

Ćwiczenie CAMMOUSE

Urządzenia wskazujące („camera mouse”)

Cele laboratorium:

- Zapoznanie się z zagadnieniami projektowania urządzeń wskazujących.

Przygotowanie

W przypadku realizacji ćwiczenia w laboratorium: w katalogu wskazanym przez prowadzącego utwórz podkatalog, w którym będą zapisywane wszystkie dane podczas ćwiczenia. Nazwa katalogu powinna zawierać: **<nr ćwiczenia>_<data>_<inicjały osoby realizującej ćwiczenie>**; zapisz i rozpakuj pliki ćwiczenia (pobrane z UPEL) do utworzonego katalogu; po skończonych zajęciach pamiętaj o usunięciu danych z dysku.

W przypadku realizacji ćwiczenia zdalnie:

- wymagania: MATLAB R2023a z modułem: Image Acquisition Toolbox, Image Processing Toolbox, Audio Toolbox, DSP System Toolbox, Deep Learning Toolbox: kamera internetowa, mikrofon, następujące AddOns: MATLAB Support Package for USB Webcams, Image Acquisition Toolbox Support Package for OS Generic Video Interface, Deep Learning Toolbox Model for AlexNet Network

Informacje na temat sprawozdania z ćwiczenia zostały zamieszczone na końcu instrukcji

Proszę o zgłaszanie nieścisłości w instrukcji drogą emailową do prowadzącego zajęcia.

Wymagania

Wykonane wcześniej ćwiczenia:

- Ćwiczenie TRACK: Śledzenie twarzy w systemie rozpoznawania mimiki

Część I – kalibracja i ocena wydajności systemu

Gdzie szukać dodatkowych informacji:

- Internet – słowa kluczowe: human-computer interfaces
- System pomocy środowiska MATLAB i modułu Image Processing Toolbox oraz Computer Vision System Toolbox
- [1] Ken Hinckley, "Input Technologies and Techniques", Microsoft Research:
<https://www.microsoft.com/en-us/research/publication/input-technologies-techniques/>
- [2] <https://www.billbuxton.com/input04.Taxonomies.pdf>

W tej części ćwiczenia będzie używany przykład śledzenia twarzy z ćwiczenia TRACK. Kod został uzupełniony i odpowiednio zmodyfikowany, ale zasadniczy algorytm śledzenia twarzy nie został zmieniony.

Celem tej części ćwiczenia będzie sprawdzenie wydajności systemu (szybkość działania) oraz jego kalibracja.

- 1) Otwórz skrypt `testHarness_CAMMOUSE_cz1.m` i zapoznaj się z nim. Ustaw parametry (numer kamery) w sekcji oznaczonej `UZUPEŁNIJ_1`:

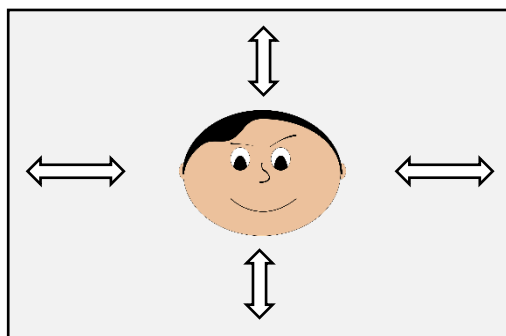
`parametryImportu.nr`

Ta część ćwiczenia wymaga do działania kamery. Jeśli kamera nie działa, możesz spróbować zrobić kalibrację offline używając nagranego pliku video. W tym celu w skrypcie należy odpowiednio zmodyfikować fragmenty kodu oznaczone `%VIDEO` (2 miejsca). Opcjonalnie można skorzystać z innego interfejsu do kamery (2 miejsca oznaczone „Alternatywny interfejs kamery”).

