

PRACA PRZEJŚCIOWA INŻYNIERSKA

PROJEKT SYSTEMU UTRZYMANIA PASA RUCHU

SEMESTR LETNI 2019

Adam Napieralski

285731

Spis treści

1 Wstęp	3
1.1 Przegląd rozwiązań	3
1.2 Cel pracy	4
2 Podstawy teoretyczne	5
2.1 Kalibracja kamery	5
2.2 Transformacja obrazu	5
2.3 Modele przestrzeni barw	5
3 Wykorzystane narzędzia	7
4 Schemat stworzonego algorytmu	8
4.1 Kalibracja kamery	8
4.2 Transformacja perspektywy	9
4.3 Progowanie warstw modeli przestrzeni barw	11
4.4 Detekcja linii	16
4.5 Określenie wysunięcia z pasa oraz promienia krzywizny	17
4.6 Projektacja na obraz źródłowy	19
5 Wnioski	20
6 Literatura	21

1 WSTĘP

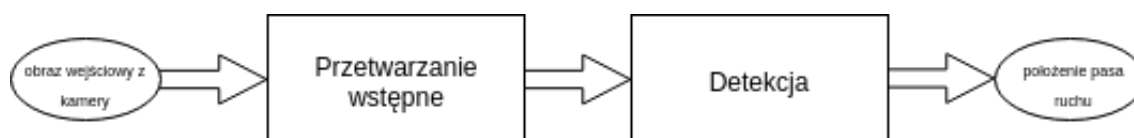
Systemy detekcji i utrzymania pasa ruchu stają się powoli standardem w nowo produkowanych samochodach - zwiększają znacznie bezpieczeństwo na drodze, szczególnie w skrajnych sytuacjach takich jak chwilowy brak uwagi kierowcy czy nawet jego przyśnięcie, kiedy to po wykryciu opuszczania pasa ruchu system daje sygnał (wibracyjny lub dźwiękowy) lub nawet na chwilę przejmując kontrolę nad układem kierowniczym i delikatnie naprowadza samochód z powrotem na pas.

Mimo że istniejące obecnie systemy są stworzone głównie do pełnienia roli awaryjnego asystenta, to autonomia w pojazdach coraz bardziej się rozwija i powoli staje się rzeczywistością - najbardziej rozpowszechnionym przykładem może być tu autopilot w samochodach Tesla.

Tym samym rośnie też odpowiedzialność spoczywająca na systemach rozpoznania pasa ruchu, przez co tak ważne jest zaprojektowanie ich by działały w sposób sprawny i niezawodny.

1.1 Przegląd rozwiązań

Mimo mnogości metod podejścia do zagadnienia systemu detekcji pasa ruchu, po przejrzeniu istniejących rozwiązań można zauważyć, że praktycznie każdy z nich da się podzielić na dwa ogólne kroki: przetwarzanie wstępne i detekcja.[1]



Rysunek 1: Schemat ogólnego procesu systemu

Przetwarzanie wstępne

Przetwarzanie wstępne ma na celu poddanie obrazu wejściowego z kamery takim zmianom i działaniom, by uzyskać dane odpowiednie do używanych metod analizy, której wynikiem będzie detekcja pasa ruchu.

Z reguły dane takie zawierają wyselekcjonowane różnymi metodami punkty (piksele), które tworzą pasy.

Jedną z metod wykrywania krawędzi pasów jest wykorzystanie operatora Sobela. Rozwiązanie to dla zagadnienia detekcji pasa ruchu zaproponował już w 2003 Y.U. Yim i in.[2], potem wykorzystali je również w swoich pracach Y.-C. Leng i in. [3]

Metodą o krok bardziej złożoną, lecz nadal zmierzającą do wyznaczenia krawędzi pasów jest zastosowanie wielostopniowego algorytmu zwanego Canny (pochodzącą od jego autora - Johna Canny'ego[4]). Korzystał z niego m.in. A.Assidiq i in.[5] w pracy dotyczącej detekcji pasa ruchu przeznaczonej konkretnie dla samochodów autonomicznych.

Popularnym i dość prostym rozwiązaniem jest natomiast skorzystanie z metody progowania obrazu,

dzięki któremu otrzymujemy jego binarną kopię z wyselekcjonowanymi pod względem wartości koloru pikselami. Została ona użyta w pracy J.Wanga i in.[6]

Wykorzystał on również na tak obrobionym obrazie transformację perspektywy do widoku z lotu ptaka, co ułatwia dalszy proces detekcji i jego możliwości, przez co metoda ta pojawia się często w innych pracach - m.in. u S. Sehesteda i in.[8]

Detekcja

Na etapie detekcji wykorzystuje się przetworzony wstępnie obraz (najczęściej zredukowany już do formy binarnej) i wykorzystując wybrane algorytmy definiuje się obszar, położenie czy inne charakterystyki pasa (jak chociażby krzywizna), a także położenie samochodu względem niego.

Z racji dość przewidywalnych i regularnych kształtów pasów jedną z częściej stosowanych metod rozpoznawania ich położenia jest transformacja Hougha[1] - wykorzystano ją w kilku spośród wyżej wspomnianych prac[2][3][5][7].

Z. Kim [9] opisuje natomiast metodę korzystającą z sieci neuronowych, a dynamiczny rozwój uczenia maszynowego może sugerować, że takie rozwiązania będą pojawiały się coraz częściej.

Aproksymacja krzywą mająca na celu dopasowanie i opisanie znajdujących pasów funkcją ciągłą jest też często spotykanym krokiem podczas detekcji, gdyż w prosty matematyczny sposób definiuje przebieg pasa. Zastosował ją m.in. H. Yoo i in.[10] korzystając z funkcji kwadratowej.

1.2 Cel pracy

Na podstawie przeglądu stosowanych i opisanych rozwiązań systemów detekcji i utrzymania pasa ruchu można stwierdzić, że jednym z prostszych, aczkolwiek skutecznych rozwiązań zdaje się być przetwarzanie wstępne na bazie progowania wartości modelów kolorów obrazu, po którym korzystać się będzie z aproksymacji pasów krzywą.

