

Sterowanie pracą maszyn z wykorzystaniem analizy obrazów.

Krzysztof Ślot

7 listopada 2019

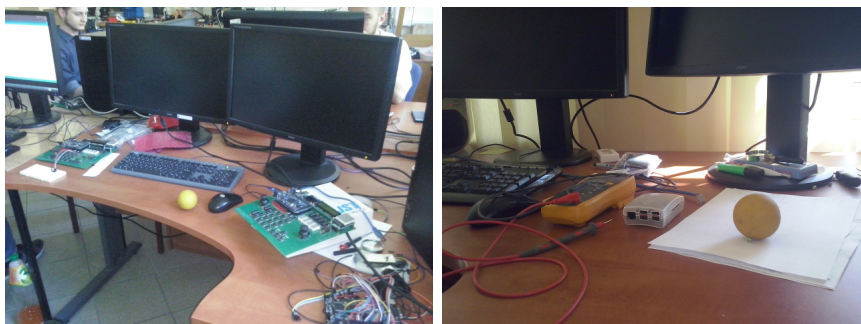
1 Wprowadzenie

Podstawową cechą maszyn autonomicznych jest zdolność pozyskiwania i samodzielnej analizy informacji o otoczeniu, w zakresie niezbędnym dla wykonywania zadanej misji. Ponieważ najbogatszym źródłem takiej informacji są obrazy (w warunkach zadowalającej widzialności), umiejętność ich analizy jest koniecznym elementem oprogramowania komputera sterującego pracą maszyny. Celem kursu jest przedstawienie wybranych obszarów dziedziny automatycznej analizy obrazów, pozwalających na realizację kilku prostych scenariuszy działania systemów autonomicznych. Rozważanymi zadaniami, które mają być wykonywane przez maszynę są:

1. Poszukiwanie w obrazie prostego obiektu (kuli o jednorodnym kolorze (rys. 1) przez pojazd, zwieńczone dojechaniem do tego obiektu na określoną odległość.
2. Jazda po linii (2), wyznaczającej ścieżkę ruchu (również w tym przypadku, zakłada się, że linie będą miały jednorodny kolor)
3. Funkcja analogiczna jak w p.1, ale tym razem dotycząca obiektów o wizualnie 'bogatej' treści - zróżnicowanej strukturze i kolorystyce (3).

Wspólnym mianownikiem wszystkich przedstawionych w dalszej części strategii analizy obrazów jest założenie rzeczywistego charakteru analizowanych obrazów (pozbawionych sztucznego tła czy wyidealizowanych warunków oświetlenia).

Przetwarzanie obrazów jest niezmiernie szeroką dziedziną, zawierającą szereg rozmaitych wątków, z których w ramach niniejszego kursu omówione zostaną jedynie nieliczne. Mimo to, przedstawione metody będą stanowiły kompletne i praktycznie użyteczne rozwiązania, a przedstawiony materiał będzie dawał podstawy wystarczające dla realizacji szeregu innych zadań analiz, przydatnych dla autonomicznych maszyn, realizujących swoje zadania z wykorzystaniem obrazów jako informacji o otoczeniu.



Rysunek 1: Przykładowe obrazy, zawierające kule stanowiące przedmiot detekcji.



Rysunek 2: Przykładowe ścieżki, których wykrycie ma być efektem analizy.

Rysunek 3: Przykładowe rzeczywiste obiekty, będące przedmiotem detekcji.

1.1 Metodyka analizy

Źródłem informacji dla maszyny będzie sekwencja obrazów (wideo), rejestrowana przez jej kamerę. U podstaw algorytmów opracowywanych dla umożliwienia realizacji podjętych zadań leżą proste reguły, których spełnienie jest często czynnikiem decydującym o powodzeniu analizy. Pierwszą z nich jest dążenie do upraszczania podejmowanego zadania, wykorzystujące całą wiedzę posiadaną o przedmiocie analizy. W odniesieniu do rozważanych problemów detekcji obiektów o jednorodnym kolorze oznacza to na przykład ograniczenie ilości rozważanej w analizie informacji wyłącznie do selektywnych zakresów barw. Drugą zasadą jest zróżnicowanie stopnia rozległości analizy przeprowadzanej w odniesieniu do kadru sekwencji, w której jeszcze nie znaleziono poszukiwanego obiektu (maszyna musi badać cały obraz, bo obiekt, jeżeli jest, może być w jego dowolnym regionie) i w odniesieniu do analizy kadrów sekwencji następujących po wcześniejszej detekcji obiektu (analizie będzie poddawany tylko fragment obrazu wokół znalezionego wcześniej położenia, w promieniu określonym przez fizycznie możliwe przemieszczenie obiektu w odstępie czasu między pobraniem kolejnych obrazów). Uzyskane w ten sposób znaczące zwiększenie szybkości przetwarzania pozwoli na płynną realizację analizy nawet przy użyciu ograniczonych mocy obliczeniowych, którymi może dysponować komputer maszyny.

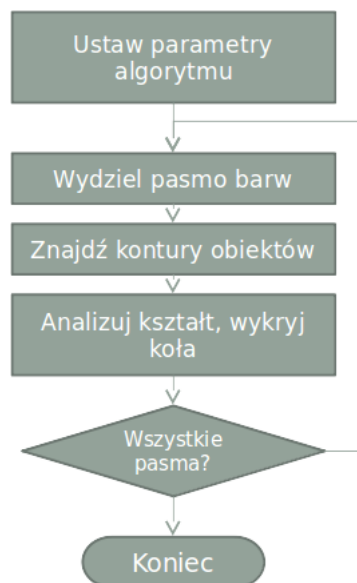
Podstawowym narzędziem do realizacji analiz będzie biblioteka OpenCV [], ale wszystkie etapy przetwarzania są na tyle podstawowe, że można je znaleźć praktycznie w każdym pakiecie oprogramowania dedykowanego do analizy obrazów.

2 Poszukiwanie kuli o jednorodnym kolorze

Narzucającym się sposobem analizy obrazu dla detekcji kuli o jednorodnej barwie jest algorytm, składający się z następujących, iteracyjnie powtarzanych kroków (rys. 4):

1. Wydzielenie z obrazu wejściowego treści odpowiadającej wąskiemu zakresowi barw. Efektem kroku jest obraz binarny, zawierający obiekty (zbiory punktów o barwie zgodnej z aktualnie wydzielanym zakresem chromatycznym) i tło (pozostała część obrazu).
2. Ekstrakcja konturów obiektów, wydzielonych w poprzednim kroku
3. Weryfikacja, czy któryś z wydzielonych konturów jest okręgiem.

Wymienione operacje będą powtarzane dla kolejnych pasm, definiujących rozważane w danej analizie zakresy barw. Zakresy pasm i ich liczba stanowią parametry algorytmu, które należy ustalić w sposób zapewniający poprawną detekcję.



Rysunek 4: Algorytm detekcji kuli o jednorodnej barwie.

Poszczególne zadania niezbędne do realizacji tego algorytmu wymagają wykonania trzech różnych typów analiz, opisanych w kolejnych podpunktach.

