



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA AUTOMATYKI I ROBOTYKI

Praca dyplomowa inżynierska

System wizyjny wspomagający zdalną obsługę paneli operatorskich

Vision system supporting remote usage of control panels

Autor:
Kierunek studiów:
Opiekun pracy:

Kamil Machoń
Automatyka i Robotyka
dr inż. Piotr Pawlik

Kraków, 2019

Upredzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „ Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie znieksztalcia taki utwór, artystyczne wykonanie, fonogram, videogram lub nadanie.”, a także upredzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) „Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godność studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „ sądem koleżeńskim ”, oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

.....

Spis Treści

- 1) Wstęp
- 2) Algorytmy wizji komputerowej
 - 2.1 Wstępna obróbka obrazu z kamery
 - 2.1.1 Korekta bieli
 - 2.1.2 Korekta perspektywy
 - 2.2 Detekcja i śledzenie kropki wskaźnika laserowego
 - 2.3 Wykrywanie i rozpoznawanie obiektów w scenie
 - 2.3.1 Histogram zorientowanych gradientów
 - 2.3.2 Analiza kształtu konturów
- 3) Opis manipulatora
- 4) Wstępna analiza panelu operatorskiego i otoczenia
 - 4.1 Znalezienie obszaru zainteresowania
 - 4.2 Określenie pozycji i orientacji łazika względem panelu operatorskiego
 - 4.3 Korekta obrazu
 - 4.4 Znalezienie przestrzeni roboczej manipulatora
- 5) Detekcja kluczowych elementów panelu operatorskiego
 - 5.1 wykrycie klawiatur i przycisków
 - 5.2 wykrycie innych ruchomych elementów panelu operatorskiego
- 6) Sterowanie manipulatorem z wizyjnym sprzężeniem zwrotnym
 - 6.1 Autonomiczny ruch manipulatora
 - 6.2 Obsługa przycisków i klawiatur
 - 6.3 Obsługa przełączników, zaworów, pokręteł
- 7) Podsumowanie
- 8) Bibliografia

Wstęp

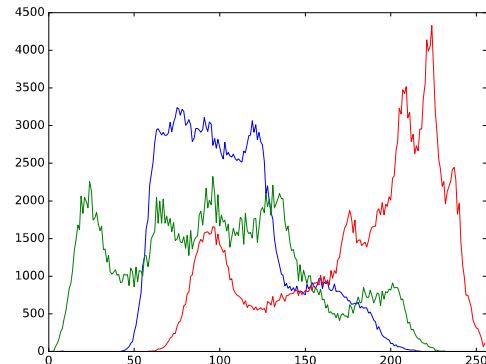
Celem pracy jest częściowa automatyzacja obsługi panelu operatorskiego przez manipulator zamieszczony na mobilnym robocie. Aplikacja tworzona była z myślą o zawodach łazików marsjańskich, gdzie jedną z konkurencji jest obsługa paneli zwierających pokrętła, klawiatury oraz różnego typu przełączniki. W przypadku zdalnego sterowania robotem, przy ograniczonej wizji z kamer wykonywanie precyzyjnych operacji manipulatorem jest procesem trudnym i czasochłonnym. Przedstawione poniżej rozwiązania mogły by znacznie ułatwić pracę operatorowi, lub nawet całkowicie wykonać za niego zadanie.

Aplikacja składa się z szeregu algorytmów wizyjnych umożliwiających zebranie informacji o scenie oraz sterujących manipulatorem. W pierwszej kolejności wykrywany jest obszar zainteresowania, czyli panel operatorski z którym chcemy pracować. Następnie określana jest pozycja i orientacja robota względem panelu, a na obrazie wykonywane są algorytmy korygujące zniekształcenia perspektywiczne i rekonstruujące płaskie elementy sceny. Na tak przygotowanym obrazie podejmowana jest próba detekcji obiektów które manipulator mógłby obsłużyć (np. klawiatury, pokrętła, przełączniki). Po zebraniu informacji o panelu i poleceń od operatora, przeprowadzana jest pełna operacja manipulatorem, bądź ustawienie manipulatora w pozycji i orientacji, z której operator najłatwiej przeprowadzi akcję samodzielnie.



3.1.1 Korekta Bieli

Jednym z problemów napotykanych w algorytmach wizji komputerowej jest oświetlenie sceny. Nierównomierność oświetlenia, oraz barwa światła padającego na obiekty sprawia, że na kolory na fotografii często znaczowo różnią się od rzeczywistych. System wizyjny człowieka bez większego problemu radzi sobie z korektą odchylenia koloru przez nierównomierne oświetlenie, lub barwę padającego światła. Zdolność tę nazywamy stałością kolorów. Dla maszynowych systemów zagadnienie to nie jest jednak trywialne i wymaga zastosowania algorytmów korekcji kolorów. Poniżej przedstawione zostało kilka możliwych algorytmów wzorowanych się na opracowaniu [numer].



Pierwszy z algorytmów znany w literaturze pod nazwą „Gray World Assumption” zakłada, że każdy z kanałów obrazu uśrednia się do szarego koloru po zliczeniu wszystkich pikseli obrazka. Obserwując histogram obrazka, czyli wykres przedstawiający ilość danego koloru zauważamy często, że jeden z kanałów jest przesunięty w stronę wysokich wartości. Jedną z przyczyn może być oświetlenie sceny światłem o odpowiadającej barwie. Przykład takiej sytuacji widoczny jest na zdjęciu [numer], gdzie scena oświetlona jest światłem o cieplej barwie co skutkuje przesunięciem histogramu czerwonego kanału jak na wykresie [numer]. Algorytm wymaga policzenia średniej wartości wszystkich pikseli, jak i średniej wartości pikseli poszczególnych kanałów. Następnie kanały skalowane są zgodnie z podanymi równaniem [numer].

