

OBLICZANIE KĄTA SKRĘTU KIEROWNICY SAMOCHODOWEJ

Bartosz Humel¹, Piotr Spętany²

1. nr albumu: 235876 235876@edu.p.lodz.pl
2. nr albumu: 235979 235979@edu.p.lodz.pl

Słowa kluczowe: kąt skreću kierownicy samochodowej, śledzenie punktu, obrót kierownicy samochodowej

1. WSTĘP

Celem naszego projektu jest obliczanie kąta skrętu kierownicy samochodowej na podstawie nagrania wideo. Jest to ciekawe zagadnienie ze względu na możliwość zastosowania go w automatycznej jeździe samochodem i w badaniu zachowania kierowców i autonomicznych samochodów.

W Internecie nie ma dostępu do wielu podobnych do naszego projektu, jednakże jeśli chodzi o tę zbliżoną tematykę można wyszczególnić następujące dwa:

Steering wheel autorstwa Aviral09[1]

Jest to stosunkowo prosty program pozwalający stwierdzić czy kierownica kręci się w lewo czy w prawo.

Kolejnym projektem[2], który można byłoby przytoczyć jest śledzenie obiektów na wideo. Odbiega nieco od tematyki kierownic, ale bazuje na jednym z kluczowych elementów, który również jest zawarty w naszym projekcie czyli śledzenie pewnych punktów na obrazie wideo.

Nasza metoda ma na celu śledzenie konkretnego punktu na kierownicy samochodowej i określenia dokładnego kąta skrętu.

2. MATERIAŁY I METODY

2.1. Dane

Użyliśmy nagrań kierownicy podczas parkowania samochodów autonomicznych[3] i przechwyconego ekranu podczas korzystania z symulatorów jazdy samochodem. (Rysunek 1, Rysunek 2)

Nagrania były barwne w rozdzielczości 1280 na 720 pikseli i posiadające 30 klatek na sekundę nagrania.

Nagrania zostały skrócone do momentu gdzie widoczny jest skręt kierownicy.



Rys1. Obraz z nagrania.



Rys2. Przechwycony obraz z symulatora.

2.2. Metody

Na początku zmieniliśmy obraz z barwnego na obraz w skali szarości używając funkcji `cvtColor[4]` (Rysunek 3). Następnie zmodyfikowaliśmy obraz za pomocą filtra Gaussa w celu zamazania nie potrzebnych detali używając jądra o rozmiarze 9×9 w funkcji `cv2.GaussianBlur[5]`. (Rysunek 4)



Rys3. Obraz w odcieniach szarości.



Rys4. Obraz rozmyty filtrem Gaussa.

Kolejnym krokiem było znalezienie okręgów na zmodyfikowanym materiale wideo przy pomocy funkcji `Cv2.HoughCircles[6]`. Parametr odpowiadający wyższemu progowi z dwóch przekazywanych do detektora krawędzi Canny'ego został ustawiony na 150 a odległość pomiędzy znalezionymi okręgami na 5. (Rysunek 5)

Zakładamy, że największy znaleziony okrąg jest kierownicą.



Rys 5. Największy znaleziony okrąg uznany za kierownicę.

Korzystając z wbudowanego narzędzia do śledzenia w bibliotece `cv2` o nazwie `TrackerCSRT[7]` wyznaczamy charakterystyczny punkt na kierownicy do śledzenia. (Rysunek 6)



Rys.6 Zaznaczony charakterystyczny punkt na kierownicy.

W wypadku niepowodzenia wbudowanego śledzenia opracowaliśmy alternatywną metodę. Używając funkcji `cv2.absdiff[8]` obliczamy różnice pomiędzy aktualną klatką nagrania a poprzednią. Na obliczonej macierzy różnic użyliśmy dylatacji obrazu używając jako elementu strukturalnego macierz jedynek o rozmiarach 5×5 .

Po wykonaniu progowania binarnego na obrazie o progu 20 używając funkcji `cv2.threshold[9]` wykrywamy kontury na binarnym obrazie używając funkcji `cv2.findContours[10]` korzystając z metody `cv2.CHAIN_APPROX_SIMPLE[11]` w trybie `cv2.RETR_EXTERNAL[12]`. (Rysunek 7)



Rys7. Zaznaczone kontury obszarów zmienionych w obrębie kierownicy.

W przypadku gdy śledzenie przy pomocy wbudowanego trackera się nie powiedzie znajdujemy przy pomocy odległości euklidesowej najbliższy położony kontur w obrębie kierownicy w którym był wykonanych ruch i ustawiamy położenie punktu śledzonego jako współrzędne tego konturu.

Przy pomocy środka okręgu wyznaczonego jako kierownicy, punktu początkowego śledzenia i aktualnego punktu śledzenia wyznaczamy kąt skrętu kierownicy przy pomocy wzoru:

$$\text{atan2}(cy - by, cx - bx) - \text{atan2}(ay - by, ax - bx)$$

Gdzie: cy, cx – współrzędne punktu śledzonego, by, bx – współrzędne środka kierownicy, ay, ax – współrzędne punktu początkowego, a atan2 – arcus tangens określonych współrzędnych[13].

Wynik równania przekształcamy z radianów na stopnie używając funkcji `math.degrees[14]`.

Finalnie nanosimy linie od środka kierownicy do aktualnego punktu śledzenia na obraz wyjściowy przy pomocy funkcji cv2.line i wyświetlamy obliczony kąt skrętu.(Rysunek8, Rysunek 9)



Rys8. Wynik obliczony na materiale rzeczywistym.



Rys9. Wynik obliczony na materiale przechwyconym.

2.3. Technologie i narzędzia

Podczas realizacji projektu korzystaliśmy z języka python w wersji 3.9 i bibliotek cv2, numpy, time oraz math.

Komputery na których projekt był testowany działały na systemie Windows, 32GB RAMu z procesorem AMD Ryzen 7 4800H. Repozytorium projektu znajduje się na github[15].

2.4. Eksperymenty

W celu poprawy działania projektu wprowadzaliśmy zmiany w parametrach używanych funkcji. Główne zmiany dotyczyły stopnia rozmycia obrazu przy pomocy filtru Gaussa oraz wykrywanie okręgu przy pomocy funkcji HoughCircles. Eksperymentowaliśmy również z metodą obliczania kąta na podstawie trzech otrzymanych punktów, tak aby projekt działał dla skrętów powyżej 90° i 180°.

3. REZULTATY

Rezultaty są prawidłowe dla kątów powyżej 180°(Rysunek 10), również jak dla materiałów przechwyconych z symulatora jazdy samochodowej (Rysunek 11).



Rys 10. Kąt powyżej 180°.



Rys 11. Wynik z symulatora jazdy.

Projekt nie działa dla niektórych nagrań wideo[16], z powodu braku wykrycia okręgu i określenia go jako kierownicy(Rysunek 12).



Rys 12. Błędne działanie programu z powodu nie wykrycia kierownicy

4. DYSKUSJA

Nasze rozwiązanie wyznacza z dużą dokładnością kąt skrętu kierownicy samochodowej w sytuacji kiedy nagranie kierownicy jest wykonane w warunkach takich jak: kierownica nagrana jest prostopadłe, w dobrym oświetleniu, posiada punkt charakterystyczny który można śledzić oraz obrót kierownicy nie jest zbyt szybki. W innych przypadkach kierownica może nie zostać wykryta lub punkt śledzenia może zostać zgubiony co poskutkuje w spadku dokładności wyznaczonego kąta.

5. PODSUMOWANIE I WNIOSKI

Nasz projekt działa poprawnie dla ściśle określonych nagrań kierownicy, dalszy rozwój projektu mógłby poszerzyć zakres nagrań o takie gdzie kamera nie znajduje się naprzeciwko kierownicy. W założeniu naszego projektu chcieliśmy by kąt był śledzony automatycznie bez potrzeby wyznaczania najpierw punktu charakterystycznego ale nie udało nam się tego zrealizować, pozostaje to potencjalną możliwością rozwoju tego projektu. Pomimo tych wad nasze rozwiązanie oferuje dokładne wyznaczenie kąta skrętu kierownicy i może zostać użyte do badania zachowań samochodów autonomicznych lub do analizy symulacji jazdy samochodem.

6. BIBLIOGRAFIA

1. <https://github.com/Aviral09/SteeringWheel>
2. <https://pysource.com/2021/01/28/object-tracking-with-opencv-and-python/>
3. <https://www.youtube.com/watch?v=N-9JEUFXvww>
4. https://docs.opencv.org/3.4/d8/d01/group_imgproc_color_conversions.html#ga397ae87e1288a81d2363b61574eb8cab
5. https://docs.opencv.org/3.4/d4/d86/group_imgproc_filter.html#gaabe8c836e97159a9193fb0b11ac52cf1
6. https://docs.opencv.org/4.x/dd/d1a/group_imgproc_feature.html#ga47849c3be0d0406ad3ca45db65a25d2d
7. https://docs.opencv.org/4.x/d7/d8f/classcv_1_1legacy_1_1TrackerCSRT.html
8. https://docs.opencv.org/3.4/d2/de8/group_core_array.html#ga6fef31bc8c4071cbc114a758a2b79c14
9. https://docs.opencv.org/4.x/d7/d1b/group_imgproc_misc.html#gae8a4a146d1ca78c626a53577199e9c57
10. https://docs.opencv.org/3.4/d3/dc0/group_imgproc_shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a
11. https://docs.opencv.org/4.x/d3/dc0/group_imgproc_shape.html#ga4303f45752694956374734a03c54d5ff
12. https://docs.opencv.org/4.x/d3/dc0/group_imgproc_shape.html#ga819779b9857cc2f8601e6526a3a5bc71
13. <https://en.wikipedia.org/wiki/Atan2>
14. <https://docs.python.org/3/library/math.html#math.degrees>
15. <https://github.com/Bartoszhumel/PSIO.kierownica>
16. <https://www.youtube.com/shorts/a6s-S8da64l>