

Sprawozdanie z laboratorium:
Komunikacja człowiek-komputer

Część III: Obrazy

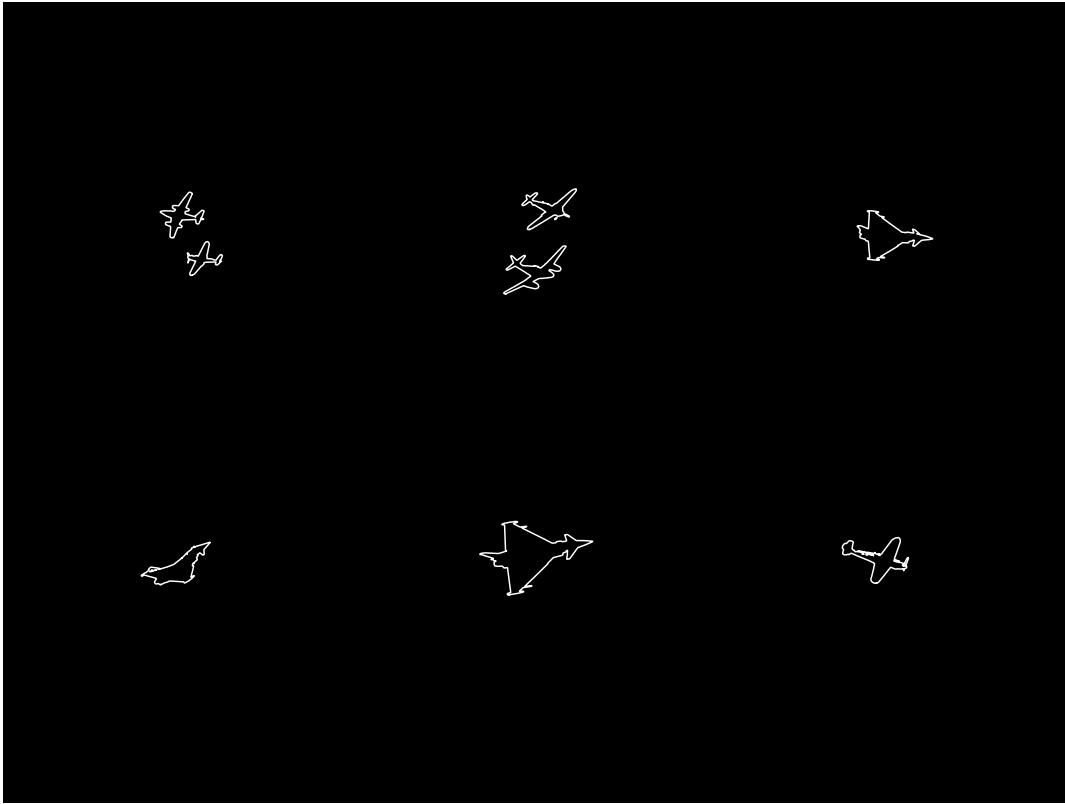
8 listopada 2016

Prowadzący: dr hab. inż. Maciej Komosiński

Autor: **Michał Lewiński** inf122505 WI michal.lewinski@student.put.poznan.pl

Zajęcia środowe, 16:50.

Oświadczam/y, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autora/ów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.



Rysunek 1: Mozaika 6 obrazów, przedstawiających kontury różnych samolotów

1 Wstęp

Zadanie laboratoryjne polegało na zapoznaniu się z różnymi metodami przekształcania obrazu oraz znalezienia krawędzi obiektów na obrazie. Do obsługi obrazu posłużono się biblioteką *scikit-image*.

2 Mozaika samolotów

Celem ćwiczenia było utworzenie mozaiki 6 obrazów, które przedstawiałyby tylko białe kontury samolotów na czarnym tle. Aby uzyskać kontury posłużono się odpowiednimi przekształceniami, które pozwoliły w sposób ogólny (nie zależny od obrazu) znaleźć krawędzie szukanych obiektów. Algorytm tworzenia mozaiki przedstawia się następująco:

1. Wczytanie pliku obrazu do macierzy z wartościami (R,G,B), za pomocą funkcji *imread* [1].
2. Utworzenie osobnego subplotu dla obrazu.
3. Zwiększenie kontrastu obrazu za pomocą tzw. „contrast stretching”, gdzie obraz jest przeskalowany, w taki sposób aby zawierał wszystkie wartości, które przypadają pomiędzy dwoma podanymi percentylami (w tym przypadku 1 i 20) [2].

4. Następnie dochodzi do konwersacji z przestrzeni *RGB* do przestrzeni *HSV*, gdyż potrzebujemy tylko wartości *V*.
5. Na podstawie wartości *V* i przy użyciu funkcji *findContours* znajdujemy krawędzie obiektów. Funkcja ta wykorzystuje algorytm „marching squares” i zwraca listę odszukanych krawędzi, które są następnie wykorzystywane przy rysowaniu konturów [3].

Literatura

- [1] Scikit-Image. Imread – skimage. <http://scikit-image.org/docs/dev/api/skimimage.data.html#skimage.data.imread>.
- [2] Scikit-Image. Contrast stretching – skimage. http://scikit-image.org/docs/dev/auto_examples/plot_equalize.html.
- [3] Scikit-Image. Contour finding – skimage. http://scikit-image.org/docs/dev/auto_examples/plot_contours.html.