Dlatego celem niniejszej pracy jest stworzenie działającego algorytmu detekcji pasa ruchu z wykorzystaniem wyżej wymienionych metod i określenie właściwego doboru warstw modeli kolorów wraz z parametrami progowania.

2 PODSTAWY TEORETYCZNE

2.1 Kalibracja kamery

Obrazy z kamery mogą być narażone na wpływ wad wykorzystanych w nich układów optycznych, co może przejawiać się *dystorsją* takiego obrazu. Głównym skutkiem tego zaburzenia jest brak zachowania liniowości rzeczywistego elementu z jego odwzorowaniem na obrazie.

Dystorsja najczęściej przejawia się w dwóch typach:

- dystorsja beczkowa
- dystorsja poduszkowa

W przypadku przetwarzania obrazu, którego istotą jest analiza rzeczywistych elementów i wykonanie na nich pomiarów, tak jak ma to miejsce w przypadku systemów wykrywania i utrzymania pasa ruchu, konieczne jest jak największe zniwelowanie takich zniekształceń.

Dokonyuje się tego przez opisanie modelu kamery zbiorem parametrów łączących rzeczywiste współrzędne punktów w 3D z odpowiadającymi im punktami na obrazie uwzględniających zniekształcenia.

2.2 Transformacja obrazu

Transformacje afiniczne pozwalają na przekształcenie obrazu w sposób, który umożliwia "sztuczne" uzyskanie m.in. innej perspektywy niż obecna na pierwotnym obrazie.

Istotne w ich wypadku jest zachowanie liniowości, równoległości oraz odwzorowywanie prostych i płaszczyzn kosztem utraty odwzorowania kątów i odległości między punktami.

Przekształcenie to może być reprezentowane przez przekształcenie liniowe:

$$x' = M \cdot x$$

gdzie:

- x - wektor współrzędnych początkowych
- M - macierz przekształcenia
- x' - wektor współrzędnych docelowych

Stąd uzyskanie odpowiedniego przekształcenia sprowadza się tylko do wyznaczenia jego macierzy.

2.3 Modele przestrzeni barw

Aby opisać kolor możemy posłużyć się różnymi modelami przestrzeni barw. Każdy z nich charakteryzuje się innym matematycznym systemem opisu koloru w przestrzeni parametrów, w specyficznym układzie współrzędnych. Najpopularniejszymi modelami są:

- RGB
Opisany 3 składowymi: R - *red* - czerwonej; G - *green* - zielonej; B - *blue* - niebieskiej
Popularny model z racji szerokiego wykorzystywania w urządzeniach analizujących obraz (np. aparaty cyfrowe, skanery) oraz w urządzeniach wyświetlających obraz (np. telewizory, monitory komputerowe).[11]
- HSV
Opisany 3 składowymi: H - *hue* - odcień; S - *saturation* - nasycenie; V - *value* - moc / jasność
- HSL
Opisany 3 składowymi: H - *hue* - odcień; S - *saturation* - nasycenie; L - *lightness* - średnie światło białe
- Lab (CIELab)
Opisany 3 składowymi: L - *lightness* - jasność / luminancja; a - barwa od zielonej do magenty; b - barwa od niebieskiej do żółtej
- LUV (CIELUV)
Opisany 3 składowymi: L - *lightness* - jasność / luminancja; U, V - odpowiadają za chromatyczność koloru

3 WYKORZYSTANE NARZĘDZIA

Materiałami źródłowymi były nagrania własne z kamery w telefonie komórkowym, co umożliwiało uprzednie określenie parametrów używanej kamery do jej kalibracji. Domyślną rozdzielczością było 1280x720.

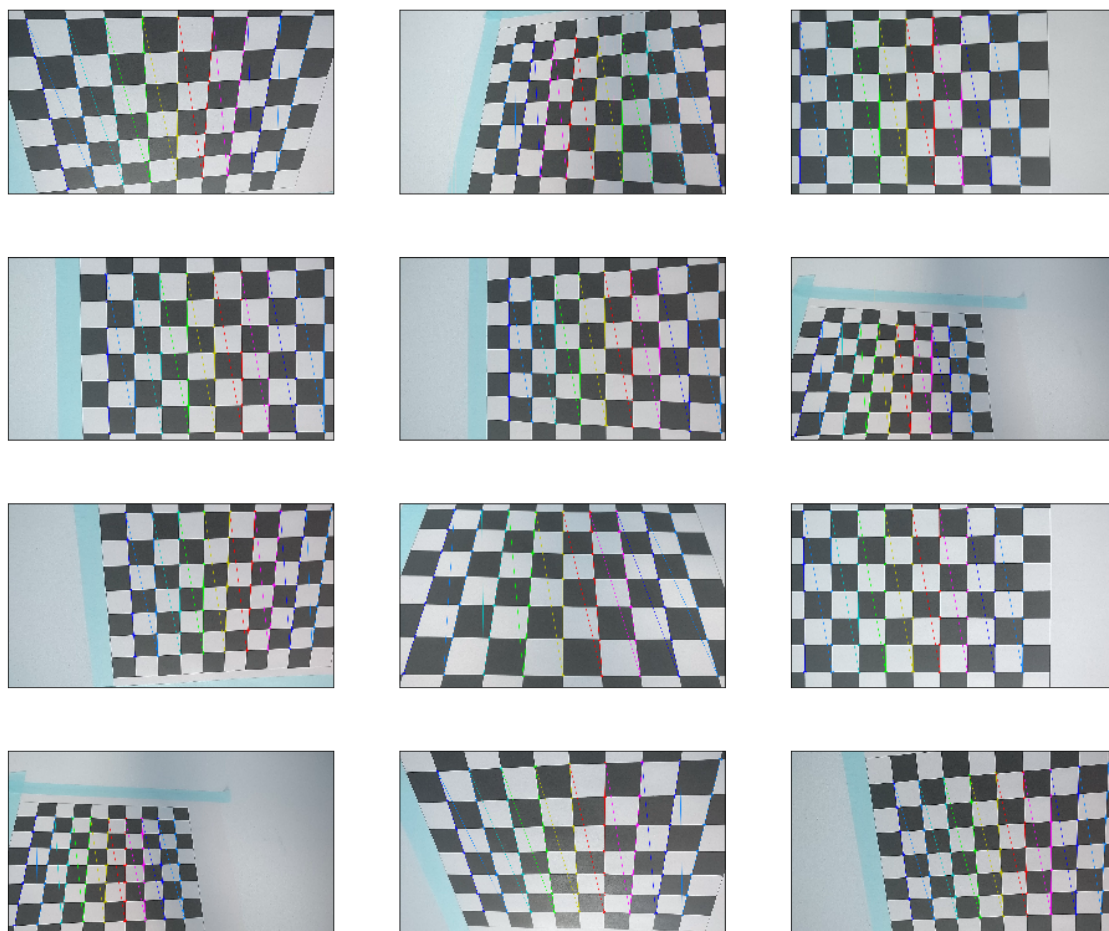
Kod pisany był w języku Python w wirtualnym środowisku korzystającym z dystrybucji *Anaconda*, zapewniającej szeroki zbiór pakietów przydatnych do przetwarzania danych, w tym bibliotekę *OpenCV*, zapewniającą funkcje obróbki obrazów oraz *numpy* wspierającą ją pod względem obsługi działań na tablicach i macierzach. Skrypt pisany był z wykorzystaniem *Jupyter Notebook*.

