
	Politechnika Bydgoska im. J. J. Śniadeckich Wydział Telekomunikacji, Informatyki i Elektrotechniki Zakład Systemów Teleinformatycznych		
Przedmiot	Przetwarzanie obrazów		
Prowadzący	mgr inż. Grzegorz Czczot		
Temat	Greenscreen (1)		
Student			
Nr lab.	6	Data wykonania	
Ocena		Data oddania spr.	

1. Cel ćwiczenia

Podczas tego laboratorium wiedza zgromadzona na poprzednich zajęciach zostanie wykorzystana do działań z zielonym ekranem.

2. Wstęp teoretyczny

Greenscreen to narzędzie wykorzystywane obecnie bardzo często przy montażu profesjonalnych (i nie) filmów oraz video streamingu. Odcień zielonego umieszczony na ekranie jest dość rzadko spotykany w rzeczywistości i dość prosty do wyeliminowania na potrzeby nagrań. Wiele filmowych nagrań wykonuje się na zielonym tle, przez co można w etapie postprodukcji podmienić tło lub wygenerować je komputerowo. Można również „podłożyć” stały obraz w miejsce zielonego tła, co będzie efektem tego laboratorium.

2.1. Jak to zrobić?

Algorytm jest następujący:

1. poszukać ładnego obrazka na tło i wczytać do programu;
2. wczytać do programu selfie zrobione przed zielonym ekranem ;
3. dopasować wielkość obrazów;
4. przepisać obraz z selfie do HSV;
5. wybrać zakres kolorów – można skorzystać ze strony: <https://colorpicker.me/> ;
6. utworzyć maskę z obrazu;
7. wykonać operacje bitowe i dodawania, aby scalić obrazy
8. wyświetlić obraz z podmienionym tłem.

3. Zadania do samodzielnego wykonania

3.1. Zadanie 1.

Zaimplementować algorytm z 2.1. w kodzie.

3.2. Zadanie 2.

Zmienić kod zadania 1 tak, aby krawędzie maski były wyrównane. Sprawdzić działanie dla filtru Gaussa, filtru medianowego oraz morfologicznego otwarcia i domknięcia. Który z wymienionych daje najlepsze rezultaty?

3.3. Zadanie 3.

Zrobić kolejne selfie na ekranie (zmienić oświetlenie). Czy kolory ustawione w zakresie `inRange` są poprawne? Jeśli nie, dostosować je do nowych warunków oświetleniowych.

3.4. Zadanie 4.

Zmienić kod programu tak, aby wyświetlony obraz był widoczny w ramce wczytanej z pliku PNG (np. podobnej do takiej: https://img.freepik.com/darmowe-wektory/szablon-ramki-posta-na-instagramie_1393-59.jpg?w=2000).

3.5. Zadanie 5.

Zmienić kod programu tak, aby tło było podmienione na obrazie live z kamery lub filmie.

3.6. Zadanie 6.

Zmienić kod programu tak, aby jeśli wskazany kolor zielony nie będzie widoczny na obrazie, pojawi się informacja „No greenscreen” na klatkach filmu.

3.7. Zadanie 7.

W OpenCV dość prosto można zaimplementować wykrywanie twarzy na obrazie. W tym celu obraz należy wczytać w skali szarości (`cvtColor`) oraz wyrównać jego histogram (`equalizeHist`). Aby wykryć twarz na obrazie, można skorzystać z jednego z wyuczonych klasyfikatorów (np. `haarcascade_frontalface_default.xml`) i użyć metody `detectMultiScale`.

Napisać program, który na zdjęciu znajdzie twarz i zaznaczy ją prostokątem.

3.8. Zadanie 8.

Zmienić kod programu tak, aby twarz była śledzona (zaznaczana na kolejnych klatkach analizowanego filmu).

3.9. Zadanie 9.

Zgromadzić zdjęcia z całej grupy. Czy wszystkie zdjęcia pozwalają na wykrycie twarzy?

4. Sprawozdanie

W sprawozdaniu należy zawrzeć:

- wypełnioną tabelę z początku instrukcji;
- skopiowane istotne części kodu programów napisanych w trakcie zajęć;
- opis wykonanych zadań ze zrzutami ekranu;
- własne spostrzeżenia jako wnioski.