2.1 Segmentacja obrazu w funkcji barwy

Jeżeli przedmiotem zainteresowania analizy są obiekty o określonej (lub jednolitej) barwie, naturalnym sposobem ich poszukiwania jest ograniczenie analizy wyłącznie do obszarów obrazu o rozważanej kolorystyce. Podstawowym sposobem reprezentacji obrazów kolorowych jest, wzorowana na sposobie percepcji kolorów przez człowieka, przestrzeń RGB (lub prezentująca te same komponenty w innej kolejności przestrzeń BGR, będąca domyślnym formatem kolorowym w OpenCV). W przestrzeni tej kolory są mieszaniem komponentów podstawowych (czyli czerwonego, zielonego i niebieskiego), przy czym proporcje mieszania dają w wyniku unikatowe zabarwienie (rys. 5), zaś wartości bezwzględne składowych koloru decydują o jasności i wyrazistości uzyskiwanych barw. Oprócz przestrzeni RGB istnieje wiele innych sposobów wyrażania obrazów kolorowych: z trzech komponentów koloru RGB można za pomocą rozmaitych transformacji uzyskiwać inne trójki deskryptorów, eksponując lub tłumiąc różne cechy reprezentacji.

Z punktu widzenia analiz, których podstawą jest zabarwienie obiektów, najdogodniejszym sposobem reprezentacji obrazów są przestrzenie HSV (Hue - barwa, Saturation - nasycenie i Value - wartość) lub HSL (L - luminancja). Komponenty przestrzeni HSV separują od siebie trzy różne aspekty

percepcji wyglądu obiektów. Wartość L (lub V) odpowiada jasności punktu obrazu (lub jasności dominującego koloru) i jest obliczana na podstawie składowych R, G, B punktu w następujący sposób:

$$L = \frac{R + G + B}{3} \quad (1)$$

Nasycenie (komponent S) ocenia stopień dominacji najintensywniejszej składowej koloru i jest określany liczbowo jako:

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)} * 255 \quad (2)$$

Informacja o kolorze jest zawarta w komponencie H , przy czym definicja sposobu jego wyznaczania jest opisowa i można ją wyrazić w następującym algorytmie:

```
delta = max(R,G,B) - min(R,G,B);
if( R == max ) H = ( G - B ) / delta;
else if( G == max ) H = 2 + ( B - R ) / delta;
else H = 4 + ( R - G ) / delta;
H *= 60;
```

Przestrzeń kolorów HSV jest często przedstawiana jako odwrócony stożek barw (rys. 5), którego wierzchołek odpowiada zerowej jasności i zerowemu nasyceniu, wysokość jest osią zmiany jasności, a promień okręgu przekroju poprzecznego oznacza nasycenie. Kąty obrotu wokół wysokości stożka skojarzone są z kolorami - pełny zakres barw jest reprezentowany liczbami z zakresu od 0 do 360 (kąt pełny). Przykładowo, kąty wokół zera odpowiadają różnym odcieniom czerwieni, w jednej szóstej zakresu - mamy żółć, połowie cyjan, w dwóch trzecich - kolor niebieski itp.

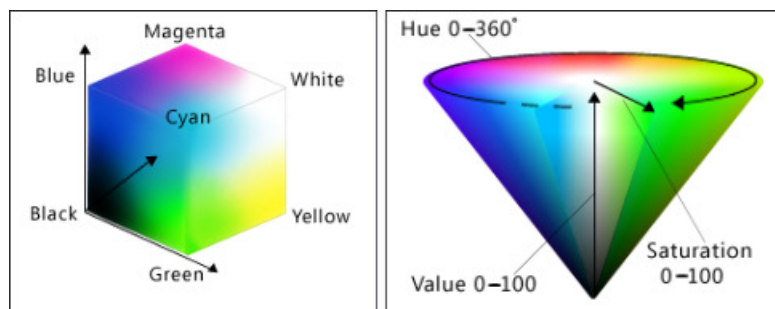
W bibliotece OpenCV do reprezentacji obrazu w przestrzeni HSV stosowane są trzy bajty (po jednym na komponenty H, S i V). Ponieważ zakres bajtu to liczby 0-255, reprezentacja miar kątów odpowiadających kolorom została dwukrotnie zredukowana, więc zakres H jest wyrażany przez wartości od 0 do 179.

2.1.1 Procedura segmentacji obrazu względem koloru

Wydzielenie z obrazu treści zawartej w ustalonym zakresie barw, jest za pomocą biblioteki OpenCV dokonywane poprzez wywołanie funkcji:

```
cv::inRange(in, GD, GG, out);
```

gdzie in oznacza obraz źródłowy, reprezentowany przez obiekt $cv::Mat$ (macierz o trzech kanałach, np. w formacie HSV lub RGB), GD oznacza dolną granicę wydzielanego zakresu wartości, GG - górną granicę, a out jest obrazem wynikowym. Górna i dolna granica zakresu to trzejelementowe wektory,



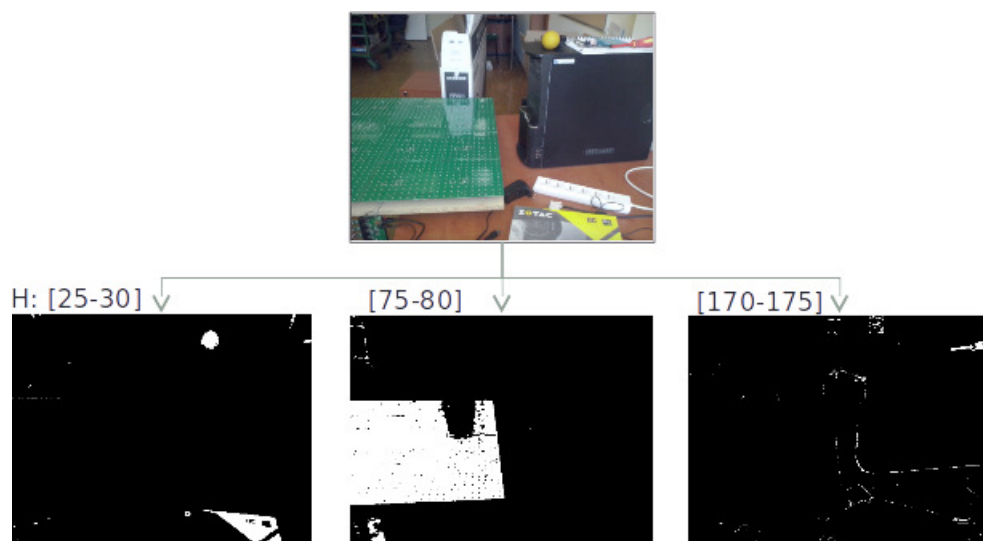
Rysunek 5: Przestrzenie kolorów: podstawowe sposoby reprezentacji obrazów barwnych: RGB (po lewej), HSV (po prawej) [1].

określające odpowiednio minimalne i maksymalne wartości komponentów piksela. Wektory takie można reprezentować w OpenCV za pomocą różnych typów danych, na przykład może to być typ `cv::Scalar` lub typ `cv::Vec3b`. Wszystkie punkty obrazu, dla których wartości wszystkich komponentów leżą wewnątrz zakresu określonego przez granice dolne i górne zostaną w wyniku wykonania operacji `cv::inRange(.)` 'wydzielone' z obrazu, czyli zostanie im przypisana wartość maksymalna (255), podczas gdy wszystkim pozostałym punktom zostanie przypisana wartość '0'. Obraz wynikowy, będzie więc obrazem binarnym, przy czym jego typ i rozmiar zostaną określone automatycznie (nie ma potrzeby jawnego przydzielania pamięci i określania typu).