4 SCHEMAT STWORZONEGO ALGORYTMU

4.1 Kalibracja kamery

Aby wyznaczyć parametry kamery - jej macierz oraz wektor współczynników zniekształceń - skorzystano z funkcji biblioteki *OpenCV* - *calibrateCamera*. Jako argumentów potrzebuje ona tablicy punktów rzeczywistych oraz tablicy punktów reprezentujących je na obrazie.

Do uzyskania takich danych użyty został wzór szachownicy, której rzeczywisty rozkład punktów był ręcznie zdefiniowany, natomiast ich odwzorowania na obrazie znajdowane były specjalną funkcją z biblioteki - *findChessboardCorners*. Do jak najlepszego określenia parametrów kamery konieczne było dostarczenie wystarczającej ilości różnorodnych danych - zrobione zostały zdjęcia szachownicy z różnych perspektyw.



Rysunek 2: Zdjęcia szachownic z wykrytymi punktami do kalibracji kamery

Po zastosowaniu funkcji kalibracyjnej otrzymano:

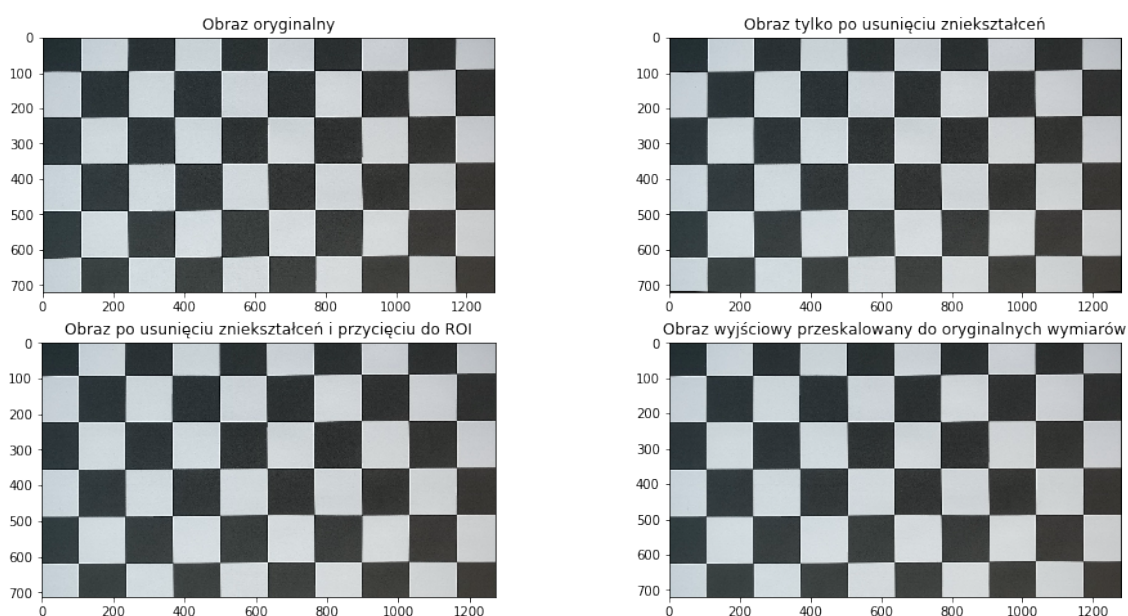
- macierz kamery
$$\begin{bmatrix} 1035.41773 & 0.0 & 647.490892 \\ 0.0 & 1028.48296e & 349.379086 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$
- wektor współczynników zniekształceń
$$\begin{bmatrix} -0.01724175 & 0.01293847 & -0.00040578 & -0.00083023 & 0.08450202 \end{bmatrix}$$

oraz dodatkowo

- obszar obrazu wolny od poprawek wynikających z usuwania zniekształceń - *region of interest (ROI)*

Zdefiniowany przez lewy górny róg prostokąta oraz jego wymiary: szerokość i wysokość:

(4,3,1273,713)



Rysunek 3: Przegląd kolejnych kroków w przygotowywaniu obrazu z usuniętymi zniekształceniami

Można zauważyć, że zmiany wprowadzone przez usunięcie zniekształceń nie są wyraźnie zauważalne, z czego wynika, że użyta kamera i przetwarzanie już na poziomie używanego telefonu komórkowego zniwelowały zniekształcenia na już wysokim poziomie.

4.2 Transformacja perspektywy

Następnym krokiem była zmiana perspektywy na tzw. z lotu ptaka. Dokonano tego opisując za pomocą trapezu bliższą kamerze część pasa na podstawie zdjęć źródłowych i definiując prostokąt, który



Rysunek 4: Obrazy przed usunięciem zniekształceń kamery (po lewej) i po (po prawej)

odpowiadałby widokowi pasa z góry.



