

Ćwiczenie TRACK

Śledzenie twarzy w systemie rozpoznawania mimiki

Cele laboratorium:

- Zapoznanie się z algorytmami detekcji i śledzenia twarzy.
- Zapoznanie się z elementami systemu rozpoznawania mimiki.

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 R2020a z modułem: Image Acquisition Toolbox, Image Processing Toolbox, Audio Toolbox, DSP System Toolbox, kamera internetowa, mikrofon, następujące AddOns: MATLAB Support Package for USB Webcams, Image Acquisition Toolbox Support Package for OS Generic Video Interface

Informacje na temat sprawozdania z ćwiczenia zostały zamieszczone na końcu instrukcji

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

Informacje teoretyczne

Obserwując komunikację człowieka z innymi ludźmi łatwo można stwierdzić, że bardzo ważną jest tu komunikacja niewerbalna, której istotnym elementem jest mimika twarzy. Ten kanał łączności między człowiekiem a komputerem i innymi systemami technicznymi (na przykład robotem medycznym albo wózkiem inwalidzkim) odgrywa szczególną rolę w przypadku niektórych osób dotkniętych szczególnie głęboką niepełnosprawnością.

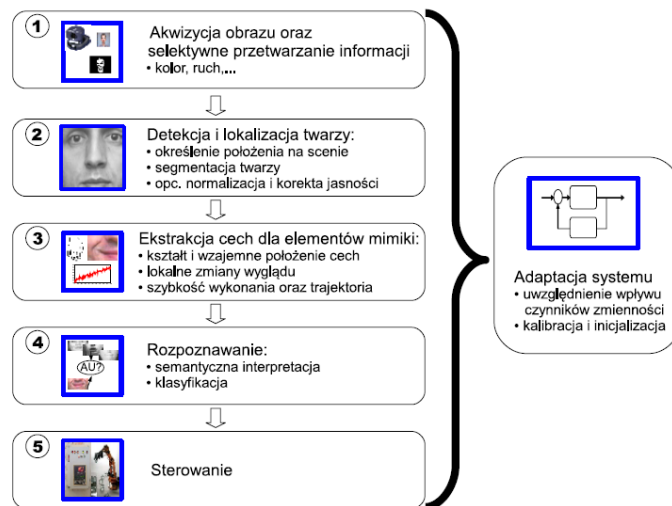
Wykorzystanie elementów mimiki do sterowania, wymaga uwzględnienia szeregu zagadnień, wśród których szczególnie istotne są: psychofizjologiczne uwarunkowania człowieka oraz sposób reprezentacji i opisu mimiki.

Pośród systemów opisu mimiki można wyróżnić metodykę FACS (ang. Facial Action Coding System) — oparty na anatomii system pomiaru ruchów mimicznych twarzy, grupujący je w jednostki czynnościowe (ang. action units) ściśle związane z anatomia muskulatury twarzy.

Sterowanie komputerem przy pomocy gestów mimicznych, wymaga od projektanta wzięcia pod uwagę szeregu czynników. Oprócz specyfikacji technicznej (dobór kamery, sposobu transmisji danych...) i informatycznej (określenie w jaki sposób sygnał wizyjny będzie interpretowany przez komputer), konieczne jest uwzględnienie wymagań wynikających z typowych zadań jakie wykonuje użytkownik podczas pracy z interfejsem graficznym.

Rozpoznawanie obiektów przez ludzi wydaje się łatwe i bezproblemowe. Na podstawie ogólnej wiedzy o danym obiekcie, np. sylwetka ludzka — człowiek natychmiast jest w stanie rozpoznać nowe osoby, niezależnie od tego z jakiego punktu widzenia są one obserwowane czy też np. siedzą lub stoją. Z punktu widzenia komputerowej analizy obrazów, rozpoznawanie wymaga wyodrębnienia z obrazu użytecznej informacji na temat interesujących obiektów. W przypadku rozpoznawania mimiki problem staje się jeszcze bardziej skomplikowany. Przykładowy schemat systemu rozpoznawania mimiki pokazuje poniższy rysunek. Można w nim wyróżnić elementy takie jak: detekcja i lokalizacja twarzy, **śledzenie twarzy** (pominięte na schemacie) oraz rozpoznawanie na podstawie wykrytych cech i ich własności. Elementy mimiki mogą charakteryzować się różnymi atrybutami, które należy zlokalizować i rozpoznać na obrazie. Są to :