Równania [numer] to wzory na średnie arytmetyczne wartości pikseli całego obrazu oraz jego poszczególnych kanałów.

$$l = \frac{\sum_{i=0}^N \sum_{j=0}^M R_{i,j} + G_{i,j} + B_{i,j}}{3NM},$$

$$l_0 = \frac{\sum_{i=0}^N \sum_{j=0}^M R_{i,j}}{NM}, \quad l_1 = \frac{\sum_{i=0}^N \sum_{j=0}^M G_{i,j}}{NM}, \quad l_2 = \frac{\sum_{i=0}^N \sum_{j=0}^M B_{i,j}}{NM}$$

Gdzie R, G i B to kanały obrazu odpowiadające za odpowiednie kolory. Powyższe parametry dla zdjęcia [numer] wyniosły.

$$l = 128.23, \quad l_0 = 180.22 \quad l_1 = 105.41 \quad l_2 = 99.05$$

Widoczne jest zatem, że dla kanału R średnia wartość jest znacznie wyższa od średniej pozostałych kanałów. Wartości wszystkich pikseli są następnie przemnażane przez iloraz średniej całego obrazka i średniej danego kanału.

$$O_{i,j,k} = \frac{I_{i,j,k} * l}{l_k}, \quad \forall i, j, k$$

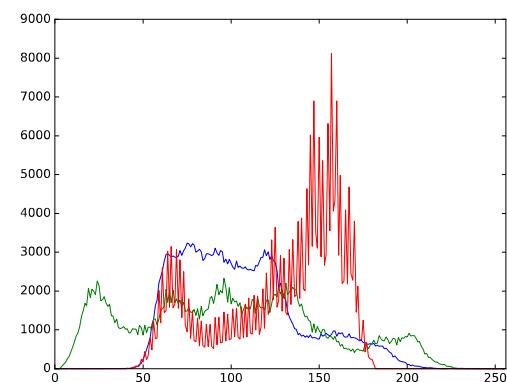
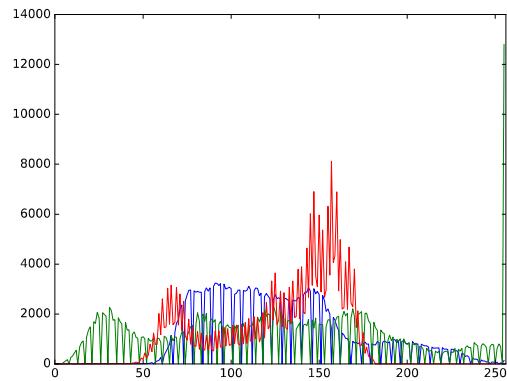
Gdzie O jest obrazem po korekcji, I obrazem modyfikowanym, k numerem kanału, a (i,j) współrzędnymi piksela. Na zdjęciu [numer] możemy zaobserwować jak algorytm wpłynął na obrazek, oraz jego histogram.

Istnieje również wersja algorytmu w której modyfikowany jest jedynie kanał o największej średniej, czyli kanał którego kolorowi odpowiada światło oświetlające scenę.

$$O_{i,j,k} = \frac{I_{i,j,k} * l}{\max(l_0, l_1, l_2)}, \quad \forall i, j, \quad k \rightarrow l_k = \max(l_1, l_2, l_3)$$

$$O_{i,j,p} = I_{i,j,p}, \quad \forall i, j, \quad p = \{0, 1, 2\} / \{k\}$$

Efekt działania tej wersji algorytmu widoczny jest na zdjęciu [numer].



Algorytm „Gray World” jest jednym z najstarszych i najprostszych w implementacji algorytmów korygujących kolor zdjęcia. Mimo, że w części przypadków sprawdza się on wystarczająco dobrze często efekt nie są jednak wystarczająco dobre. Brak spełnienia założenia algorytmu o uśrednianiu się do szarości, wiele różnych źródeł oświetlających scenę, lub prosty obraz o niewielkiej różnorodności są przeszkodami, które uniemożliwiają skuteczną korekcję.

Kolejnym analizowanym algorytmem jest rozciąganie kontrastu obrazu. Polega on na założeniu, że najjaśniejszy element obrazu nie jest biały, a najciemniejszy czarny ze względu na barwę światła oświetlającego scenę. Implementacja polega na przeszukaniu obrazu celem znalezienia najmniejszej i największej wartości piksela każdego kanału i przeskalowania wszystkich wartości do otrzymanego w ten sposób przedziału.

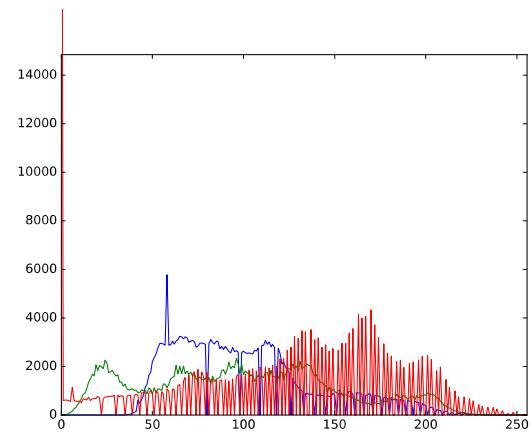
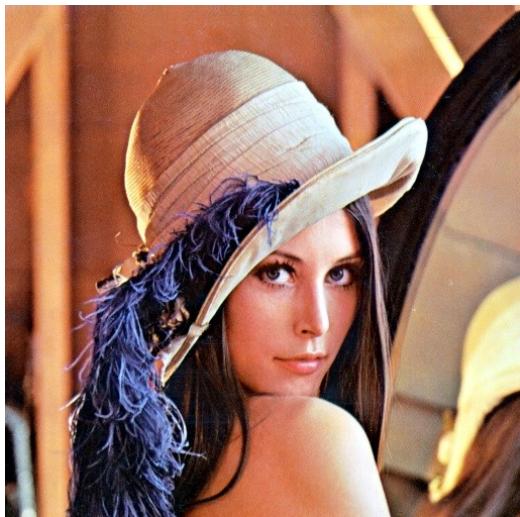
$$u_k = \max(I_{i,j,k}), \quad d_k = \min(I_{i,j,k}), \quad \forall i, j, k$$
$$O_{i,j,k} = \frac{I_{i,j,k} - d_k}{u_k - d_k} * 255, \quad \forall i, j, k$$

Gdzie (u_k, d_k) to maksymalna i minimalna wartość kanału k , I jest obrazem wejściowym, natomiast O obrazem wynikowym algorytmu.