Rysunek 5: Opisanie pasa czworokątami do transformacji perspektywy

Korzystając w odpowiedniej funkcji z biblioteki na podstawie czterech punktów źródłowych i docelowych tych wielokątów wyznaczono macierz transformacji obrazu, która służyła następnie do zmiany perspektywy obrazów źródłowych.

Otrzymana macierz transformacji:

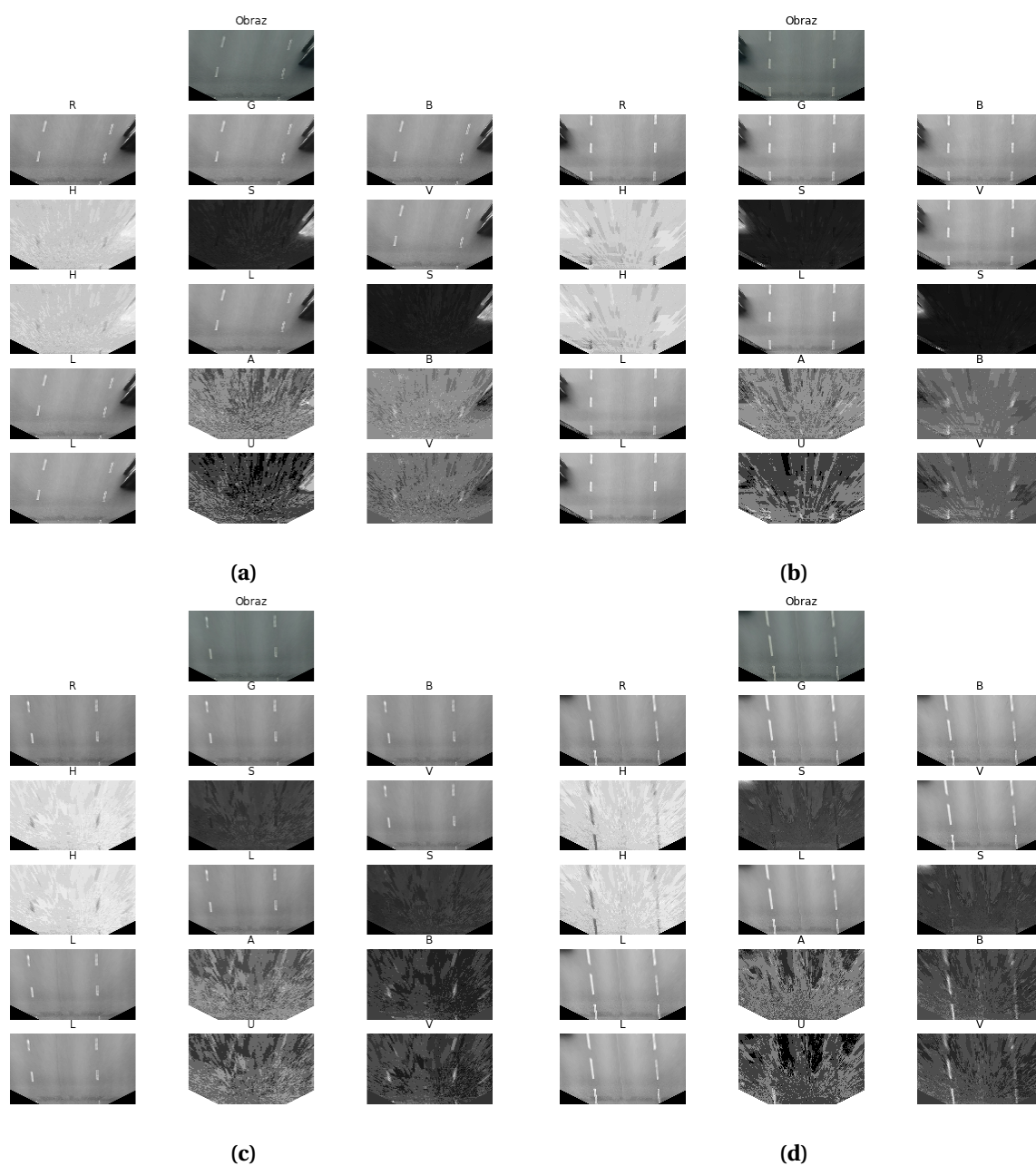
$$\begin{bmatrix} -6.75818374e-01 & -2.05561422e+00 & 1.09246040e+03 \\ 7.47939722e-16 & -2.79948375e+00 & 1.10859556e+03 \\ 1.01978938e-18 & -3.13856623e-03 & 1.00000000e+00 \end{bmatrix}$$