- Atrybuty stałe, związane z budową morfologiczną twarzy (oczy, usta, brwi, bruzdy). Są one widoczne cały czas i mogą podlegać deformacji w zależności od rodzaju gestu. Zaliczamy do nich: kształt elementów twarzy (np. brwi) oraz ich wzajemne położenie (np. wielkość szczeliny ust).
- Atrybuty zmienne, wśród których wymienić należy różnego rodzaju zmarszczki, bruzdy oraz fałdy skórne. Może wystąpić również deformacja już istniejących bruzd lub pogłębienie zmarszczek. Powstają one podczas ruchów mimicznych i wywołują najczęściej lokalne zmiany wyglądu.
- Atrybuty dynamiczne takie jak np. szybkość wykonania gestu lub trajektoria ruchu. Mimo iż człowiek jest w stanie rozpoznać np. emocje ze statycznego obrazu, dynamika zmian wyglądu twarzy niesie dużą część informacji.



Rysunek 5.2: Ogólny schemat systemu rozpoznawania mimiki.

Część I – detekcja twarzy oraz jej atrybutów

1. Pobierz kilka obrazów z kamery zawierających twarz widzianą pod różnymi kątami i z różnych odległości. Parametry akwizycji (o ile to możliwe dla posiadanego typu kamery): 320*240, RGB.
2. Zapoznaj się z dokumentacją do obiektu `vision.CascadeObjectDetector` oraz spróbuj użyć tego obiektu do detekcji twarzy na obrazie `testowy_0_0000.jpeg` oraz własnych obrazach (przykład poniżej).

```
>> doc vision.CascadeObjectDetector
>> detector = vision.CascadeObjectDetector();
>> RGB = imread(<twoja nazwa pliku>);
>> bboxes = detector(RGB);
>> liczbaWykrytychTwarzy = size(bboxes,1)
```

3. Zapoznaj się z dokumentacją do funkcji `detectMinEigenFeatures` oraz spróbuj użyć tej funkcji do detekcji punktów charakterystycznych (przykład poniżej) na obrazie `testowy_0_0000.jpeg` oraz własnych obrazach.

```
>> doc detectMinEigenFeatures
>> GRAY=rgb2gray(RGB);
>> corners = detectMinEigenFeatures(GRAY);
>> corners

>> figure; imshow(RGB)
>> hold on; plot(corners.selectStrongest(250))
```

4. Zapoznaj się z dokumentacją do funkcji `insertShape`, `insertMarker`, `bbox2points` oraz spróbuj użyć ich do zwizualizowania rezultatów detekcji twarzy i atrybutów. Zwróć uwagę iż w poniższym przykładzie ograniczamy detekcję cech tylko do obszaru twarzy. W sprawozdaniu zamieść przykładowe rezultaty detekcji twarzy oraz jej cech (wewnątrz obszaru twarzy) dla przynajmniej 3 własnych obrazów oraz obrazu `testowy_0_0000.jpeg`. Zanotuj wnioski. Czy wszystkie punkty charakterystyczne, które zostały wykryte w obszarze twarzy odpowiadają cechom związanym z budową morfologiczną twarzy – atrybuty stałe (brwi, usta, itp.)? W jakich miejscach koncentrują się wykryte punkty?

```
>> doc insertShape
>> doc insertMarker
>> doc bbox2points
>> corners2 = detectMinEigenFeatures(GRAY, 'ROI', bboxes(1, :));

>> xyPoints = corners2.Location;
```

```
>> bboxPoints = bbox2points(bboxes(1, :));  
>> bboxPolygon = reshape(bboxPoints', 1, []);  
>> RGB2 = insertShape(RGB, 'Polygon', bboxPolygon, 'LineWidth', 3);  
>> RGB2 = insertMarker(RGB2, xyPoints, '+', 'Color', 'white');  
>> figure; imshow(RGB2)
```

5. Spróbuj wykryć oczy i nos przy pomocy detektora `vision.CascadeObjectDetector`. Zanotuj rezultaty oraz liczbę wykrytych atrybutów twarzy (nos, usta) dla każdego z detektora.

```
>> doc insertObjectAnnotation  
  
>> noseDetector = vision.CascadeObjectDetector('Nose');  
>> mouthDetector = vision.CascadeObjectDetector('Mouth');  
  
>> noseBBox1=step(noseDetector, RGB);  
>> mouthBBox1=step(mouthDetector, RGB);  
  
>> RGB2 = RGB;  
>> RGB2 = insertObjectAnnotation(RGB2, 'rectangle',noseBBox1,'nos');  
>> RGB2 = insertObjectAnnotation(RGB2, 'rectangle',mouthBBox1,'usta');  
>> imshow(RGB2)
```

6. Opracuj prosty algorytm, który będzie weryfikował czy wykryty nos i usta zostały wykryte prawidłowo. Poniżej przykład zawierający warunki : wykrywanie cech tylko w obszarze ROI twarzy oraz sprawdzenie pola powierzchni wykrytej cechy.