Fragment przykładowej procedury, wydzielającej z obrazu o nazwie `hsv` (otrzymanego w wyniku konwersji wczytanego z pliku obrazu `img`, dokonanej za pomocą metody `cv::cvtColor(.)`) kolejne pasma koloru, o szerokościach `WinWidth` (ustalonych na 5), z odstępem co '1', ma postać:

```
cv::cvtColor(img,hsv,cv::COLOR_BGR2HSV);
int WinWidth=5;
int minSat=100, minVal=100, maxSat=255, maxVal=255, MinDl=250;
cv::Mat Pasma;
for(int nrPasma=0; nrPasma<180-WinWidth; nrPasma++)
{
    cv::inRange(hsv,cv::Vec3b(nrPasma,minSat,minVal),
               cv::Vec3b(nrPasma+WinWidth,maxSat,maxVal),Pasma);
    // ... detekcja konturu
    // ... weryfikacja konturu
}
```

Przykładowe wyniki ekstrakcji punktów obrazu o wartościach komponentu H należących do trzech różnych zakresów (określonych przez wartości parametrów programu `nrPasma` i `WinWidth` zostały przedstawione na rys. 6.



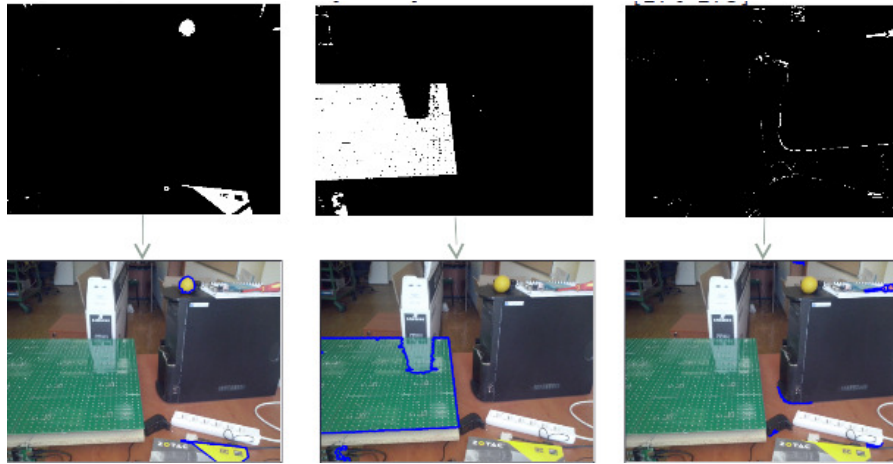
Rysunek 6: Przykładowy obraz (góra) i treść odpowiadająca obszarom o barwach z trzech różnych zakresów komponentu Hue (dół).

2.2 Detekcja konturów obiektów obrazu binarnego

Binarny wynik detekcji obiektów o barwie zawartej w rozważanym aktualnie paśmie stanowi punkt wyjścia drugiej operacji procedury, którą jest ekstrakcja konturów obiektów obrazu. Kontur obiektu obrazu binarnego to zbiór jego punktów brzegowych. Istnieje wiele propozycji realizacji procesu ekstrakcji konturu obiektów obrazów binarnych. Detekcja punktów brzegowych obiektów binarnych jest zadaniem stosunkowo prostym - stanowi ona szczególny przypadek detekcji konturów obiektów w obrazach monochromatycznych, opisaną w części 4. Podstawą detekcji punktów brzegowych jest operator, stanowiący przybliżenie pochodnej za pomocą różnicy wstecznej (patrz rozdział 4), aplikowany kolejno w kierunkach poziomym i pionowym. Dla każdego punktu przefiltrowanego obrazu jest wyznaczany moduł wynikowego gradientu, stanowiący podstawę do podjęcia decyzji, czy punkt jest punktem brzegowym, czy nie.

Operacja ekstrakcji konturu z obrazu binarnego w OpenCV jest wykonywana poprzez wywołanie metody `cv2.findContours()` (patrz fragment kodu). Pierwszym parametrem metody jest obraz binarny (w przedstawionym przykładzie to zmienna `Pasmo`), z którego wydzielane będą kontury, które z kolei będą zapisywane w postaci listy konturów (zawierających listy punktów) o nazwie `Kontury`.

```
Kontury = cv2.findContours(Pasmo,cv2.CV_RETR_EXTERNAL,
                           cv2.CV_CHAIN_APPROX_NONE);
```



Rysunek 7: Wyniki ekstrakcji konturu z trzech różnych obrazów binarnych, nałożone na obraz oryginalny.

Przedstawione w przykładowym kodzie wartości parametrów metody oznaczają żądanie generacji tylko pojedynczego, zewnętrznego konturu dla danego obiektu (`cv::RETR_EXTERNAL`) oraz żądanie, by na liście punktów należących do konturu umieszczone zostały wszystkie jego elementy (`cv::CHAIN_APPROX_NONE`). Efekty wydzielania konturów obiektów z obrazów odpowiadających trzem różnym pasmom barw, zostały przedstawione na rys.7.

2.3 Weryfikacja kształtu konturu

Ostatnia faza algorytmu to weryfikacja, czy kontury wydzielone z obrazu odpowiadają kształtem oczekiwanemu obiektowi, a więc czy są to okręgi. Zadanie to jest szczególnym przypadkiem ogólnego problemu rozpoznawania obiektów na podstawie ich kształtu, dla którego opracowanych zostało wiele różnych metod postępowania, bazujących na schemacie: określ model poszukiwanego obiektu - oceń podobieństwo rozważanego konturu do tego modelu. Stopień złożoności i specyficzności modeli zależy od rodzaju analizowanego kształtu. Okrąg jest najprostszym z nich - posiada prostą definicję matematyczną, którą można wykorzystać jako podstawę do budowy procedury jego detekcji (oczywiście, dla wykrywania okręgów możliwe jest również wykorzystanie procedur ogólnych, pozwalających na rozpoznawanie różnych kategorii kształtów).

Przedstawione w dalszej części procedury detekcji obiektów o konturze zbliżonym do okręgu obejmują zarówno podejścia specyficzne (adekwatne wyłącznie dla tej klasy figur) jak i ogólne. W pierwszym przypadku modelem okręgu będzie jego definicja, a analiza konturu będzie polegała na ocenie podobieństwa rozważanej krzywej (lub jej cech) do okręgu (lub jego cech).

W drugim przypadku, model okręgu (lub koła) będzie zbudowany o wybrane deskryptory statystyczne a rozpoznawanie rozważanego konturu (figury) będzie dokonywane poprzez analizę podobieństwa tych parametrów.

2.3.1 Detekcja kuli na bazie modelu matematycznego

Okrąg to zbiór punktów spełniających równanie:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (3)$$

Za model okręgu można więc uznać zbiór parametrów tego równania, czyli współrzędne środka a, b oraz promień. Podobieństwo nieznanego konturu do okręgu można więc określić badając, czy gdyby środek skupienia punktów nieznannej krzywej (wartość średnia punktów) znalazł się w środku okręgu (punkcie (a, b)), to współrzędne wszystkich elementów tej krzywej spełniałyby relację (3).