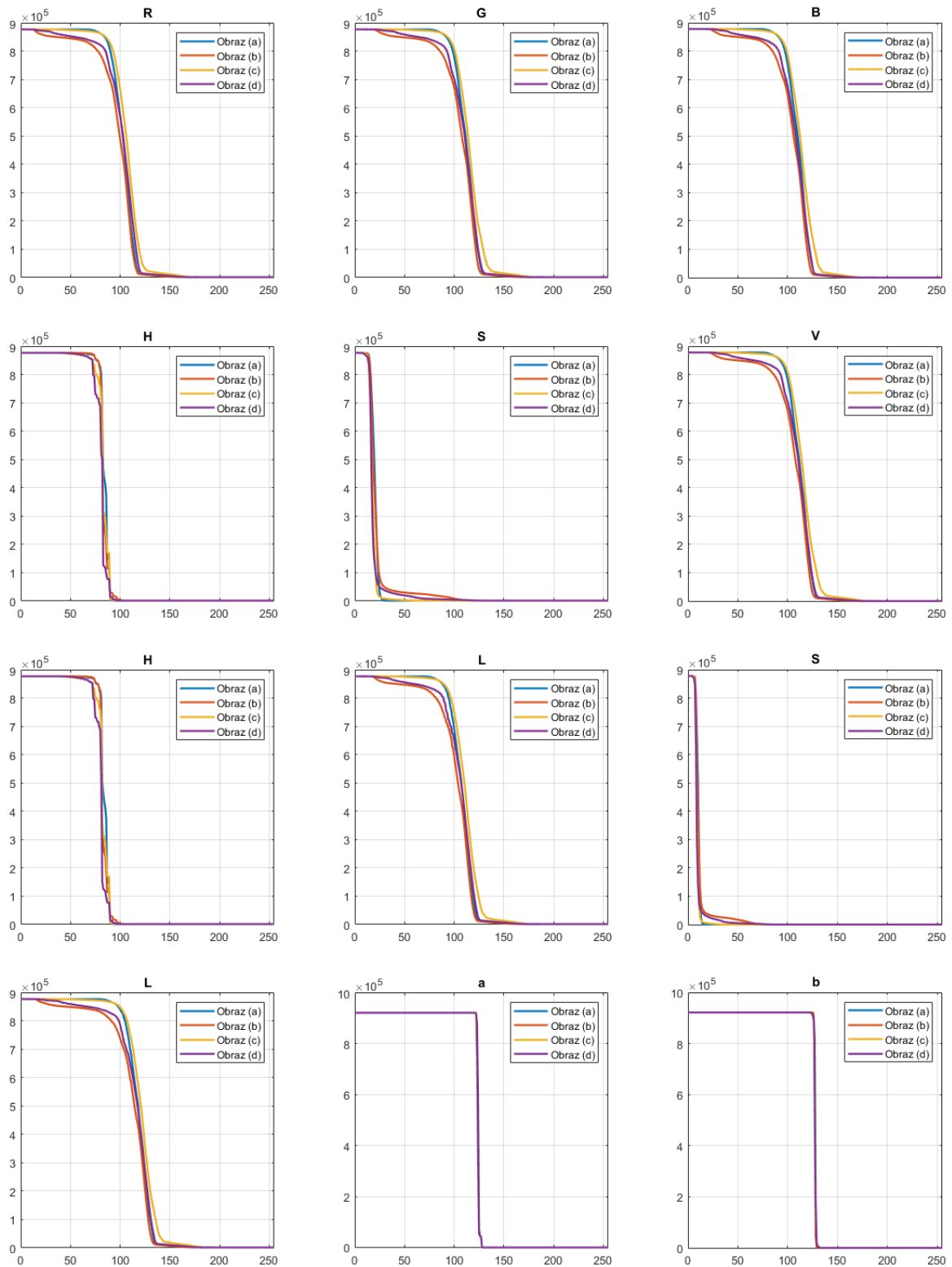
Rysunek 6: Obrazy z zastosowaniem transformacji perspektywy

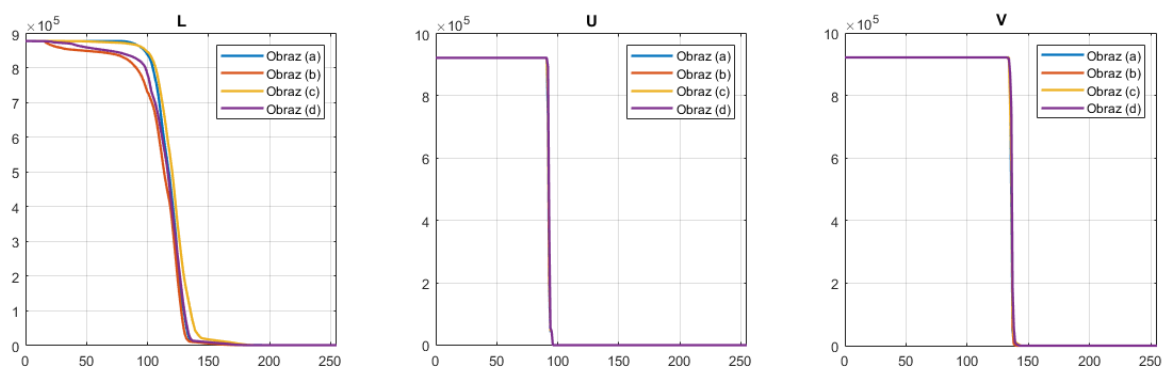
4.3 Progowanie warstw modeli przestrzeni barw

W celu progowania, które oddzieliło by pasy od otoczenia, należało zbadać zachowanie rozważanych modeli przestrzeni barw (RGB, HSV, HLS, Lab, LUV) i sposób w jaki ich warstwy reprezentują obraz pasów (po transformacji).



Rysunek 7: Reprezentacja pojedynczych warstw modeli przestrzeni barw dla obrazów



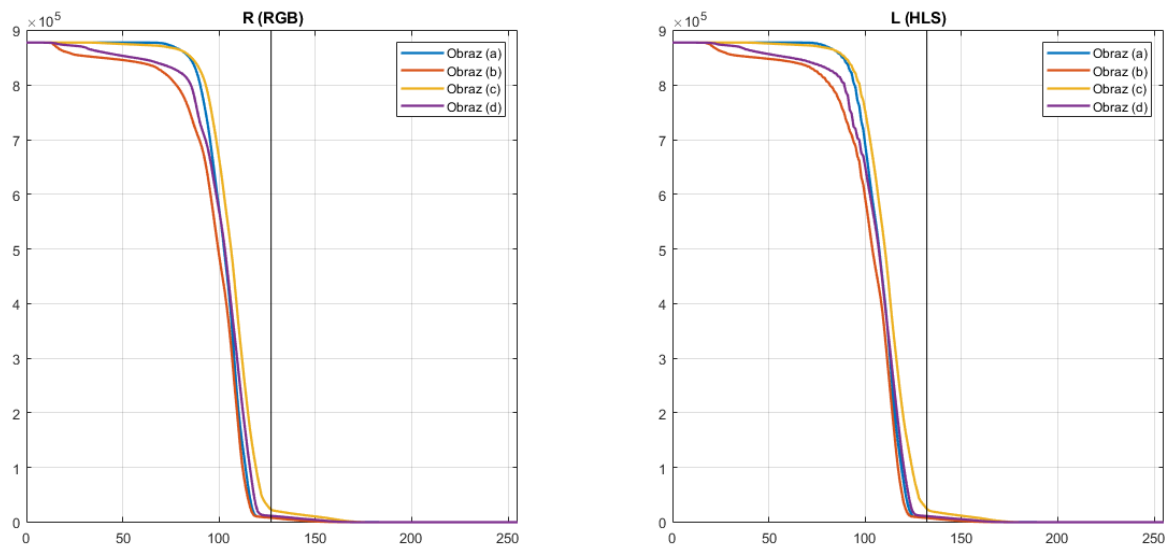


Rysunek 8: Wykresy ilości widocznych pikseli od poziomu progowania dla warstw modeli przestrzeni kolorów dla 4 próbkowych obrazów (wykonane w MATLABie)

Bazując na przedstawionych warstwach z Rys. 7 można przypuszczać, że nieprzydatne do wykrycia białych pasów mogą okazać się warstwy H i S z modelu HSV oraz HLS, oraz a i b z Lab i U, V z LUV. Potwierdza się to na wykresach z Rys. 8 ilości widocznych pikseli od punktu progowania - w przypadku a, b i U, V widać bardzo gwałtowny spadek z maksimum do zera, co uniemożliwiłoby praktyczne zadanie parametru progowania. Podobna sytuacja występuje dla wartw S z HLS i HSV, natomiast dla wartw H widać sporą nieregularność wraz z gwałtownym spadkiem przebiegów.

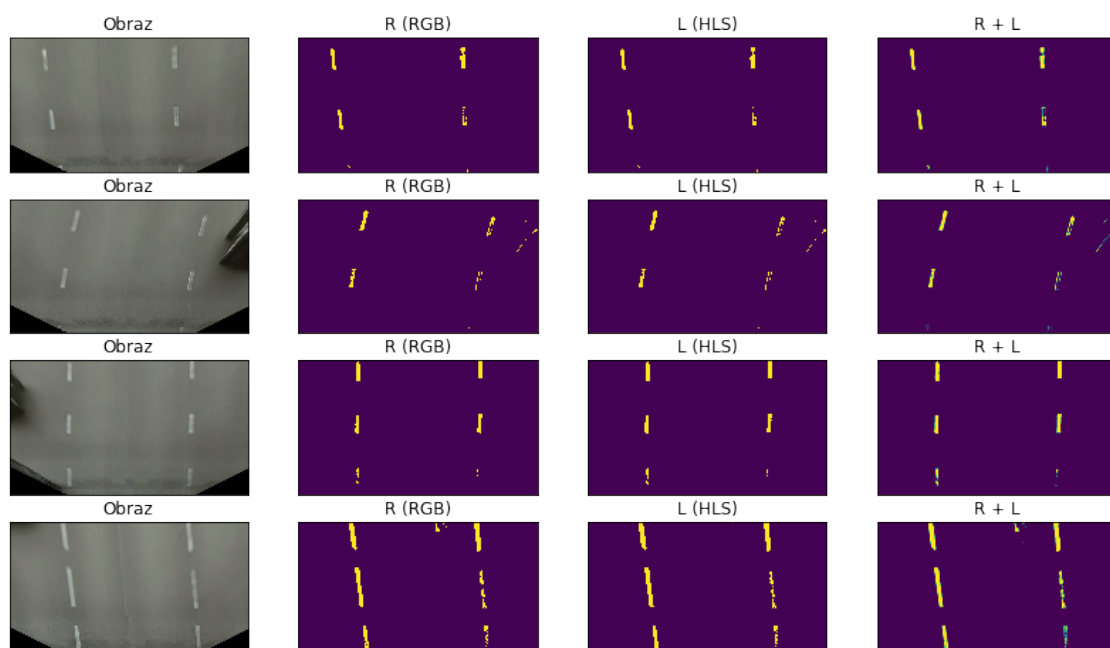
Wyniki przedstawione na wykresach z Rys. 8 dla warstw V (z HSV), L z HLS, z Lab i z LUV są do siebie bardzo zbliżone, przez co można je ze sobą utożsamiać i wybrać do dalszego przetwarzania tylko jeden.

Warstwy RGB również są zbliżone, można jednak zauważyć, że dla warstwy R serie danych dla różnych obrazów są delikatnie bardziej skondensowane. Można też przypuszczać, że kolor czerwony, będzie dużo rzadszy na jezdni (której barwy łatwiej będą podchodzić pod niebieski/zielony).



Rysunek 9: Wykresy ilości widocznych pikseli od poziomu progowania dla warstw R (RGB) i L (HLS) wybranych do dalszej analizy

Na tej podstawie do dalszej analizy oparto się na wartwie R z RGB i L z HLS. Dla R zadano parametr progowania równy 129, a dla L 132 - zaznaczone pionową linią na Rys. 9, odpowiadające momentowi wyhamowania spadku ilości widocznych pikseli na obrazach binarnych. Po zastosowaniu progowania na każdej z tych warstw połączono je w jeden obraz operacją binarną OR (Rys. 10).

