tzw. „suwaków" , które umożliwiają dynamiczną zmianę wybranego parametru używanego przez program. Zacznijmy od zdefiniowania zmiennych globalnych (tak, też tego nie lubię ale to tak tylko na chwilę – żeby było łatwiej) (tak, to te nad funkcją *main*).

```
Mat before, after;
const char* name = "Sliders example";
const int slider_max = 100;
int slider = 0;
```

Kolejne zmienne oznaczają:

- *before*, *after* – patrz lista 9,
- *name* – nazwa okna, w którym będzie wyświetlany obraz i suwak,
- *slider_max* – maksymalna wartość na suwaku,
- *slider* – aktualna wartość suwaka.

Wczytywanie obrazka i tworzenie okna wykonujemy w sposób standardowy (tak, to już jest w funkcji *main*):

```
before = imread("sharp.jpg", 1);
namedWindow(name, CV_WINDOW_AUTOSIZE);
```

Teraz czas na zasadnicze tworzenie suwaka:

```
char TrackbarName[50];
sprintf(TrackbarName, "%d options", (slider_max+1));
createTrackbar(TrackbarName, name, &slider, slider_max, changeBlur);
changeBlur(0, 0);
```

Co tu się dzieje:

- *TrackbarName* – zmienna „trzymająca” opis suwaka (zrozumiesz później),
- *sprintf* – funkcja, która robi w tej chwili tylko tyle, że pozwala nam „dynamicznie” (tylko za sprawą zmiany w kodzie zmiennej *slider_max*) zmienić opis suwaka,
- *createTrackbar* – główna funkcja tworząca suwak – ostatnia zmienna to nazwa innej funkcji uruchamianej przy każdym ruchu suwaka – musi ona mieć odpowiednią postać o czym poczytasz tutaj, a co prawdę powiedziawszy niewiele nas w tej chwili obchodzi... (ale poczytaj!!!)

Oczywiście musimy także utworzyć naszą funkcję *changeBlur*. To ona będzie odpowiedzialna za modyfikację wyświetlanego obrazu. Załóżmy więc, że to co chcemy w obrazie zmieniać to stopień zniekształcenia obrazu po zastosowaniu funkcji *blur* (trudno się było tego domyśleć, nieprawdaż?). Wtedy nasza funkcja powinna wyglądać tak (tak, należy ją dopisać nad funkcją *main*) (nie, nie nad zmiennymi globalnymi – pod nimi):

```
void changeBlur(int, void*)
{
    blur(before, after, Size(slider+1, slider+1), Point(-1,-1));
    imshow(name, after);
}
```

Jeszcze raz zwrócę uwagę na to, że należy pamiętać o odpowiedniej formie tej funkcji. Mimo, że tak naprawdę nic tutaj do tej funkcji nie przekazujemy to musimy „powiedzieć” o tym jakie zmienne ta funkcja może przyjmować. Bo twórca tak chciał, a więc trzeba się z tym liczyć. Zawartość tej funkcji powinna być już Wam znana.

Wybieralne[1] zadania „do domu”:

- pobaw się suwakiem (nie tym suwakiem – tym, który zrobiłeś w trakcie zajęć),
- dodaj drugi suwak i spróbuj zmieniać dowolne dwie wybrane wartości mające wpływ na wyświetlany obraz (np. jasność i kontrast),
- wczytaj nowy obrazek i dodaj suwak, który będzie zmieniał efekt przetwarzania obrazu (np. pozycja suwaka 1 – *blur* obrazu, pozycja 2 – erozja obrazu, pozycja 3 – dylatacja obrazu, itp. itd.).

[1] wybieralne – ja je dla Was wybrałem, Wy je musicie zrobić – wszystkie.