Praktyczne testy wykazały, że znacznie lepsze wyniki można otrzymać poprzez znalezienie średniej z 3% maksymalnych i minimalnych wartości jako parametry skalowania obrazu. Wynika to z faktu, że bardzo często na obrazie pojawiały się pojedyncze piksele o wartościach granicznych (255 lub 0, które są wartościami granicznymi w dla kanałów o ośmiobitowej rozdzielcości), co skutkowało całkowitym brakiem wpływu na obraz wejściowy. Parametry u_k i d_k dla zdjęcia [numer] wyniosły.

$$u_k = [54.71 \quad 12.46 \quad 78.82], \quad d_k = [190.20 \quad 209.09 \quad 245.34]$$

Algorytm rozciągania kontrastu pomimo swojej prostoty zwraca zadowalające wyniki dla większości testowych obrazów. Pozytywną cechą tego rozwiązania jest również fakt, że nawet dla wyjątkowo wymagających przypadków, dla których bilans kolorów nie został wykonany wystarczająco dobrze nie zauważono negatywnego wpływu na jakość obrazu wejściowego jak miało to miejsce w przypadku algorytmu „Gray World”. Zdjęcie [numer] pokazuje obraz po korekcji barw wraz z jego histogramem.



Ostatnim omawianym algorytmem jest Retinex, metoda opracowana w 1986 roku przez Edwina Landa [numer]. Inspiracją dla algorytmu było zaobserwowanie faktu, iż postrzeganie kolorów przez człowieka nie zależy jedynie od mocy i długości fali, lecz również od światła emitowanego przez sąsiedztwo obserwowanego obiektu. Założeniem algorytmu jest więc estymacja jasności obrazu, a następnie jego przeskalowanie. Istnieje wiele implementacji algorytmu, poniżej przedstawiam jej przykład. Implementacja zakłada użycie kanały zielonego jako referencyjnego dla czerwonego i niebieskiego. Wynika to z faktu, że kolor zielony jest zazwyczaj najmniej zaburzany przez wpływ oświetlania.

Pierwszym krokiem algorytmu jest przemnożenie każdego piksela kanału czerwonego i niebieskiego przez iloraz największej wartości kanału zielonego i maksymalnej wartości aktualnego kanału.

$$O_{i,j,k} = I_{i,j,k} * \frac{\max(I_{i,j,g})}{\max(I_{i,j,k})} \quad \forall i, j, \quad x=\{r, b\}$$

Gdzie O to obraz wynikowy, I obraz wejściowy, (i, j) to współrzędne piksela, natomiast k to kanał obrazu.

Następnie obliczane jest macierzowe równanie dla kanału R oraz B

$$Ax=b$$

$$A = \begin{bmatrix} \sum_{i=0}^N \sum_{j=0}^M I_{i,j}^2 & \sum_{i=0}^N \sum_{j=0}^M I_{i,j} \\ \max(I_{i,j}^2) & \max(I_{i,j}) \end{bmatrix}, \quad b = \begin{bmatrix} \sum_{i=0}^N \sum_{j=0}^M G_{i,j} & \max(G_{i,j}) \end{bmatrix}$$

Którego rozwiązaniem są współczynniki

$$x = [x_1 \quad x_2]$$

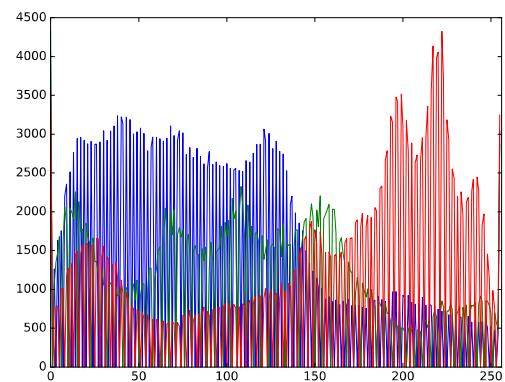
Z wykorzystaniem współczynników x wyliczana jest finalna wartość każdego piksela

$$O_{i,j} = x_1 I_{i,j}^2 + x_2 I_{i,j}$$

Gdzie O jest obrazem wyjściowym algorytmu. Należy również upewnić się, że żaden z pikseli na wskutek skalowania nie przekroczył dozwolonego zakresu wartości.

$$\begin{aligned} O_{i,j} < 0 &\rightarrow O_{i,j} = 0 \\ O_{i,j} > 255 &\rightarrow O_{i,j} = 255 \end{aligned}$$

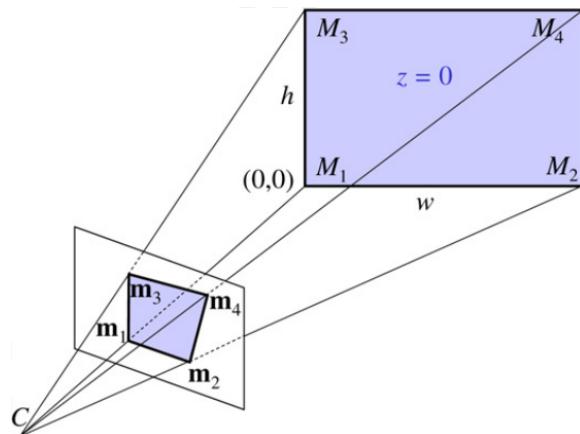
Obraz na którym przeprowadzono algorytm widać na zdjęciu [numer].



Podczas testów zaobserwowano poprawę kolorów, które wydawały się najbardziej naturalne spośród przedstawionych algorytmów. Szczególnie dobre efekty można było zauważać na ciemnych, niedoświetlonych zdjęciach, na których po obróbce można było dostrzec niewidoczne wcześniej szczegóły i detale.

3.1.2 Korekta Perspektywy

Korekta perspektywy będąca algorytmem transformacji 2d wykonywanym na obrazie ułatwia analizę sceny, której płaszczyzna nie jest prostopadła do płaszczyzny urządzenia dokonującego akwizycji. Umożliwia odtworzenie płaskiej sceny znacząco ułatwiając analizę i detekcję kształtów (np. widoczny na zdjęciu nieregularny czworokąt po korekcji zniekształceń perspektywicznych będzie tak jak w rzeczywistości prostokątem). Wizualizację tego problemu możemy zaobserwować na pochodzącej z książki Muralidhara Rangaswamy [numer] ilustracji [numer].



Schemat [numer]. Wizualizacja zaburzenia perspektywicznego [numer]

Transformacja perspektywiczna ma postać:

$$h(x) = Hx$$

Gdzie H to macierz homografii odpowiedzialna za przekształcenie współrzędnych piksela x w nowy układ współrzędnych. Transformacja perspektywiczna składa się z transformacji podobieństwa (H_s), transformacji afanicznej (H_a). Są to operacje hierarchiczne, gdzie każda kolejna operacja posiada większą liczbę stopni swobody.

$$H_p \ni H_a \ni H_s$$

Transformacja podobieństwa składa się z macierzy rotacji R , wektora przesunięcia t , oraz parametru skalowania równokierunkowego s . Operacja ta ma cztery stopnie swobody i umożliwia obrócenie rozpatrywanego obiektu o kąt θ , przesunięcie go o w w obu osiach, oraz proporcjonalną zmianę jego wymiarów.

$$H_s = \begin{bmatrix} s \cos(\theta) & -s \sin(\theta) & t_x \\ s \sin(\theta) & s \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} sR & t \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Transformacja afinczna (transformacja podobieństwa) ma sześć stopnie swobody i odpowiada za zmianę kątów pomiędzy sąsiadującymi bokami o kąt Φ skutkując deformacją obiektu. Transformacja podobieństw jest operacją liniową, niezmienne pozostają zatem własności takie jak równoległość prostych, lub zachowanie stosunku pól figur. Macierz tej operacji dana jest wzorem:

$$H_a = \begin{bmatrix} A & t \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad A = R(\theta)R(-\phi)DR(\phi), \quad D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

Gdzie macierze $R(\pm\phi)$ odpowiadają za kierunek skalowania, a parametry λ_1, λ_2 są współczynnikami skalowania.

Transformacja perspektywiczna wprowadza dwa dodatkowe stopnie swobody poprzez zastosowanie nieliniowego obrotu oraz przesunięcia za które odpowiedzialny jest wektor v^T . Finalna macierz dana jest wzorem:

$$H = \begin{bmatrix} A & t \\ v^T & v \end{bmatrix}$$

Warto zauważyć, że w macierzy H istotny jest jedynie stosunek między elementami, a nie ich wartość. W miejsce parametru v nie została wpisana stała wartość jedynie ze względu na osobliwości które mogłyby się wtedy pojawić.

Niezbędne do obliczenia macierzy projekcji są przynajmniej 4 punkty znajdujące się na płaszczyźnie, na której chcemy wykonać transformację oraz 4 punkty w docelowym układzie współrzędnych które będą im odpowiadać. Żadne 3 punkty z wybranych czwórek nie mogą być współliniowe. Na zdjęciu [numer] punktami źródłowymi są rogi szachownicy, natomiast punktami docelowymi kolejne wierzchołki kwadratu o wybranej arbitralnie długości boku. Macierz homografii dla tego przypadku wyniosła:

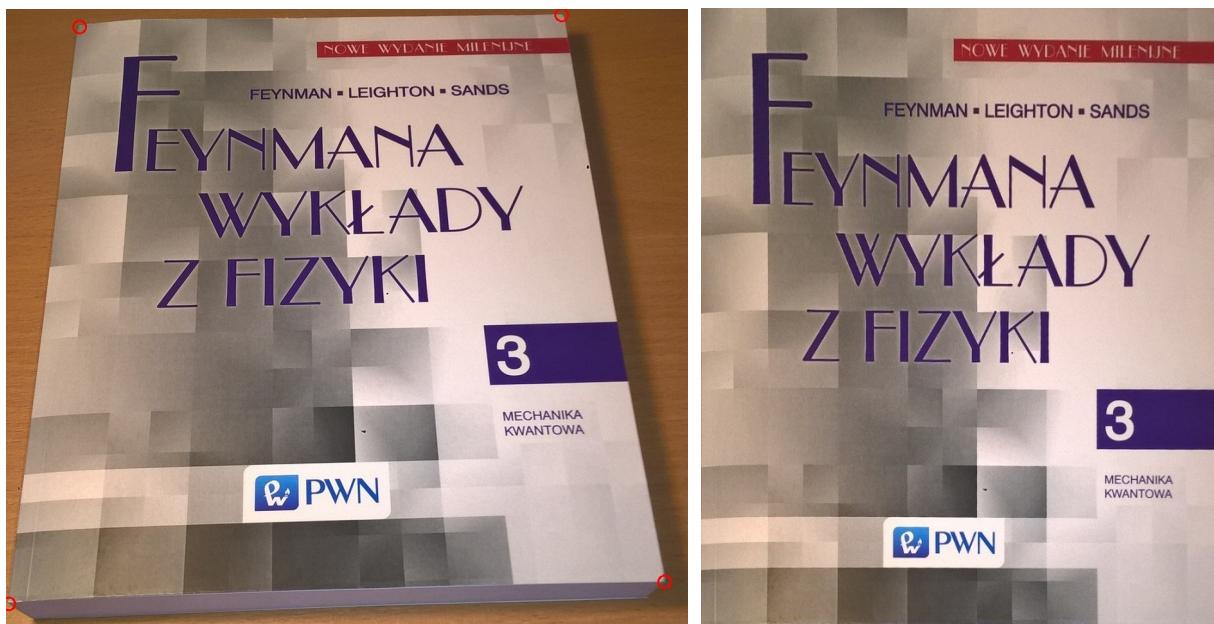
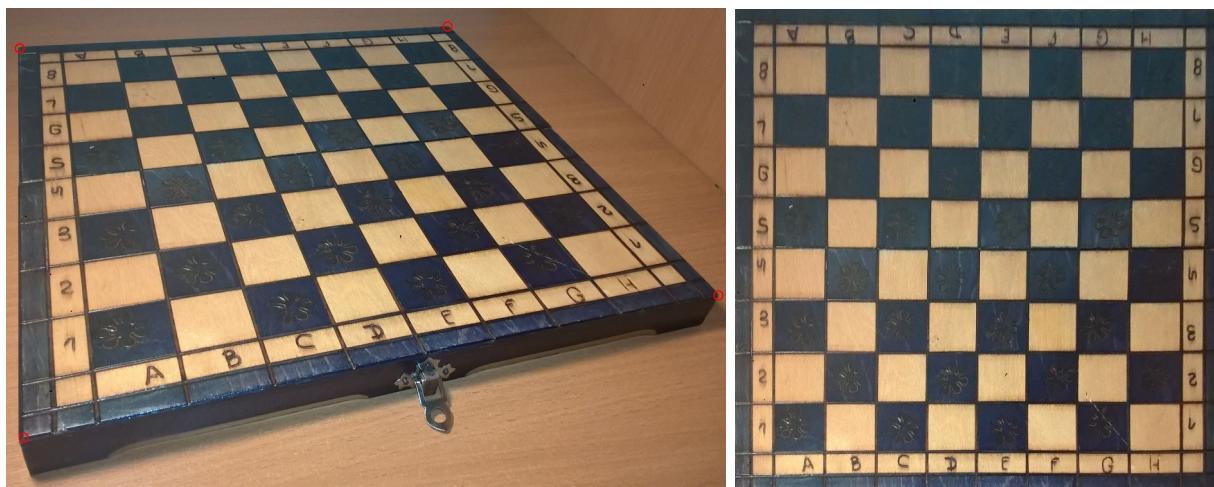
$$H = \begin{bmatrix} 0.554 & -0.007 & -0.011 \\ 0.066 & 1.372 & -117.527 \\ 0 & 0.001 & 1 \end{bmatrix}$$

W przypadku szachownicy z poprzedniego przykładu sytuacja była dość prosta, ponieważ stosunek jej boków jest jednostkowy. W przypadku prostokątnego obiektu jak na zdjęciu [numer] potrzebne są jednak dodatkowe obliczenia, szacujące proporcje figury. Umożliwi to odtworzenie oryginalnego prostokąta o arbitralnie przyjętej wielkości.

Poniższe równanie przedstawia zależność współrzędnych rzeczywistych M z współrzędnymi obrazka m oraz parametrami urządzenia dokonującego akwizycji (f,s) [numer].

$$\lambda \cdot m = A \cdot [R \ t] \cdot M, A = \begin{bmatrix} f & 0 & u_0 \\ 0 & sf & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

gdzie f to ogniskowa kamery, a współczynnik s to stosunek długości boków piksela na obrazie. Przyjmujemy, że piksele są kwadratowe, zatem s = 1. R oraz t opisuje transformację między układami współrzędnych, $[u_0, v_0]$ jest natomiast punktem opisującym środek obrazu. Przekształcając powyższe równanie analogicznie jak w [] otrzymujemy wzór, który z wystarczająco dobrą dokładnością szacuje proporcje prostokąta. Wynik przeprowadzonej w ten sposób transformacji widoczny jest na zdjęciach [numer].



2.2 Detekcja i śledzenie kropki wskaźnika laserowego

Znajdująca się na obrazie kropka lasera może dostarczyć cennych informacji na temat położenia i orientacji obiektu do którego przymocowany jest wskaźnik. Przedstawiony w tym rozdziale algorytm umożliwia detekcję kropki lasera, oraz śledzenie jej położenia w czasie.

Celem uzyskania lepszych wyników na wejściowym obrazie z kamery przeprowadzana jest filtracja obrazu za pomocą filtra obustronnego (bilateral filter). Filtracja ta pozwala pozbyć się z obrazu części szumów, przy jednoczesnym braku wygładzania krawędzi. Wyjściowa wartość każdego piksela jest uzależniona od geometrycznego oraz fotometrycznego położenia jego sąsiadów (opracowanie na podstawie [numer]).

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi$$
$$k = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi$$

Pierwsza składowa filtracji $c(\xi, x)$ mierzy geometryczną bliskość, natomiast składowa $s(\xi, x)$ fotometryczną bliskość piksela centralnego x oraz piksela sąsiadującego ξ . Funkcja k normalizuje wartość piksela. W najczęściej stosowanym przypadku funkcje c oraz s są identycznymi funkcjami gaussowskimi, zapisywanymi jak we wzorze [numer].

$$g(\xi, x) = e^{-1/2 \left(\frac{|\xi - x|}{\sigma} \right)^2}$$

Gdzie σ jest parametrem odpowiadającym za intensywność filtracji. Przykładowe wyniki filtracji widoczne są na obrazie [numer]



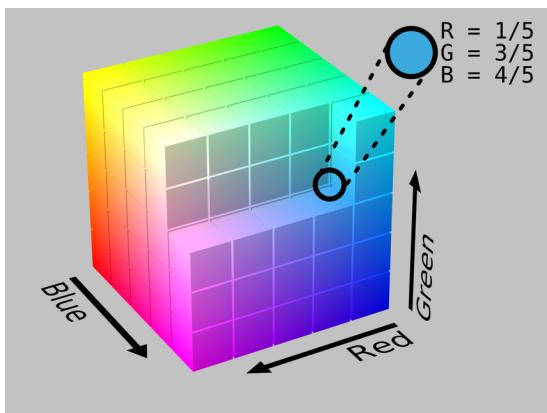
Następnie obraz po filtracji poddawany jest binaryzacji z granicami, aby na finalnym obrazku pozostały jedynie czerwone piksele. Domyślna przestrzeń barw RGB nie jest w tym przypadku wygodna w użyciu, przeprowadzona więc została konwersja na przestrzeń HSV (odcień, nasycenie, wartość). Przestrzeń ta w odróżnieniu od sześcienniej przestrzeni RGB ma kształt stożka, gdzie odcień jest kątem na kole w przedziale $0\text{--}360^\circ$, nasycenie jest promieniem podstawy, natomiast wartość jego wysokością. Zaletą przestrzeni HSV jest fakt, iż informacja o kolorze przechowywana jest na jednym kanale, co ułatwia dobranie odpowiednich progów binaryzacji. Transformację przestrzeni RGB w HSV przedstawiają wzory [numer - numer].

$$M = \max(R, G, B), \quad m = \min(R, G, B), \quad C = M - m$$

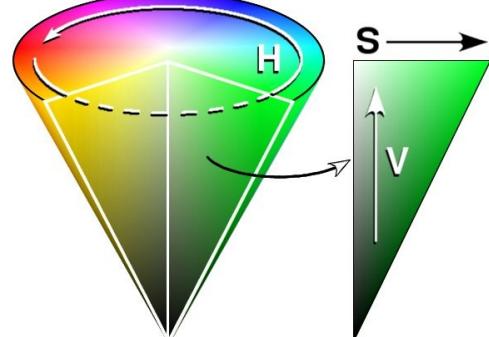
$$H = 60 \begin{cases} 0, & \text{jeżeli } M = 0 \\ \text{modulo}_6 \left(\frac{G - B}{C} \right), & \text{jeżeli } M = R \\ \frac{B - R}{C} + 2, & \text{jeżeli } M = G \\ \frac{R - G}{C} + 4, & \text{jeżeli } M = B \end{cases}$$

$$V = M$$

$$S = \begin{cases} 0, & \text{jeżeli } V = 0 \\ \frac{C}{V}, & \text{w przeciwnym przypadku} \end{cases}$$



Rysunek [numer]. Wizualizacja przestrzeni RGB.
By SharkD - Own work. Download source code., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=9803283>



Rysunek [numer]. Wizualizacja przestrzeni HSV.
By Wapcaplet - From en wiki, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=308191>

Kolor czerwony w przestrzeni HSV reprezentowany jest jako 0° , dozwolony zakres na kanale H został zatem wybrany jako przedział $340-20^\circ$. Granica dla kanałów S i V wybrana tak, aby zaakceptowany został przedział 0.6-1. Schemat [numer] przedstawia operację binaryzacji w granicach.

$$I_{i,j} = \begin{cases} 1, & \text{jeżeli } H_{i,j} \in \langle 320, 20 \rangle \wedge S_{i,j} \in \langle 0.6, 1 \rangle \wedge V_{i,j} \in \langle 0.6, 1 \rangle, \\ 0, & \text{w przeciwnym wypadku} \end{cases}, \quad \forall i, j$$

Obraz binarny poddawany jest następnie morfologicznej operacji zamknięcia, czyli postępującej po sobie operacji erozji i dylatacji. Pozwala to na pozbycie się drobnych szumów z obrazu, jak i wypełnienie dziur i wygładzenie krawędzi właściwego obszaru. Operacje erozji i dylatacji pokazane są na wzorach [numer].

$$\begin{aligned} E_{i,j} &= \min(I_{i\pm k, j\pm k}) \quad \forall i, j \\ D_{i,j} &= \max(I_{i\pm k, j\pm k}) \quad \forall i, j \end{aligned}$$

Gdzie k jest parametrem odpowiadającym za rozmiar nakładanej maski. Na tym etapie na obrazie powinny pozostać już tylko obiekty w kolorze czerwonym. Aby odróżnić kropkę lasera od innych plam na obrazie dla każdego białego obszaru znajdowany jest najmniejszy okrąg zamykający wszystkie piksele plamy. Na tej podstawie obliczane są dwie cechy umożliwiające klasyfikację obszaru.

- promień znalezionej kropki dający informację o wielkości białej plamy.
- stosunek pola okręgu do ilości białych pikseli w obszarze informujący o podobieństwie białego pola do koła.

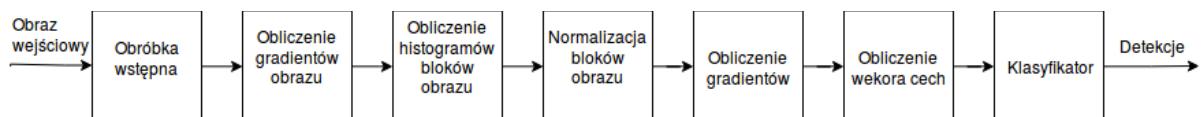
Jeśli cechy mieścią się w dozwolonych granicach obszar traktowany jest jako kropka wskaźnika laserowego, a współrzędne jego środka zapamiętywane. Jeśli kolejna iteracja algorytmu zwróci współrzędne znajdujące się dostatecznie blisko poprzedniej detekcji punkt dodawany jest do listy będącej historią położen kropki lasera. Przykładowa historia znajduje się na zdjęciu [numer]. Aktualna detekcja jest tam oznaczona zielonym okręgiem, historia położeń jest natomiast jako linie łączące ostatnie n punktów o coraz jaśniejszym odcieniu koloru czerwonego.



Zdjęcie [numer]. Detekcja kropki lasera wraz z historią jego położień.

3.3.1 Histogram zorientowanych gradientów

Histogram zorientowanych gradientów, dalej HOG jest deskryptorem cech obrazu, czyli jego uproszczoną reprezentacją. Deskryptor cech umożliwia zachowanie istotnych dla detekcji cech obrazu przechowywanych w wygodnej, zazwyczaj wektorowej formie, oraz pozbycie się danych nie mających na detekcję większego wpływu. W przypadku HOGa zachowywana jest informacja o gradiencie obrazu, czyli jego krawędziach, odrzucane są natomiast dane dotyczące koloru. Finalnie deskryptor ma postać jednowymiarowego wektora, który może zostać wykorzystany zarówno do nauki klasyfikatora jak i detekcji obiektu na obrazie.



Schemat [numer]. Operacje składające się na stworzenie klasyfikatora z użyciem deskryptora cech HOG.

Aby poprawnie zastosować HOGa wymagane jest, aby obrazy z których tworzone są wektory cech miały stały współczynnik proporcji. Pierwszym krokiem obróbki wstępnej jest zatem zmiana niezgodnej z założeniami wielkości obrazka. Następnie korygowane są kolory poprzez zastosowanie algorytmu

Obróbka wstępna polega na przeprowadzeniu korekcji koloru oraz zmiany rozdzielczości obrazu wejściowego tak aby jego współczynnik proporcji pokrywał się z arbitralnie wybranym współczynnikiem. Jako algorytm korekcji koloru przyjęty został algorytm rozciągania kontrastu zaprezentowany w rozdziale 3.1.1.

Następnie wyliczane są gradienty kierunkowe obrazu poprzez nałożenie następujących masek, znanych jako operatory Sobela o rozmiarze jeden. Maska k_x pozwala wykryć linie poziome, natomiast k_y linie pionowe.

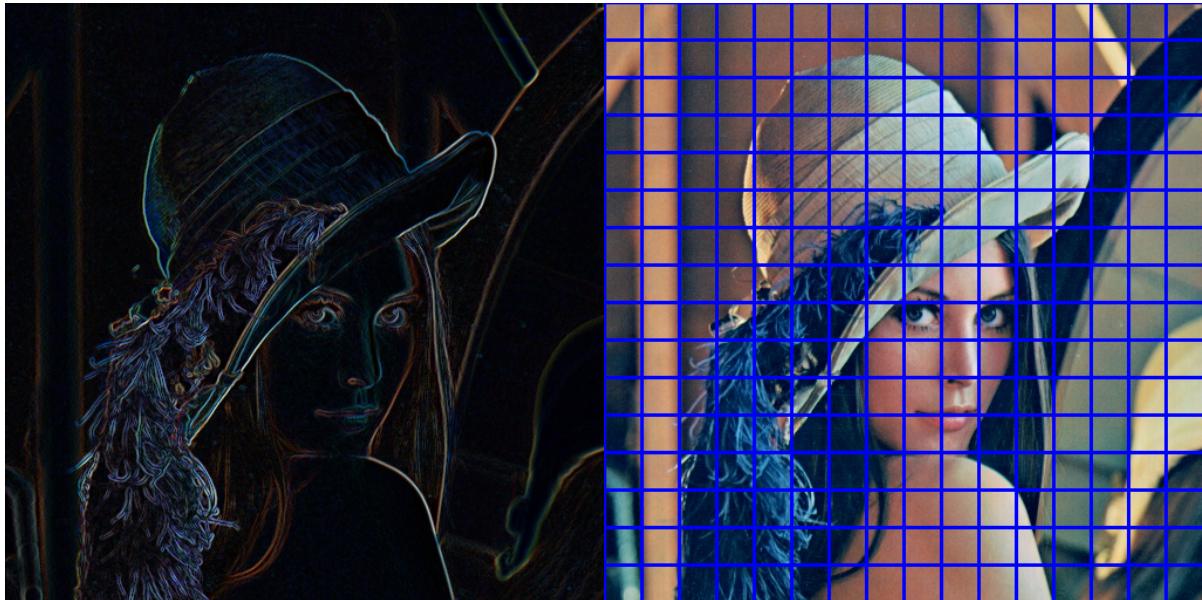
$$k_x = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}, \quad k_y = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$g_{x,i,j} = k_x^T [I_{i-1,j} \quad I_{i,j} \quad I_{i+1,j}], \quad g_{y,i,j} = k_y^T \begin{bmatrix} I_{i,j-1} \\ I_{i,j} \\ I_{i,j+1} \end{bmatrix}, \quad \forall i, j$$

Wykorzystując gradienty kierunkowe obliczana jest wartość, oraz kierunek gradientu w punkcie.

$$g = \sqrt{g_x^2 + g_y^2}, \quad \theta = \arctan\left(\frac{g_y}{g_x}\right)$$

Wizualizacja gradientu obrazu widoczna jest na zdjęciu [numer]. Porównując ten obraz z przedstawionym wcześniej oryginałem [numer] można zauważyć, iż kształt pozostał zachowany i jest łatwo rozpoznawalny. Odrzucone została natomiast większa część tła jak i informacja o kolorze obrazu.



Trzecim etap algorytmu rozpoczyna się od podzielenie obrazu na pomniejsze bloki o jednakowej wielkości. Wymiary bloku wybierane są arbitralnie, tak aby uśredniony dla wycinka gradient możliwie najlepiej opisywał cechy interesującego nas obiektu. Dla zdjęcia testowego wybrany został blok o wielkości 32x32 pikseli tak jak zostało to przedstawione na zdjęciu [numer]. Następnie dla każdego bloku znajdowany jest histogram, gdzie oś pozioma reprezentuje kierunek gradientu, natomiast pionowa sumę wartości gradientów o tym kierunku. Dla uproszczenia późniejszych obliczeń pozioma oś histogramu sprowadzana jest do 9 przedziałów, po 20° każdy. Wartości znajdujące się poza środkiem przedziału są proporcjonalnie rozdzielane pomiędzy przedziały sąsiadujące. Uproszczenie to pozwala na przedstawienie histogramu w postaci wektora o 9 elementach.

Tak przygotowane histogramy poddawane są normalizacji, aby jeszcze bardziej niż zostało to przeprowadzone w obróbce wstępnej, uniezależnić deskryptor od oświetlenia sceny. W tym celu wyliczana jest wektorową norma L_2 dla histogramu normalizowanego, oraz 4 jego sąsiadów. Każdy element normalizowanego wektora jest następnie dzielony przez obliczoną normę.

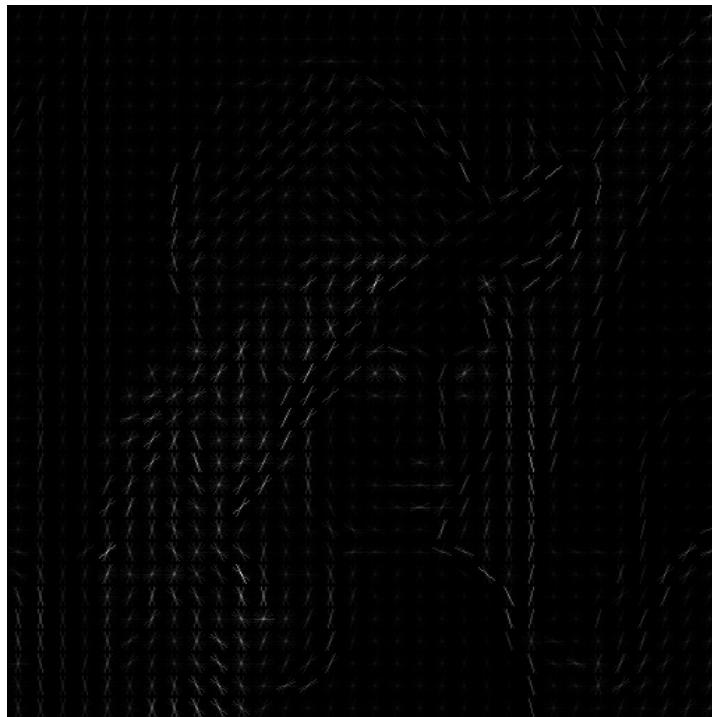
$$L_2 = \sqrt{\sum_{i=0}^N \sum_{j=0}^M \sum_{k=0}^4 H_{i,j,k}^2}, \quad \forall i, j, k$$

$$Hn_{i,j,0} = \frac{H_{i,j,0}}{L_2}, \quad \forall i, j$$

Gdzie Hn jest znormalizowanym histogramem, (i,j) to indeksy bloku obrazu, natomiast k jest indeksem normalizowanego bloku, oraz 4 jego sąsiadów. Wizualizacja znormalizowanego histogramu została przedstawiona na obrazie [numer].

Przygotowane w poprzednich krokach informacje o gradiencie łączone są finalnie w jednowymiarowy wektor będący Histogramem Zorientowanych Gradientów. Jego długość jest iloczynem ilości bloków na które podzielony został obraz i długości wektora reprezentującego histogram pojedynczego bloku. W przypadku przedstawianego przypadku HOG będzie składał się 9216 elementów. Jest to bardzo duża wartość, wynikająca przed wszystkim z rozmiaru obrazka wejściowego. Duży obrazek wybrany został, dla celów pokazowych, w praktycznym zastosowaniu jego rozmiary będą znacznie mniejsze.

Zastosowanie HOGa pozwoliło skompresować obraz o 3 kanałach i 512*512 pikselach do jednowymiarowego wektora, przy jednoczesnym zachowaniu informacji o kształcie obiektu który chcemy sklasyfikować. Korzystając ze spreparowanych w ten sposób danych możemy użyć względnie prostego w działaniu klasyfikatora SVM (wektory podpierające) celem nauczenia modelu na próbkach uczących i właściwej klasyfikacji w scenie rzeczywistej.



3.3.2 Analiza kształtu konturów

8. Bibliografia

- [1] Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection, 2005.
- [2] Edwin H. Land, John J McCann. Lightness and Retinex Theory, 1971.
- [3] Vivek Agarwal, Besma R. Abidi, Andreas Koschan, Mongi A. Abidi. An Overview of Color Constancy Algorithms, 2006.
- [4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, André van Schaik. EMNIST: an extension of MNIST to handwritten letters, 2017.
- [5] Muralidhar Rangaswamy, Fred Harris. Digital Signal Processing A Review Jurnal, 2007.
- [6] Richard Hartley. Multiple View Geometry in Computer Vision, 2003.
- [7] C. Tomasi R. Manduchi. Bilateral Filtering for Gray and Color Images, 1998.