Jeżeli środek okręgu (3) leży w środku układu współrzędnych: $(a, b) = (0, 0)$, jego równanie przyjmuje prostą formę:

$$x^2 + y^2 = r^2 \quad (4)$$

Środek nieznanego konturu K to wartość średnia jego punktów, więc jeżeli współrzędne wszystkich jego elementów (x_j, y_j) zostaną skorygowane poprzez odjęcie od nich wartości średniej (środka konturu (m_x, m_y)):

$$X_j = x_j - m_x, \quad Y_j = y_j - m_y, \quad (5)$$

to zadanie sprawdzenia czy kontur jest okręgiem sprowadza się do sprawdzenia dla każdego jego punktu, czy zachodzi:

$$X_j^2 + Y_j^2 = R_j^2 \stackrel{?}{\approx} r^2 \quad (6)$$

Zbadanie czy spełnione są zależności (6) dla wszystkich n punktów konturu K jest równoważne funkcji kryterialnej mierzącej sumaryczny błąd dopasowania, rozumiany jako średni kwadrat (błędy dodatnie i ujemne są traktowane jednakowo) różnicy między wynikiem pożądanym (r) a otrzymanym (R_j), a więc funkcji o postaci:

$$J(r, K) = \frac{1}{n} \sum_{j=0}^{n-1} (R_j^2 - r^2)^2 = \frac{1}{n} \sum_{j=0}^{n-1} (X_j^2 + Y_j^2 - r^2)^2 \quad (7)$$

Wyrażenie (7) nazywane jest błędem średniokwadratowym (ang. *Mean Squared Error* - MSE), oceniającym w rozważanym przypadku jakość dopasowania konturu do okręgu o nieznanym promieniu r . Ponieważ jest ono

funkcją zmiennej r , można określić wartość promienia, zapewniającego jego minimalizację - wystarczy obliczyć pochodną funkcji $J(r)$ względem r i sprawdzić, kiedy jest ona równa zeru:

$$\frac{\partial J}{\partial r} = \frac{-4r}{n} \left(\sum_{j=0}^{n-1} (X_j^2 + Y_j^2 - r^2) \right) = 0 \quad (8)$$

co prowadzi do dwóch warunków:

$$\frac{-4r}{n} = 0 \rightarrow r = 0,$$

który (jak łatwo sprawdzić) odpowiada maksimum funkcji oraz:

$$\sum_{j=0}^{n-1} (X_j^2 + Y_j^2 - r^2) = 0,$$

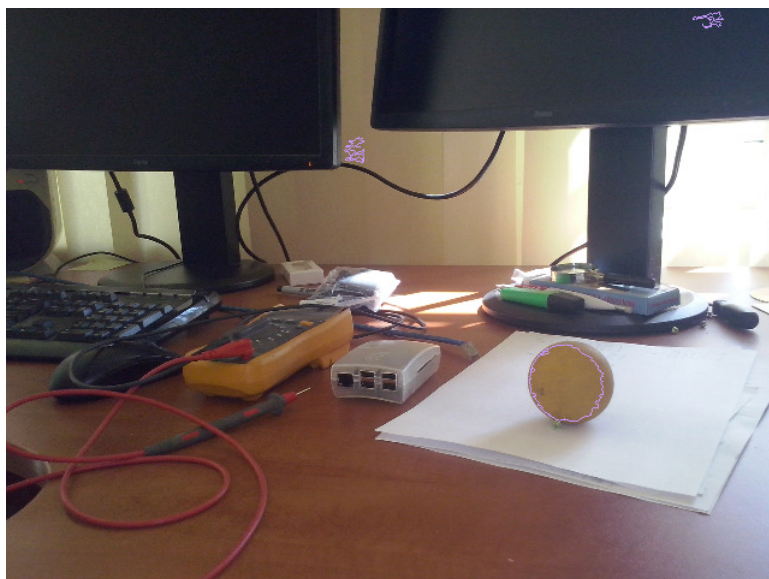
który daje w wyniku wyrażenie pozwalające na obliczenie promienia okręgu najlepiej dopasowanego do zbioru punktów:

$$r = \sqrt{\frac{1}{n} \sum_{j=0}^{n-1} (X_j^2 + Y_j^2)} \quad (9)$$

Decyzja odnośnie podobieństwa analizowanego aktualnie konturu do okręgu jest podejmowana na podstawie wartości średniokwadratowego błędu dopasowania (7). Jeżeli jest on mniejszy od przyjętego progu θ , kontur może być uznany za okrąg, w przeciwnym wypadku - hipoteza może być odrzucona:

$$Y = \begin{cases} 1 & J(r, K) < \theta \\ 0 & J(r, K) \geq \theta \end{cases} \quad (10)$$

Wartość progu powinna być kompromisem między 'specyficznością' a 'czułością' detekcji. Mała wartość progu sprawi, że tylko obiekt bardzo podobny do okręgu będzie zaakceptowany, a więc detekcja będzie bardzo selektywna. Niestety, ekstrakcja konturu obiektów rzeczywistego obrazu jest obciążona błędami. Błędy pojawiają się podczas akwizycji (obraz jest rejestrowany przez dyskretną siatkę, która dodatkowo nie ma idealnej struktury) oraz segmentacji (obiekty nie są jednorodnie oświetlone, a ich części mogą znajdować się w cieniu, co sprawia, że fragmenty obiektu nie spełniają kryterium segmentacji). W efekcie, uzyskiwane kontury są jedynie przybliżeniem rzeczywistego kształtu obiektu, a więc założenie restrykcyjnie niskiego progu decyzyjnego (dużej specyficzności detekcji) będzie skutkowało częstym odrzucaniem właściwych konturów, czyli małą czułością detekcji. Zwiększenie wartości progu, prowadzi do zwiększenia czułości detekcji, ale nieuchronnie

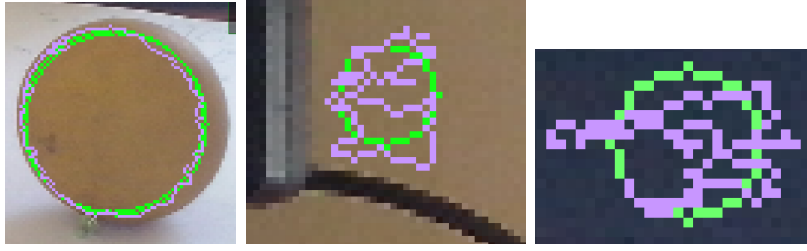


Rysunek 8: Wyniki detekcji 'okręgów' przy użyciu kryterium minimalizacji błędu średniokwadratowego, przy założeniu że minimalna długość konturu wynosi 100. Widoczne dwa błędnie wydzielone kontury.

proceedzi do zmniejszenia jej specyficzności, prowadząc do akceptacji kształtów coraz mniej podobnych do założonego. Wskazówką dla wyboru wartości konkretnego progu jest sens, kryjący się za wyrażeniem (7): wartość wyrażenia to średni kwadrat odstępstw odległości punktów konturu od długości promienia (podniesionych do kwadratu), odniesiony do długości konturu.

Przyjęcie założenia, że wynikiem detekcji konturu, który zapewnia minimalizację błędu średniokwadratowego (7) będzie krzywa o kształcie przypominającym okrąg jest niedocenieniem problemów serwowanych przez obrazy rzeczywiste. Pierwszym z nich, choć stosunkowo prostym do wyeliminowania, jest obfitość krótkich konturów o kształtach przypominających okręgi, wydzielanych z przypadkowo powstających w wyniku segmentacji kształtów podczas segmentacji obrazu dla przedziale kolorów rozważanego w danej fazie procedury. Eliminacja fałszywych detekcji polega na eliminacji z analizy wszystkich konturów krótszych niż zadany próg (lub lepiej, konturów mogących reprezentować okręgi o zbyt małym promieniu), co przekłada się na założenie, że poszukiwany obiekt musi posiadać w obrazie pewien minimalny rozmiar. Niestety, w praktyce takie podejście nie jest wystarczające, a wyniki błędnych detekcji konturów, dla których założono minimalną długość 100 elementów, zostały przedstawione na rys.8.