Skrypt realizuje następującą funkcjonalność:

- pobranie ramki z kamery
- wywołanie śledzenia twarzy (`myAlgorithmCAMMOUSE.m`)
 - śledzenie podobnie jak w ćwiczeniu TRACK, przy czym pominięto detekcję ust i nosa
 - istotnym rezultatem z algorytmu są współrzędne ROI wykrytej twarzy na podstawie których wyznaczany jest środek ROI (środek twarzy)
- eksport danych do pliku TXT (`myExportDataCAMMOUSE.m`)
 - nazwa pliku wyjściowego: `calibdata.txt`
 - format zapisywanych danych CSV (ang. comma separated files):
`iter, facex, facey, t2`
(numer iteracji, współrzędna x twarzy, współrzędna y twarzy, czas przetwarzania poprzedniej ramki)
- wizualizacja (`myVisualizationCAMMOUSE.m`)

- 2) Uruchom skrypt i zbierz dane potrzebne do kalibracji sterowania kursorem przy pomocy ruchów głowy. Wcześniej ustaw kamerę tak, aby twarz znajdowała się w centrum obrazu i zajmowała jego niewielką część (zbyt duża twarz może powodować gubienie śledzenia przy brzegach ekranu). W trakcie działania skryptu poruszaj głową maksymalnie w prawo, w lewo, w górę i w dół, tak jakbyś chciał/chciała sterować kursorem na ekranie.



Ponieważ za każdym uruchomieniem skryptu nadpisywany jest plik TXT, po zakończonym działaniu skryptu zrób kopię utworzonego pliku tekstowego, np. z nazwą `calibdata_1.txt` (kopia będzie używana do kalibracji).

- 3) Otwórz skrypt `kalibrujDane1.m` i wykonuj kolejne sekcje kodu, notując w sprawozdaniu rezultaty oraz uzupełniając odpowiednie fragmenty kodu:
- nie zapomnij zmienić nazwę pliku TXT z poprzedniego etapu tak aby odpowiadała nazwie pliku w tym skrypcie
 - wyznaczanie średniego FPS (frames per second)
 - wyznaczanie współczynników kalibracyjnych dla sterowania bezwzględnego

Zaobserwuj trajektorię ruchu twarzy i zastanów się w jaki sposób można przełożyć ruchy twarzy w fizycznej przestrzeni (odwzorowanej na obrazie 2D) na ruch kursora na ekranie komputera.

Sprawdź w literaturze (wykład „Interakcja i Komputer”, [2]) czym różni się sterowanie bezpośrednie od pośredniego (ang. relative vs absolute Controllers).

Jednym z parametrów urządzeń wejściowych jest współczynnik C-D (ang. control to display ratio), określający mapowanie fizycznego przemieszczenia urządzenia wskazującego na ruch kursora na ekranie.

$$CD = \frac{dx \text{ (przemieszczenie urządzenia w rzeczywistym świecie)}}{DX \text{ (przemieszczenie kursora na ekranie)}}$$

Zanotuj obliczone współczynniki CD dla osi x oraz y (oznaczone dalej w kodzie `CDx`, `CDy`). Poprawne współczynniki powinny pozwolić na kalibrację w taki sposób, aby wartości współrzędnych x i y po kalibracji mieściły się w zakresie $<0,1>$, czyli aby maksymalnemu wychyleniu głowy w prawo/lewo/góra/dół odpowiadały współrzędne 0/1,0/1. Zwróć uwagę, że DX (przemieszczenie kursora na ekranie) jest znormalizowane w zakresie 0-1. Dzięki temu można je przeskalować łatwo do dowolnej rozdzielczości.

- rezultaty działania skryptu zapisywane są do pliku `calibldata.mat` używanego później w trakcie działania systemu.
- 4) W przypadku małego FPS (<10) możesz dobrać (`parametryImportu`) odpowiednią rozdzielczość akwizycji oraz skalowania obrazu tak aby zwiększyć FPS. Zanotuj w sprawozdaniu użyte parametry.

Część II – sterowanie bezwzględne

Celem tej części ćwiczenia będzie realizacja sterowania bezwzględnego kursorem.