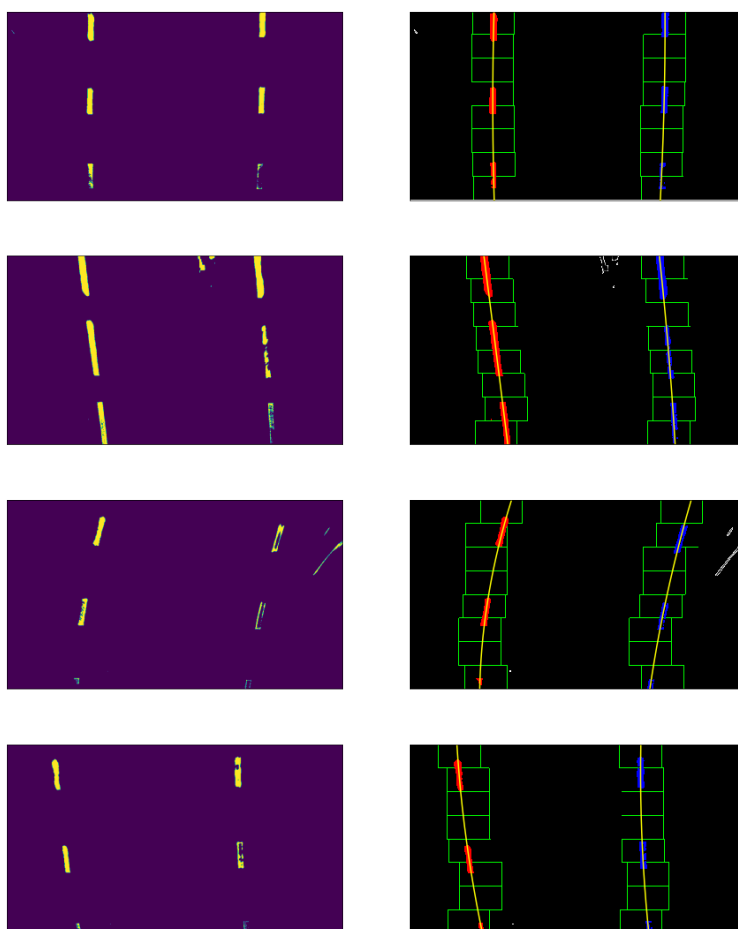


Rysunek 10: Obrazy binarne z zastosowaniem progowania - dla warstw R, L i ich połączenie

4.4 Detekcja linii

Po przetworzeniu wstępnym i otrzymaniu obrazu binarnego, do odnalezienia i wyznaczenia przebiegu pasa wykorzystano sposób oparty na krokach:

1. Wyznaczenie histogramu obrazu do $\frac{2}{3}$ wysokości i przyjęcie jego dwóch największych argumentów z osi x z lewej i prawej połowy obrazu jako przybliżeń początkowych rozpoczęcia pasa
2. Umieszczenie w tych miejscach "okienek", które sprawdzają występowanie pikseli na ich obszarze i wyznaczenie położenia kolejnego okienka na podstawie ich średniej z rozkładu ich występowania.
Powtarzanie umiejscawiania okienek nad sobą aż do górnej granicy obrazu.
3. Na podstawie położenia wszystkich pikseli z obszaru z wyznaczonych okienek aproksymuje się je funkcją wielomianową (zastosowano funkcję kwadratową).



Rysunek 11: Detekcja pasów z użyciem przesuwających się okienek z aproksymującą funkcją wielomianową

4.5 Określenie wysunięcia z pasa oraz promienia krzywizny

Przy założeniu, że widok na obrazach pochodzi z poziomu osi symetrii samochodu, w celu wyznaczenia wysunięcia samochodu względem środka pasa ruchu obliczamy różnicę położenia jego środka i środka obrazu.

$$x_{mid} = \frac{x_L - x_P}{2} \quad (1)$$

$$\Delta = x_{mid} - \frac{width}{2} \quad (2)$$

Znając funkcje wielomianowe opisujące lewy i prawy pas obliczamy ich wartości na dolnej krawędzi obrazu: x_L, x_P , a ich średnia określa środek pasa (1).

Pozycję na pasie Δ (oddalenie od jego środka) określamy jako różnicę (2).

Promień skrętu wyznaczany jest z wykorzystaniem wzoru:

$$R = \frac{[1 + (\frac{dx}{dy})^2]^{\frac{3}{2}}}{|\frac{d^2x}{dy^2}|} \quad (3)$$

gdzie używana jest pierwsza i druga pochodna funkcji aproksymującej znalezione pasy w postaci: $f = x(y)$.

Zatem:

$$f = x(y) = a \cdot y^2 + b \cdot y + c \quad (4)$$

$$x'(y) = \frac{dx}{dy} = 2a \cdot y + b \quad (5)$$

$$x''(y) = \frac{d^2x}{dy^2} = 2a \quad (6)$$

skąd mamy że:

$$R = \frac{[1 + (2ay + b)^2]^{\frac{3}{2}}}{|2a|} \quad (7)$$

Do wyznaczenia promienia krzywizny zakrętu będziemy do wzoru (7) podstawiali y dolnej granicy obrazu (aby uzyskać dane najbliższe samochodowi). Po zastosowaniu go do funkcji aproksymujących lewy i prawy pas wyznaczona zostanie średnia arytmetyczna z dwóch końcowych wartości.

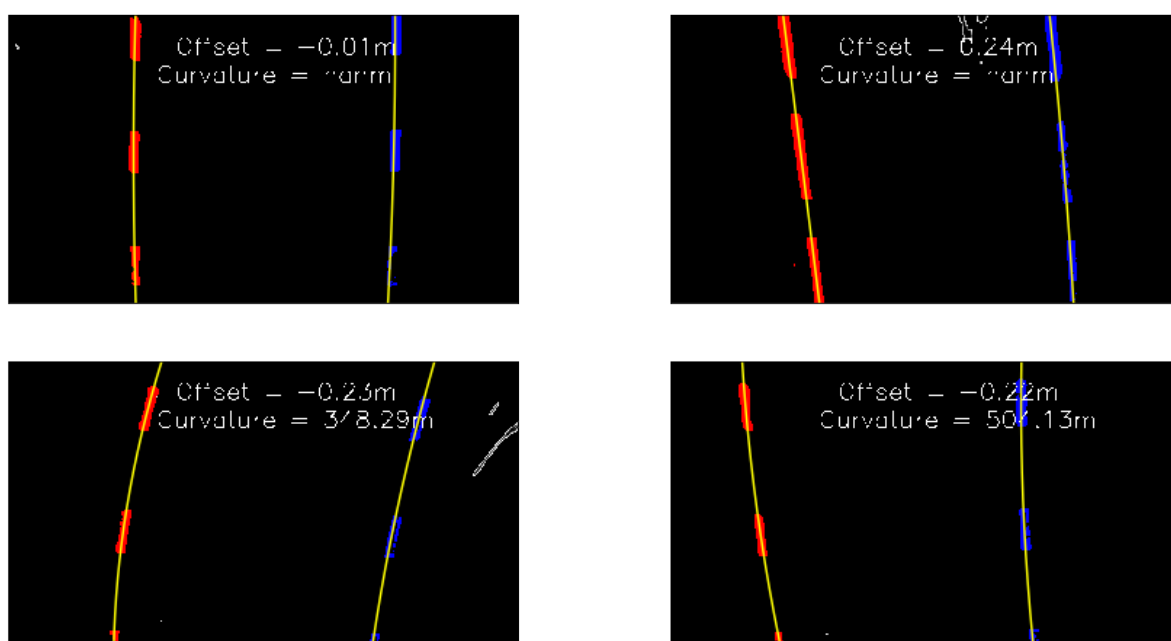
Aby otrzymywać praktyczne wyniki w obu przypadkach konieczne będzie określenie rzeczywistej odległości (w metrach) przypadającą na odległość na obrazie (osobno w osi x i y). Na podstawie długości i szerokości pasa przyjęto:

$$k_x = \frac{3m}{640px} \quad k_y = \frac{15m}{720py} \quad (8)$$

W przypadku wysunięcia, aby uzyskać rzeczywistą wartość, należy otrzymany wynik przemnożyć przez współczynnik k_x :

$$\Delta_{rz} = k_x \cdot \Delta \quad (9)$$

Dla promienia krzywizny, trzeba przemnożyć współrzędne pikseli wykorzystanych wcześniej do aproksymacji wielomianowej przez odpowiednie współczynniki i dokonać jej ponownie, a następnie podstawić do wzoru (7) podstawić również wartość y dla dolnej krawędzi obrazu przemnożoną przez k_y .



Rysunek 12: Zaznaczone przebiegi pasów wraz z obliczonym wysunięciem i promieniem krzywizny

	Wysunięcie	Promień krzywizny
Obraz (a)	$-0.01m$	∞
Obraz (b)	$0.24m$	∞
Obraz (c)	$-0.23m$	$348.29m$
Obraz (d)	$-0.22m$	$504.13m$

Tabela 1: Obliczone wysunięcia i promienie krzywizny

4.6 Projektacja na obraz źródłowy

Po wykonaniu wszystkich kroków przetworzenia i detekcji, otrzymane wyniki przetransformowane i zrzutowane zostały na obraz początkowy, dając tym samym obraz końcowy z praktycznym podglądem wszystkich danych.

Transformacja została dokonana korzystając z odwrotnej macierzy transformacji perspektywy obliczonej w kroku 4.2. Dodatkowo zakolorowany został obszar między dwiema krzywymi aproksymującymi pasy i wyświetlone informacje o wysunięciu i promieniu krzywizny.



Rysunek 13: Obrazy wyjściowe z wyświetlonymi danymi z detekcji i przetworzenia

5 WNIOSKI

Na podstawie przeprowadzonego projektu, można stwierdzić, że poprawnie działający system utrzymania pasa ruchu można zrealizować opierając się na krokach przedstawionych w pracy.

Kluczowe jest jednak odpowiednie wyselekcjonowanie pasów spoza otoczenia, które w tym wypadku znacząco opierało się na dobranych na sztywno parametrach progowania. Na podstawie wykresów z Rys. 8 i Rys. 9 można zauważyć, że granica właściwego progowania jest bardzo podatna na zmiany w charakterystyce obrazu - już przy bardzo zbliżonych obrazach wykonanych na tej samej jezdni występują drobne, lecz zauważalne różnice w przebiegach wykresów w okolicach progu. W niedalekiej odległości po lewo od progu następuje gwałtowny wzrost ilości wyświetlanych pikseli, który przy innej charakterystyce obrazu mógłby nachodzić na próg i przepuszczać piksele nienależące do pasów i zakłócające poprawne działanie detekcji.

Optymalnie należałoby wprowadzić dynamiczne ustawianie parametru progowania w zależności od charakterystyki obrazu.

6 LITERATURA

- [1] A.M.Kumar, P. Simon: *Review of lane detection and tracking algorithms in advanced driver assistance system*, International Journal of Computer Science Information Technology (IJCSIT) Vol 7, No 4, August 2015
- [2] Y.U. Yim and S.- Y. Oh, *Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving*, IEEE Trans. Intell. Transp. Syst. , vol. 4, no. 4, pp. 219-225, Dec. 2003.
- [3] Y.-C. Leng and C.-L. Chen, *Vision-based lane departure detection system in urban traffic scenes*, in Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on , 2010, pp. 1875- 1880.
- [4] J. Canny, *A Computational Approach to Edge Detection*, in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-8, NO. 6, NOVEMBER 1986
- [5] A.Assidiq, O.Khalifa, R. Islam, and S. Khan, *Real time lane detection for autonomous vehicles*, in Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on. IEEE,2008, pp. 82-88.
- [6] J.Wang, T.Mei, B. Kong, and H.Wei *An approach of lane detection based on Inverse Perspective Mapping* In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on pp. 35-38. IEEE.
- [7] G.Liu, F.Wörgötter, and I.Markelić *Combining statistical hough transform and particle filter for robust lane detection and tracking* in Proc. IEEE IV, Jun. 2010, pp. 993 – 997
- [8] S.Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, *Robust lane detection in urban environments*, in Proc. IEEE Intell. Robots Syst., Oct. 2007, pp. 123-128.
- [9] Z.Kim, *Robust lane detection and tracking in challenging scenarios*, in IEEE Trans. Intell. Transp. Syst. , vol. 9, no. 1, pp. 16-26, Mar. 2008
- [10] H.Yoo, U.Yang, and K. Sohn, *Gradient-enhancing conversion for illumination-robust lane detection*, in IEEE Transactions on Intelligent Transportation Systems , vol. 14, pp. 1083 -1094, September 2013.
- [11] Jankowski, M., and Wydawnictwa Naukowo-Techniczne, *Elementy grafiki komputerowej* Wyd. 2. ed. Kłasyka Informatyki. Warszawa: Wydawnictwa Naukowo-Techniczne, 2006.
- [12] M. Sanchez-Cuevas, R. Aguilar-Ponce, J. Tecpanecatl-Xihuitl *A Comparison of Color Models for Color Face Segmentation* Procedia Technology, Volume 7, 2013, Pages 134-141