Błędnie wydzielone kontury w ogóle nie przypominają okręgu, mimo to, przechodzą pomyślnie test podobieństwa z dość restrykcyjnie ustalonym pro-



Rysunek 9: Kontury wydzielone w wyniku detekcji (zilustrowanej na rys.8) z dopasowanymi okręgami.

giem akceptacji. Przyczyna błędów, to nie najgorsze dopasowanie konturów do okręgów, a przede wszystkim, duża liczba elementów konturu zmniejszająca błąd (duża wartość n w mianowniku równania (7)), co zostało zilustrowano na rys.9. Dlatego też, sugerowaną modyfikacją kryterium podobieństwa (7) jest modyfikacja ilorazu, do postaci:

$$J(r, K) = \frac{1}{(2\pi r)^2} \sum_{j=0}^{n-1} (R_j^2 - r^2)^2 = \frac{1}{(2\pi r)^2} \sum_{j=0}^{n-1} (X_j^2 + Y_j^2 - r^2)^2 \quad (11)$$

gdzie kwadrat długości okręgu został użyty w celu uproszczenia doboru progu (kryterium mówi teraz o łącznym kwadracie odstępstw od okręgu, odniesionym do długości tego okręgu).

Dodatkowym zabiegiem, który powinien stanowić element procedury odfiltrowywania błędnych konturów, powinno być sprawdzenie, czy długość konturu odpowiada, z zadaną tolerancją (np 20%) długości okręgu o optymalnej wartości promienia.

Model matematyczny 2

Jeżeli przedstawione w poprzedniej części zabiegi okażą się niewystarczające, możliwe jest sformułowanie alternatywnego kryterium oceny podobieństwa konturu do okręgu. Podstawą tego kryterium jest model, zakładający że okrąg to krzywa parametryczna, której elementy spełniają warunek:

$$x(i) = R \cos\left(\frac{2\pi}{T} i\right), \quad y(i) = R \sin\left(\frac{2\pi}{T} i\right), \quad i = 1 \dots n \quad (12)$$

gdzie założono, że: okrąg ma środek w początku układu współrzędnych, pierwsza próbka sekwencji ($x(i = 1)$) odpowiada kątowi '0' i kąty zmieniają się w kierunku odwrotnym do ruchu wskazówek zegara (patrz rys.??). Ponieważ okrąg ma długość $2\pi R$, równania okręgu (12) przekształcają się do postaci:

$$x(i) = R\cos(i/R), \quad y(i) = R\sin(i/R), i = 1 \dots n \quad (13)$$

Porównanie, czy nieznaną kontur K^a jest podobny do okręgu to sprawdzenie, jakie jest średnie odstępstwo jego elementów od krzywej parametrycznej (13), a więc określenie wartości kryterium:

$$J^p(r, K^a) = \frac{1}{n} \left(\sum_{i=1}^n (x_i^a - r\cos(i/r))^2 + \sum_{i=1}^n (y_i^a - r\sin(i/r))^2 \right) \quad (14)$$

gdzie r jest znalezionym (przy użyciu równania 9) optymalnym promieniem, a x_i^a, y_i^a są kolejnymi elementami konturu K^a .

Przykładowy kod programu, zawierający procedury oceny dopasowania konturów do okręgu znajduje się na stronie kursu.

Algorytm detekcji

Sugerowany algorytm analizy podobieństwa konturu do okręgu składa się z następujących faz:

1. Eliminacja krótkich konturów. Próg dotyczy **liczby** elementów konturu (n) i powinien być szacowany w sposób zapewniający wykrycie kuli o zadanym, minimalnym promieniu r_{min} ($n \approx 2\pi r_{min}$).
2. Oszacowanie promienia okręgu najlepiej dopasowanego do rozważanego konturu (r) i ponowna eliminacja konturów mniejszych od zadanego progu (tym razem próg to promień okręgu, a odrzucane są kontury, dla których $r < r_{min}$).
3. Sprawdzenie kryterium średniokwadratowego podobieństwa punktów konturu do okręgu (kryterium (7)). Kontury, dla których kryterium podobieństwa ma wartości mniejsze od założonego progu θ stanowią wstępny wynik detekcji.
4. Opcjonalna weryfikacja zaakceptowanych konturów. Metodą weryfikacji może być na przykład sprawdzenie drugiego z przedstawionych kryteriów zgodności konturu z modelem okręgu (14).

2.3.2 Detekcja na podstawie statystycznych deskryptorów kształtu

Alternatywą dla modelu okręgu budowanego na podstawie równania, jest zastosowanie ogólnych metodologii parametrycznego opisu kształtów (a więc kojarzącego z danym kształtem wartości ściśle zdefiniowanych parametrów). Najczęściej stosowanym podejściem jest statystyczny opis kształtu, czyli

opis wykorzystujący podstawowe deskryptory statystyczne, takie jak wartości średnie, wariancje, kowariancje, itd., określane ogólnie mianem 'momentów zwykłych' lub 'momentów centralnych'. Momenty zwykłe dla figur lub konturów określonych w dyskretnej, dwuwymiarowej siatce, są zdefiniowane jako:

$$m_{p,q} = \sum_i \sum_j I(x,y) x^p y^q \quad (15)$$

gdzie $I(x,y)$ jest funkcją określającą jasność punktów obrazu binarnego poddawanego analizie (przyjmującą wartości '0' dla punktów tła i '1' dla punktów należących do obiektów), a x,y są współrzędnymi tych punktów (indeksami kolumn i wierszy). p i q oznaczają 'rzędy' momentów. Przykładowo, gdy dany jest binarny obiekt, to obliczony dla niego moment m_{00} określa jego powierzchnię ($x^0 y^0 = 1$, czyli 15 sprowadza się do zsumowania wszystkich punktów należących do obiektu). Podobnie, dla konturu, moment m_{00} oznacza jego długość. Momenty m_{10} i m_{01} podzielone przez m_{00} to wartości średnie punktów obiektu lub konturu, odpowiednio dla kierunku poziomego i pionowego.

Momenty centralne to modyfikacja momentów zwykłych polegająca na przeniesieniu układu współrzędnych do centroidu rozważanego obiektu:

$$\mu_{p,q} = \sum_i \sum_j I(x,y) (x - \bar{x})^p (y - \bar{y})^q \quad (16)$$

gdzie $\bar{x}, \bar{y}$ oznaczają wartości średnie dla współrzędnych pionowej i poziomej (czyli $\bar{x} = m_{10}/m_{00}$, $\bar{y} = m_{01}/m_{00}$)

Wartości momentów centralnych stanowią istotne statystyczne charakterystyki obiektów. Moment μ_{20} określa wariancję punktów obiektu w kierunku poziomym, μ_{02} - wariancję w kierunku pionowym, a μ_{11} określa kowariancję, czyli mierzy statystycznie liniowy związek między współrzędnymi pionowymi i poziomymi punktów.