- Otwórz skrypt `testHarness_CAMMOUSE_cz2.m` i zapoznaj się z nim.
 - Skopiuj dobrane wcześniej (`testHarness_CAMMOUSE_cz1.m`) parametry akwizycji z kamery (rozdzielczość...).

Ta część ćwiczenia wymaga do działania kamery. Jeśli kamera nie działa, możesz spróbować zrobić kalibrację offline używając nagranego pliku video. W tym celu w skrypcie należy odpowiednio zmodyfikować fragmenty kodu oznaczone `%VIDEO` (2 miejsca). Opcjonalnie można skorzystać z innego interfejsu do kamery (2 miejsca oznaczone „Alternatywny interfejs kamery”).

 - Ustaw odpowiednie parametry algorytmu `parametryAlg.calibName` – upewnij się, że nazwa pliku MAT z danymi kalibracyjnymi jest poprawna (domyślnie jest ustawiona nazwa `calibldata.mat`)
 - Otwórz plik `sterowanie1.m` i zapoznaj się z nim. Funkcja ta jest odpowiedzialna za generowanie sterowania kursorem. Twoim zadaniem będzie uzupełnienie odpowiednich fragmentów kodu (oznaczonych `UZUPEŁNIJ`) oraz przetestowanie działania sterowania kursorem.
 - Funkcja przyjmuje jako argumenty wejściowe: obraz, ROI twarzy `[x,y,w,h]` oraz wyznaczone w procesie kalibracji parametry: `offset_x`, `offset_y`, `CDx`, `CDy`
 - Na podstawie ROI twarzy wyznaczany jest jej środek - współrzędne `x`, `y` - które następnie będą odpowiednio normalizowane przy pomocy współczynników z kalibracji.
 - Obliczone współrzędne kursora (`x_cursor`, `y_cursor`) są dodatkowo korygowane, tak aby nie wykraczały poza zakres `<0,1>` oraz odwracane są osie (tak aby ruch głowy w prawo odpowiadał ruchowi kursora w prawo). Współrzędne kursora są zwracane przez funkcję i przekazywane do wizualizacji.
 - Podpowiedź: zwróć uwagę, iż wyznaczanie współrzędnych kursora (`x_cursor`, `y_cursor`) odbywa się podobnie jak w skrypcie do kalibracji (`kalibrujDane1.m`)
- Uruchom skrypt `testHarness_CAMMOUSE_cz2.m` i sprawdź poprawność sterowania kursorem.
 - Spróbuj tak sterować kursorem, aby w oknie wykresu po prawej narysować prostokąt. Zanotuj rezultat (przycisk „zrzut ekranu” – tworzy plik `sterowanie_rezultat_1.jpg`)
 - Spróbuj tak dobrać kalibrację (patrz poprzedni punkt ćwiczenia) aby sterowanie było jak najbardziej wygodne. Zanotuj rezultat sterowania (wraz z nowymi parametrami kalibracji).
 - W sprawozdaniu zamieść fragmenty uzupełnionego kodu oraz rezultaty. Zanotuj wnioski.
 - Jeśli potrzebne jest wydłużenie czasu pojawiania się na ekranie trajektorii ruchu (czerwone kropki), można zmodyfikować kod w skrypcie `myVisualizationCAMMOUSE.m` w linii 36 `obj.cursorData = -1*zeros(30*5,2);`
liczba 30 odpowiada za długość bufora

Część III – sterowanie względne

Celem tej części ćwiczenia będzie realizacja sterowania względnego kursorem. Ruchy głowy będą pełniły rolę wychyleń joysticka (prawo, lewo, góra, dół) sterującego kursorem.

1) Otwórz skrypt `testHarness_CAMMOUSE_cz3.m` i zapoznaj się z nim.

Ta część ćwiczenia wymaga do działania kamery. Jeśli kamera nie działa, możesz spróbować zrobić kalibrację offline używając nagranych plików video. W tym celu w skrypcie należy odpowiednio zmodyfikować fragmenty kodu oznaczone `%VIDEO` (2 miejsca). Opcjonalnie można skorzystać z innego interfejsu do kamery (2 miejsca oznaczone „Alternatywny interfejs kamery”).

- Otwórz plik `sterowanie2.m` i zapoznaj się z nim. Funkcja ta jest odpowiedzialna za generowanie sterowania kursorem.
 - Funkcja przyjmuje jako argumenty wejściowe: obraz, ROI twarzy `[x,y,w,h]` oraz ustawione w głównym skrypcie parametry (nie bierzemy pod uwagę parametrów z procesu kalibracji, ale jest możliwe rozbudowanie skryptu o taką funkcjonalność).
 - Na podstawie ROI twarzy wyznaczany jest jej środek - współrzędne `x, y` - które następnie są odpowiednio skalowane do zakresu `<0,1>` wraz z uwzględnieniem translacji do środka układu współrzędnych. W ten sposób wychylenie głowy od pozycji spoczynkowej (`x2, y2`) jest traktowane jak wychylenie joysticka. Dla uproszczenia przyjmujemy, że głowa powinna być położona na środku obrazu z kamery. Dodatkowo, aby uwzględnić niedokładność położenia, wprowadzony został parametr histerezy – wychylenie jest wykrywane dopiero po przekroczeniu progu (`parametryAlg.histereza`) w każdą ze stron. Po przekroczeniu progu, aktualne odchylenie głowy od pozycji spoczynkowej jest mnożone przez współczynniki wzmocnienia i dodawane do aktualnego położenia kursora na ekranie.
 - Obliczone współrzędne kursora (`x_cursor, y_cursor`) są dodatkowo korygowane, tak aby nie wykraczały poza zakres `<0,1>` oraz odwracane są osie (tak aby ruch głowy w prawo odpowiadał ruchowi kursora w prawo). Współrzędne kursora są zwracane przez funkcję i przekazywane do wizualizacji (wraz z parametrami oraz informacjami o odchyleniu).

2) Uruchom skrypt `testHarness_CAMMOUSE_cz3.m` i sprawdź poprawność sterowania kursorem.

- Spróbuj tak sterować kursorem, aby w oknie wykresu po prawej narysować prostokąt. Zanotuj rezultat (przycisk „zrzut ekranu” – tworzy plik `sterowanie_rezultat_1.jpg`).
- Spróbuj tak dobrać parametry (histereza, wzmocnienia) aby sterowanie było jak najbardziej wygodne. Zanotuj rezultaty sterowania (wraz z parametrami) dla wykonanych prób.
- W sprawozdaniu zamieść fragmenty uzupełnionego kodu oraz rezultaty. Zanotuj wnioski.

Część IV (opcjonalna) – gra Pong

1. Otwórz skrypt `myVisualizationCAMMOUSEpong.m` i zapoznaj się z nim
2. Zmodyfikuj skrypt `testHarness_CAMMOUSE_cz3.m` tak aby używać do wizualizacji skryptu: `myVisualizationCAMMOUSEpong.m`

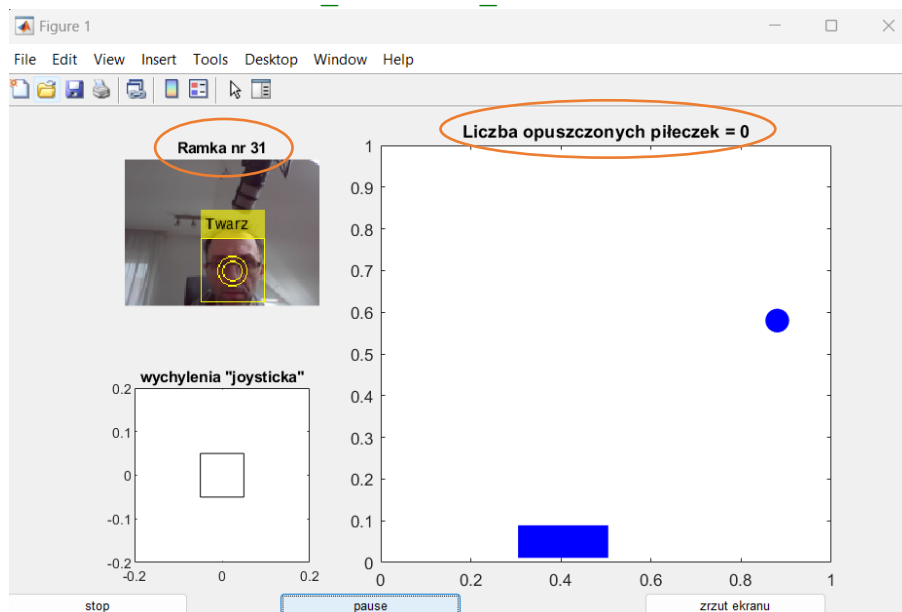
Linie:

```
%-wizualizacja danych
showObj          = myVisualizationCAMMOUSE();
```

zamień na:

```
%-wizualizacja danych
showObj          = myVisualizationCAMMOUSEpong();
```

3. Uruchom `testHarness_CAMMOUSE_cz3.m`



- a. Domyślnie skrypt uruchamia się w trybie pauzy, tak aby dać użytkownikowi czas na powiększenie okna i przygotowanie się do gry.
 - b. Rozpocznij grę (przycisk *continue*)
 - c. Zrób zrzut ekranu ze swoim rekordem i zamieść w sprawozdaniu (minimalna liczba opuszczonych piłeczek w jak najdłuższym czasie/liczbie ramek).
4. Podobnie jak wyżej, zmodyfikuj skrypt `testHarness_CAMMOUSE_cz2.m` a następnie zanotuj swój rekord w sprawozdaniu robiąc zrzut ekranu.

Raport z ćwiczenia

Raport z ćwiczenia należy dostarczyć poprzez system UPEL, w formacie **PDF**. Raport składa się z 3 części. Pierwsza część zawiera rezultaty otrzymane w trakcie realizacji ćwiczenia i należy ją umieścić na UPEL pod koniec ćwiczeń. Kolejne części (analiza i wnioski oraz odpowiedzi na pytania), należy umieścić na UPEL w terminie określonym przez prowadzącego zajęcia (najczęściej jest to tydzień czasu – przed następnym ćwiczeniem).

W sprawozdaniu proszę zamieścić:

Data realizacji ćwiczenia i godzina: Imię i nazwisko:
<u>Rezultaty</u> 1) Cz.I: Zamieść rezultaty oraz fragmenty kodu uzupełnione w trakcie ćwiczenia. 2) Cz.II: Zamieść rezultaty oraz fragmenty kodu uzupełnione w trakcie ćwiczenia. 3) Cz.III: Zamieść rezultaty oraz fragmenty kodu uzupełnione w trakcie ćwiczenia. 4) Cz.IV: Zamieść zrzuty ekranu przedstawiające Twoje rekordy w grze PONG
<u>Analiza i wnioski</u> 1) Cz.I: Zamieść wnioski i odpowiedzi na pytania z tej części ćwiczenia (patrz m.in. 2 pytania w skrypcie <code>kalibrujDane1.m</code>) 2) Cz.II: Zanotuj wnioski ze sterowania bezwzględnego. Czy szybkość analizy video jest odpowiednia do sterowania ? 3) Cz.III: Zanotuj wnioski ze sterowania względnego. Zastanów się czy wprowadzenie zamiast stałych współczynników wzmocnienia, współczynników zależnych od np. stopnia wychylenia lub szybkości ruchu poprawiłoby wygodę sterowania?
<u>Pytania</u> 1) Cz.I: Czym różni się sterowanie bezpośrednie od pośredniego? 2) Cz.III: Co to jest funkcja przejścia (transfer function)? 3) Zanotuj (ankieta na UPEL "Postęp realizacji ćwiczenia na 1szych zajęciach") na pierwszych zajęciach z tego ćwiczenia, do którego punktu udało Ci się zrealizować ćwiczenie w trakcie zajęć"