W razie potrzeby dodaj własne warunki weryfikacji cech (np. pole powierzchni obszaru cechy w określonych granicach > lub <, położenie środka cechy w określonym obszarze twarzy np. nos – środek, usta dół twarzy, itd).

W sprawozdaniu zamieść kod oraz rezultaty wraz z komentarzem. Testy przeprowadź na obrazie *testowy_0_0000.jpeg* oraz własnych obrazach.

```
>> faceIM = imcrop(RGB, bboxes(1, :)); % wycięcie obszaru twarzy  
>> imshow(faceIM)  
  
% zawężenie detekcji cech tylko do obszaru twarzy  
>> noseBBox2=step(noseDetector, faceIM);  
>> mouthBBox2=step(mouthDetector, faceIM);
```

```
% dodatkowe warunki weryfikujące cechy
RGB2 = faceIM;
RGB2 = insertObjectAnnotation(RGB2, 'rectangle', noseBBBox2, 'nos');
figure; subplot(1,2,1); imshow(RGB2)

srodekTwarzy = [bboxes(1,3)/2 bboxes(1,4)/2]
% uwaga bboxes to współrzędne przed wycięciem twarzy
srodekNosa = [noseBBBox2(:,1)+noseBBBox2(:,3)/2 noseBBBox2(:,2)+noseBBBox2(:,4)/2]
hold on
plot(srodekTwarzy(1),srodekTwarzy(2),'ro')
plot(srodekNosa(:,1),srodekNosa(:,2),'go')

proporcjaObszaruTwarzyiNosa = 0.3;
poleTwarzy = bboxes(1,3).*bboxes(1,4)
poleNosa = noseBBBox2(:,3).*noseBBBox2(:,4)
nos_warunek_1 = poleNosa > proporcjaObszaruTwarzyiNosa*poleTwarzy;

noseBBBox2A = noseBBBox2(nos_warunek_1,:);
RGB2 = faceIM;
RGB2 = insertObjectAnnotation(RGB2, 'rectangle', noseBBBox2A, 'nos');
subplot(1,2,2); imshow(RGB2)
```

Część II – Śledzenie twarzy.

Uwagi do algorytmu śledzenia twarzy:

- Do śledzenia wykorzystywane są cechy wykryte przy pomocy `detectMinEigenFeatures`, dzięki temu nie jest konieczna detekcja twarzy na każdej ramce (co przyspiesza działanie algorytmu). Pomiędzy kolejnymi ramkami śledzenie odbywa się przy pomocy obiektu `vision.PointTracker`
- Jeśli liczba śledzonych cech jest mniejsza niż próg, odbywa się inicjalizacja lub re-inicjalizacja trackera czyli ponowna detekcja twarzy przy pomocy `vision.CascadeObjectDetector` i ponowne wykrywanie punktów charakterystycznych `detectMinEigenFeatures` w obszarze twarzy
- Jeśli liczba śledzonych cech jest odpowiednia, algorytm estymuje położenie tych cech na kolejnej ramce. Dodatkowo wyznaczana jest transformacja ROI (obszaru) twarzy do nowego położenia na podstawie obliczonej transformacji geometrycznej punktów pomiędzy kolejnymi ramkami.

Instrukcja:

1. Nagraj sekwencję video zawierającą twarz oraz ruchy głowy (np. przesuwanie, przechylenie, oddalanie i przybliżanie).
2. Zbadaj w jakich warunkach twarz nie jest poprawnie wykrywana i śledzona. Zapisz i umieść w sprawozdaniu przykładowe obrazy takich przypadków.
3. Sprawdź co się stanie jeśli (podczas śledzenia) nie zostaną wykryte punkty charakterystyczne (np. przysłoń twarz ręką). W jaki sposób można temu zaradzić?
4. Zbadaj dokładność detekcji atrybutów twarzy (usta, nos) dla różnych warunków oraz parametrów (**UZUPEŁNIJ_1 w testHarness_TRACK.m**). Zapisz i umieść w sprawozdaniu przykładowe obrazy ciekawych przypadków.
5. Opcjonalnie rozwiń sposób eliminacji fałszywych/wielokrotnych detekcji atrybutów twarzy (usta, nos). Uzupełnij/zmień kod funkcji **weryfikacja1.m** i załącz do sprawozdania ten fragment.

Część III – Biblioteka do rozpoznawania mimiki twarzy.

Gdzie szukać dodatkowych informacji:

- Internet – słowa kluczowe: face detection and tracking
- http://en.wikipedia.org/wiki/Facial_Action_Coding_System
- System pomocy środowiska MATLAB
- <https://github.com/TadasBaltrusaitis/OpenFace/wiki>

Zapoznaj się z dokumentacją oraz opisem biblioteki OpenFace. Ze względu na duży stopień skomplikowania instalacji i kompilacji biblioteki ta część ćwiczenia jest teoretyczna. Możesz jednak spróbować zainstalować i skompilować bibliotekę i uruchomić przykłady (opcjonalnie).

Zwróć uwagę na:

- Funkcjonalność biblioteki (jakie informacje pozwala wyodrębnić z obrazu twarzy)
- Co to jest “Point Distribution Model”?
- Co to jest Facial Action Coding System? Jak jednostki czynnościowe AU są powiązane z mimiką twarzy?

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 przykładowe rezultaty detekcji twarzy oraz jej cech dla przynajmniej 3 obrazów oraz obrazu testowy_0_0000.jpeg. Zannotuj liczbę wykrytych punktów. 2) W sprawozdaniu zamieść przykładowe rezultaty detekcji oraz liczbę wykrytych atrybutów twarzy (nos, usta) dla przynajmniej 3 własnych obrazów. 3) Cz.I: Zamieść fragmenty kodu uzupełnione w trakcie ćwiczenia (algorytm weryfikacji detekcji cech twarzy – oczy i usta). 4) Cz.II: Umieść w sprawozdaniu przykładowe obrazy przypadków w których twarz nie jest poprawnie wykrywana. 5) Cz.II: Umieść w sprawozdaniu przykładowe obrazy przypadków w których usta i nos nie są poprawnie wykrywane. 6) Cz.II: Opc. zamieść fragmenty kodu uzupełnione w trakcie ćwiczenia.
<u>Analiza i wnioski</u> 1) Cz.I: Czy wszystkie punkty charakterystyczne (<code>detectMinEigenFeatures</code>), które zostały wykryte w obszarze ROI twarzy odpowiadają cechom związanym z budową morfologiczną twarzy – atrybuty stałe (brwi, usta, itp.)? W jakich miejscach koncentrują się wykryte cechy? 2) Cz.I: Czy detektory atrybutów twarzy (usta, nos) wykrywają poprawnie te elementy twarzy? Jeśli nie to zastanów się dlaczego tak jest. 3) Cz.I: Jakie inne sposoby weryfikacji wykrytych atrybutów morfologicznych twarzy możesz zaproponować? 4) Cz.II: Zannotuj wnioski z działania algorytmu śledzenia twarzy i wykrywania jej cech. Zaproponuj sposoby poradzenia sobie z przypadkami błędnych detekcji.
<u>Pytania</u> 1. Zapoznaj się z dokumentacją i opisem biblioteki OpenFace pozwalającej na detekcję i śledzenie twarzy oraz jej cech i rozpoznawanie jednostek czynnościowych twarzy: https://github.com/TadasBaltrusaitis/OpenFace/wiki Zwróć uwagę na: <ul style="list-style-type: none"> • Funkcjonalność biblioteki (jakie informacje pozwala wyodrębnić z obrazu twarzy) • "Point Distribution Model" • Facial Action Coding System oraz powiązanie jednostek czynnościowych AU z mimiką twarzy 2. Wymień i krótko uzasadnij 3 elementy z biblioteki OpenFace, które zwróciły Twoją uwagę lub są Twoim zdaniem istotne do zrozumienia działania opisanych algorytmów. 3. Zadaż 1 pytanie związane z ćwiczeniem lub z ww biblioteką - co jest dla Ciebie niezrozumiałe lub niejasne.