Z punktu widzenia detekcji okręgów, można sformułować proste kryterium wykorzystujące momenty, pozwalające na zgrubną ocenę podobieństwa konturu do okręgu. Okrąg jest obiektem, dla którego wariancje w kierunkach pionowymi i poziomym są jednakowe oraz nie istnieje liniowa zależność między współrzędnymi punktów. Te właściwości prowadzą do następującego wyrażenia:

$$J^m = \frac{(\mu_{20}/\mu_{02} - 1)^2}{1 - \mu_{11}^2} \quad (17)$$

stanowiącego możliwą miarę podobieństwa konturu do okręgu.

W OpenCV istnieje kilka sposobów pozwalających na wyznaczenie i stosowanie takiego opisu dla porównywania kształtów. Dla wyznaczania momentów dla konturów służy funkcja:

```
cv::Moments cv::moments(cv::InputArray K, bool bin = false)
```

której pierwszym argumentem jest Kontur (lista należących do niego punktów, na przykład `std::vector<cv::Point>`), drugim, zmienna boolowska, która w przypadku, gdy pierwszym argumentem jest wektor punktów, powinna być ustawiona na `false`. Funkcja zwraca strukturę typu `cv::Moments`, której elementami są momenty zwykle rzędów od zerowego do trzeciego, momenty centralne rzędów 1 i 2 oraz momenty centralne znormalizowane.

3 Detekcja obiektów o złożonym wyglądzie

Przedstawiona w poprzedniej części metodologia detekcji dotyczyła najprostszego możliwego obiektu rzeczywistego - kuli o jednorodnej barwie. Obraz kuli jest zawsze kołem (o jednorodnym kolorze, pozwalającym na prostą ekstrakcję), niezależnie od tego, z jakiej perspektywy rejestruje ją kamera. Jedyne źródło zmienności geometrii tego obiektu to, wynikająca z różnej odległości od kamery, zmiana rozmiaru.

Detekcja dowolnego innego obiektu wymaga zasadniczych modyfikacji przedstawionej procedury, o stopniu zależnym od stopnia złożoności jego wyglądu. W szczególności, podstawowym kryterium poszukiwania alternatywnych metodologii jest dokładne sprecyzowanie celu analizy: czy ma być nim detekcja dowolnych obiektów należących do określonej kategorii, czy też detekcja konkretnego obiektu (np. czy próbujemy wykryć w obrazie sylwetki ludzi, a więc dowolne elementy, należące do kategorii 'człowiek', czy wykryć obecność konkretnej osoby). Realizacja pierwszego celu ma charakter badań podstawowych prowadzonych w obszarach sztucznej inteligencji i uczenia maszynowego, wykraczających poza ramy niniejszego opracowania. Dla realizacji drugiego celu opracowano podejścia specyficzne dla obszaru przetwarzania obrazów, które, aby były przydatne w praktyce, muszą zapewnić możliwość rozwiązania następujących problemów, niezbędnych dla poprawnej detekcji:

1. Zmienności rozmiaru obiektu (skala)
2. Możliwych obrotów obiektu (rotacja) oraz, ogólniejszego problemu zmiennej orientacji w przestrzeni obiektu (poza)
3. Możliwego przesłaniania fragmentów obiektu przez inne elementy sceny

Pomyślne rozwiązanie wymienionych problemów jest w ogólnym przypadku bardzo trudne. Najskuteczniejsza z zaproponowanych strategii detekcji obiektów wykorzystuje ich opis za pomocą zbioru punktów charakterystycznych (nazywanych często **punktami kluczowymi** - ang. *keypoints*).

Metody detekcji i rozpoznawania obiektów bazujące na punktach kluczowych są nazywane często metodami bazującymi na cechach lokalnych (ang. *local features*).

Punkty charakterystyczne to takie elementy obrazu, wokół których występuje duża zmienność treści obrazu. Stanowią one istotne elementy pozwalające na identyfikację danego obiektu (np. elementy charakterystyczne twarzy - oczy, usta, nos, brwi - są wyraźnie różne od otaczających je obszarów, a ich występowanie w określonym wzajemnym położeniu upoważnia do stwierdzenia, że zawierający je fragment obrazu odpowiada twarzy). Koncepcja wykorzystania punktów kluczowych jako podstawy analizy treści obrazu ma oparcie w wiedzy o sposobie analizy scen dokonywanej w systemie wzrokowym człowieka - podczas analizy obrazu źrenica wykonuje szybkie 'skoki' między elementami obrazu charakteryzującymi się maksymalną zmiennością.

Zgodnie z przedstawianą koncepcją, obiekt jest reprezentowany z użyciem:

- zbioru znalezionych w nim punktów kluczowych (miejsc, gdzie w wyglądzie obiektu zachodzą duże zmiany)
- skojarzonych z punktami deskryptorów (opisów) lokalnej zmienności treści obrazu w obszarze wokół punktu
- informacji o wzajemnych relacjach przestrzennych między punktami (wzajemne położenia)

Proces detekcji obiektów może być podsumowany w postaci algorytmu, zawierającego następujące etapy:

1. Zbudowanie modelu obiektu

- (a) Wykrycie w obiekcie, który będzie przedmiotem detekcji, punktów charakterystycznych
- (b) Sporządzenie deskryptora dla każdego z wykrytych punktów
- (c) Zbudowanie modelu obiektu, podsumowującego przestrzenne relacje między punktami kluczowymi

2. Detekcja obiektu

- (a) Wykrycie w analizowanym obrazie wszystkich punktów charakterystycznych i sporządzenie ich deskryptorów
- (b) Poszukiwanie dla kolejnych punktów reprezentujących poszukiwany obiekt odpowiadających im punktów w obrazie (na podstawie porównywania ich deskryptorów)
- (c) Dla znalezionych w obrazie odpowiedników sprawdzenie zachodzenia relacji przestrzennych wymaganych przez model obiektu

Kolejne kroki przedstawionej procedury zostaną opisane w dalszej części rozdziału. Należy podkreślić, że reprezentacja obiektów z wykorzystaniem punktów charakterystycznych jest stosowana również dla realizacji innych, ważnych zadań analizy obrazu, z których najistotniejszymi są śledzenie obiektów i stereowizja.

3.1 Detekcja punktów charakterystycznych

Ponieważ punkty charakterystyczne zostały zdefiniowane jako piksele obrazu, wokół których występuje silna zmienność treści obrazu (poziomów jasności), naturalnym sposobem ich poszukiwania byłoby przekształcenie obrazu w sposób uwypuklający jego lokalną zmienność, a więc obliczanie pochodnej przestrzennej obrazu. Z uwagi na dwuwymiarową strukturę obrazu, do reprezentacji zmienności obrazu w każdym punkcie stosowany jest gradient, a więc wektor, zawierający dwa elementy, odpowiadające pochodnym wyznaczonym dla dwóch prostopadłych kierunków analizy (w praktyce - poziomym i pionowym):

$$\mathbf{g}_{ij} = \begin{bmatrix} g_{ij}^H \\ g_{ij}^V \end{bmatrix} = \begin{bmatrix} \frac{\partial I(i,j)}{\partial j} \\ \frac{\partial I(i,j)}{\partial i} \end{bmatrix} \quad (18)$$

gdzie i, j to współrzędne wiersza i kolumny obrazu, w której obliczany jest gradient, $I(i, j)$ to poziom jasności obrazu w rozważanym punkcie, zaś H, V oznaczają odpowiednio komponenty poziomy i pionowy gradientu.

Wyznaczenie pochodnych pionowych i poziomych jest dokonywana, jak wspomniano w rozdziale 2.2 za pomocą filtrów Sobela określonych zależnościami (??) i () i w efekcie, otrzymuje się dwa obrazy gradientów (poziomych - rys. x i pionowych - rys.). Wszystkie punkty obrazu, w których moduł gradientu $g_{i,j}$:

$$g_{i,j} = \sqrt{(g_{ij}^H)^2 + (g_{ij}^V)^2}$$

jest dostatecznie duży (przekracza przyjętą za miarę istotności) wartość progową) odpowiadają miejscom, w których następują istotne zmiany treści obrazu. Jednak, dla uznania piksela obrazu za punkt charakterystyczny, nie wystarczy spełnienie warunku odpowiedniej wartości gradientu: jest to jedynie warunek konieczny. Test odpowiednio dużej wartości modułu gradientu był podstawą określania punktów należących do krawędzi obiektów, zaś punkt kluczowy to szczególna lokalizacja obiektu - obiekt ma być dobrze opisany za pomocą ich niewielkiej liczby, a nie setek lub tysięcy punktów brzegowych: dla wykrycia w obrazie twarzy wystarczy wykrycie kilku jej cech - oczu (wskazywanych przez punkty umieszczone np. w środku źrenicy), lub ust (wskazywanych np. przez ich kąciaki), i nie jest do tego potrzebna znajomość wszystkich elementów konturów ust, oczu itp.). Dlatego też definicja punktu kluczowego musi być dodatkowo uściślona: wymagamy by nie tylko wartość gradientu w takim punkcie była duża, ale również, by w takim

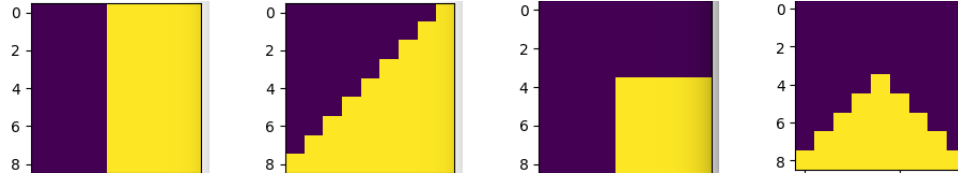
punkcie duża była 'krzywizna' takiego konturu. To doprecyzowanie definicji punktu kluczowego zostało wprowadzone w pierwszej metodzie analizy obrazu, bazującej na cechach lokalnych, gdzie punkt kluczowy został utożsamiony z narożnikiem. W późniejszych pracach sposób rozumienia punktu charakterystycznego został uogólniony, ale podstawowe idee detekcji takich punktów bazują na koncepcji detekcji narożników zaproponowanej przez Harrisa w [?] (tzw. **detektor Harrisa**).

Analityczne kryterium selekcji punktów, w których kontury mają maksymalną krzywiznę bazuje na elementarnej analizie matematycznej. Miarą krzywizny funkcji 1D jest druga pochodna tej funkcji (druga pochodna opisuje zmienność zmienności). Ponieważ obraz jest funkcją dwuwymiarową, druga pochodna staje się macierzą (obliczane są pochodne poziome i pionowe, zarówno dla gradientów poziomych jak i pionowych), nazywaną **Hesjanem** (lub macierzą Hessa):

$$\mathbf{H}_{i,j} = \begin{bmatrix} \frac{\partial g_{i,j}^H}{\partial i} & \frac{\partial g_{i,j}^H}{\partial j} \\ \frac{\partial g_{i,j}^V}{\partial i} & \frac{\partial g_{i,j}^V}{\partial j} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 I(i,j)}{\partial i^2} & \frac{\partial^2 I(i,j)}{\partial i \partial j} \\ \frac{\partial^2 I(i,j)}{\partial i \partial j} & \frac{\partial^2 I(i,j)}{\partial j^2} \end{bmatrix}$$

O ile w przypadku 1D detekcja punktu o największej krzywiznie funkcji była prosta - było to maksimum drugiej pochodnej, o tyle przypadek 2D jest bardziej skomplikowany. Ponieważ informacja o drugich pochodnych zawarta jest w macierzy, konieczne jest określenie skalarnej miary, która będzie dobrze odzwierciedlać krzywiznę w punkcie. Proste uogólnienie kryterium stosowanego dla przypadku 1D prowadziło by do próby oceny wartości np. sumy modułów Hesjanu (albo ich kwadratów, czyli normy Frobeniusa macierzy) i porównania, czy przekraczają zadany próg. Podstawą dla takiego rozumowania byłoby założenie, że jeżeli duża krzywizna dla przypadku 1D odpowiada dużej wartości drugiej pochodnej, tego samego należałoby się spodziewać w dwóch wymiarach. Aby wykazać błędność takiego rozumowania, rozważmy następujący przykład. Załóżmy, że dane są fragmenty obrazu, zawierające krawędzie: poziomą i diagonalną oraz narożnik w dwóch różnych orientacjach (rys.10). Fragmenty macierzy obrazu w otoczeniu centralnego punktu obrazu mają odpowiednio postać:

$$\mathbf{I}^V = \begin{bmatrix} 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \end{bmatrix} \quad \mathbf{I}^D = \begin{bmatrix} 0 & 0 & 0 & 0 & 200 \\ 0 & 0 & 0 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 200 & 200 & 200 & 200 \\ 200 & 200 & 200 & 200 & 200 \end{bmatrix}$$



Rysunek 10: Fragmenty krawędzi (pionowa i diagonalna) oraz narożnik w dwóch orientacjach. Kolor żółty odpowiada jasności '200', fioletowy - jasności '0'.

$$\mathbf{I}^{C1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \\ 0 & 0 & 200 & 200 & 200 \end{bmatrix} \quad \mathbf{I}^{C2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200 & 0 & 0 \\ 0 & 200 & 200 & 200 & 0 \\ 200 & 200 & 200 & 200 & 200 \end{bmatrix}$$

Hesjany, wyliczone dla centralnych punktów przedstawionych macierzy wynoszą odpowiednio:

$$\mathbf{H}^V = \begin{bmatrix} -200 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{H}^D = \begin{bmatrix} -200 & -200 \\ -200 & -200 \end{bmatrix}$$

$$\mathbf{H}^{C1} = \begin{bmatrix} -200 & 200 \\ 200 & -200 \end{bmatrix} \quad \mathbf{H}^{C2} = \begin{bmatrix} -400 & 0 \\ 0 & -200 \end{bmatrix}$$

Gdyby do oceny krzywizny wykorzystać prostą sumę modułów elementów Hesjanu lub jego normę Frobeniusa, jako punkty o największej krzywiznie zostałyby wybrane centralne punkty dwóch środkowych obrazów z rys. 10.

Struktura w istocie jednowymiarowa

Metodą oceny, czy struktura jest jedno- czy dwuwymiarowa jest dekompozycja macierzy z wykorzystaniem wektorów własnych i sprawdzenie relacji między jej wartościami własnymi. Wektory własne macierzy stanowią podstawę do jej budowy - wykorzystanie 'ważniejszego' wektora własnego daje zgrubną aproksymację macierzy, a aproksymacja jest dopasowywana do właściwej postaci z wykorzystaniem drugiego wektora własnego. Miarą istotności komponentów są wartości własne. Jeżeli pierwsza z nich jest znacząco większa od drugiej, oznacza to, że praktycznie cała informacja o drugiej pochodnej może być zrekonstruowana na podstawie analizy zmienności zachodzącej w jednym kierunku, a więc struktura obiektu w otoczeniu danego punktu jest w istocie jednowymiarowa. Natomiast jeżeli obydwie wartości własne będą do siebie zbliżone, oznacza to dwuwymiarowy charakter zmienności - można ją prawidłowo odtworzyć tylko, gdy rozważy się zmienność w dwóch ortogonalnych kierunkach. Dla rozważanych przykładów, wartości własne wynoszą odpowiednio:

3.2 Opis punktów charakterystycznych

3.3 Dopasowywanie deskryptorów punktów kluczowych

3.4 Modelowanie obiektu i detekcja obiektu

4 Detekcja krawędzi w obrazach monochromatycznych

Krawędzie to obszary obrazu charakteryzujące się silną zmiennością jasności, zwykle informujące o granicach obiektów znajdujących się w obrazie. Z tego powodu detekcja krawędzi jest jedną z najbardziej istotnych analiz obrazu.

Podstawą detekcji krawędzi jest elementarna wiedza z analizy matematycznej. Rozważmy funkcję 1D, opisującą zmiany intensywności obrazu wzdłuż wybranego wiersza i oznaczmy ją $f(x)$. Naturalnym sposobem poszukiwania krawędzi jest analiza zmienności funkcji, a więc, analiza pochodnej tej funkcji (pochodna to zmiana w funkcji odpowiadająca zmianie argumentu). Krawędzie obiektów to obszary o silnej zmienności wartości funkcji, a więc obszary w których moduł pochodnej jest duży (przekracza pewną założoną wartość progową. Tak więc, punkt obrazu x_0 jest punktem krawędzi (należy do zbioru $\mathcal{K}$), gdy:

$$x_0 \in \mathcal{K} \rightarrow \left| \frac{df(x_0)}{dx} \right| > \theta$$

W świecie obliczeń numerycznych konieczne jest dokonanie aproksymacji pochodnej funkcji, co może być dokonane na kilka sposobów: jako 'różnica centralna', czyli wyrażenie:

$$\frac{df(x_0)}{dx} \approx \frac{f(j+1) - f(j-1)}{2}$$

lub 'różnica wsteczna':

$$\frac{df(x_0)}{dx} \approx \frac{f(j+1) - f(j)}{2}$$

gdzie j to indeks elementu wiersza (kolumny obrazu), w którym obliczana jest pochodna. Ponieważ przedmiotem zainteresowania analizy są nie bezwzględne wartości pochodnych a ich wartości względne, powyższe wyrażenie jest najczęściej upraszczane poprzez usuwanie mianownika (zamianę kroku 2 na krok 1):

$$\frac{df(x_0)}{dx} \approx f(j+1) - f(j-1)$$

Przedstawione wyrażenie może być wyrażone jako wynik wykonania operacji iloczynu skalarnego dwóch wektorów: pierwszy z nich odpowiada fragmentowi funkcji $f(\cdot)$ wokół lokalizacji $j - [f(j-1), f(j), f(j+1)]$, a drugi -

wektorowi współczynników $\mathbf{h}$ o postaci: $\mathbf{h} = [-1, 0, 1]$ (dla różnicy centralnej) lub $\mathbf{h} = [-1, 1, 0]$ (dla różnicy wstecznej):

$$g_j = 1 \cdot f(j+1) + 0 \cdot f(j) + (-1) \cdot f(j-1)$$

$$g_j = -1 \cdot f(j-1) + 1 \cdot f(j) + 0 \cdot f(j+1)$$

Wektor wyników obliczania pochodnej dla kolejnych punktów rozważanego wiersza obrazu będzie zawierać wyniki wykonywania określonych powyżej iloczynów skalarnych, w których wektor $\mathbf{h}$ jest stały, a drugi argument zmienia się w zależności od treści obrazu w wierszu (różne wartości $f(\cdot)$). Obliczenie pochodnej odbywa się więc poprzez wykonywanie operacji na kolejnych podciągach wektora wejściowego, zgodnie z regułą zakodowaną w wektorze $\mathbf{h}$. Wektor ten można więc nazwać 'operatorem' przekształcenia, a operacja, której wynik to wyznaczenie odpowiedniego iloczynu skalarnego, nazywa się operacją korelacji:

$$\mathbf{y} = \mathbf{f} \odot \mathbf{h} : \quad y_j = \sum_{l=-r}^r f_{j+l} h_l$$

Jeżeli wektor $\mathbf{h}$ jest przekształconym poprzez odbicie zwierciadlane względem środkowego elementu wektorem $\mathbf{H}$ (tzn. elementy w $\mathbf{H}$ są ustawione w odwrotnej kolejności niż w $\mathbf{h}$), wtedy operacja wyznaczenia wartości pochodnej może zostać wyznaczona w alternatywnej formie:

$$\mathbf{y} = \mathbf{H} * \mathbf{f} : \quad y_j = \sum_{l=-r}^r f_{j-l} H_l \quad (19)$$

Operacja (19) nazywa się operacją **splotu** (lub **konwolucji**) i jest jedną z podstawowych operacji przetwarzania danych dyskretnych. Operator $\mathbf{H}$ koduje regułę, zgodnie z którą odbywa się przetwarzanie i zależnie od jego postaci, przetwarzanie będzie uwypuklać różne właściwości badanej funkcji. Przykładowo, operator obliczania drugiej pochodnej funkcji można zbudować w wyniku złożenia dwóch operatorów pochodnej - dla różnicy wstecznej, będzie miał on więc postać:

$$\mathbf{H} = [-1, 2, -1]$$

co można łatwo sprawdzić przeprowadzając operację splotu operatorów różnicy wstecznej. Operator drugiej pochodnej jest nazywany operatorem Laplace'a i służy do detekcji punktów osobliwych funkcji.

Jedną z typowych cech obrazów rzeczywistych jest obecność zakłóceń (*szumu*), objawiających się przypadkowymi, zwykle niewielkimi wahaniami wartości poziomów jasności kolejnych punktów obrazu. Przykładowo, fragment wiersza obrazu, odpowiadający jednorodnej powierzchni o jasności np.

100 nie będzie reprezentowany przez ciąg liczb [..., 100, 100, 100, 100, ...], ale przez ciąg liczb [... 97, 102, 100, 98, ...] (różnice mogą wynikać np. z losowych rozrzutów współczynników konwersji natężenia oświetlenia na napięcia w kolejnych elementach sensora światła). Wyznaczenie pochodnej dla takiego rzeczywistego fragmentu obrazu da wartości, które mogą zostać mylnie zinterpretowane jako krawędzie - zamiast wyniku [..., 0, 0, 0,...] dostaniemy [..., 5, -2, -2,...], a jeśli próg detekcji krawędzi zostanie ustalony na 5, wykryta zostanie 'fałszywa' krawędź. Dlatego też, w przypadku detekcji w obrazach rzeczywistych, operacja obliczania pochodnych jest poprzedzona operacją uśredniania, redukującą przypadkowe wahania wartości. Możliwa realizacja operatora uśredniania to wektor $0.25 * [1 \ 2 \ 1]$. Splot fragmentu rozważanej funkcji z tą maską daje (po zaokrągleniu do liczby całkowitej) wynikowy wektor: [..., 100, 100, ...]. