Przetwarzanie i Analiza Danych Multimedialnych

System porównywania zmian otoczenia

Konfiguracja

Maciej Wadowski i Dawid Karaś

Wymagania oraz biblioteki

Program tworzony był w oparciu o język Python z interpreterem w wersji 3.9. Do implementacji zostały wykorzystane takie biblioteki jak:

- numpy (1.21.4)
- scikit-image (0.19.2)
- opencv-python (4.5.5.64)
- imutils (0.5.4)

Uruchamianie programu

Do prawidłowego działania programu, powinien on być uruchamiany w głównym folderze (“piadm”) komendą “python software/main.py”. Do porównywania wykorzystywane są pliki “left.jpg” oraz “right.jpg” z folderu “resources”. Przetworzone obrazy oraz efekt docelowy znajdują się w folderze “obrazy_i_dane”.

Zmienne sterujące

Podczas implementowania w kodzie zostało zawartych kilka informacji, którymi warto manipulować w zależności od tego na jakich obrazach będzie wykorzystywany system (rozmiar) czy jakiej wielkości chcemy znajdować różnice. Pierwszym takim elementem jest:

```
thresh = height * width * 0.01 * 0.01
```

Linijka ta określa jakiej wielkości zmiany będą zaznaczone na obrazie wynikowym (w powyższym przypadku wielkość prostokąta traktowanego jako element zmieniony wynosi 1% rozmiaru całego obrazu).

Kolejnym elementem jest porównywanie średniej kolorów (zmniejszonej do 8) dla prostokątów, które zostały uwzględnione jako zmiana. Informacja ta zawiera się w:

```
if abs(mean_diff) > 10:
```

Dla powyższego przykładu różnica średnich kolorów dla zmienionego pojedynczego fragmentu (prostokąta) w obrazie pierwotnym oraz docelowym wynosi 10. Być może jednak warto w zależności od sytuacji wziąć pod uwagę odchylenie standardowe lub inną różnicę średnich albo całkowicie inną metodą porównania kolorów.

Warto także wspomnieć, że filtry morfologiczne domyślnie mają przyjęty poziom erozji oraz dylatacji w ilości 3, który może być także prosto zmieniony poprzez dodanie kolejnego argumentu wywołań funkcji “dilate” oraz “erode” jako zmienną typu całkowitego.

Następnym elementem wartym uwzględnienia jest detektor krawędzi Canny’ego, który domyślnie przyjmuje $\sigma = 0.5$, a w zależności od wymagań także można to zmienić w poniższym fragmencie:

```
res = detect_edges(grayscale_img, 0.5)
```

Być może także wartym uwzględnienia elementem jest wybór samych obrazków do porównania. Ze względu na brak takiej potrzeby nie jest on parametryzowany podczas uruchamiania programu (być może powinien), ale jak powyższe zmienne można wybrane obrazki zmienić w poniższym miejscu w kodzie:

```
left = imread('resources/right1.jpg')
```

```
right = imread('resources/right2.jpg')
```

Wszystkie powyższe zmienne znajdują się w głównym pliku programu (“main.py”).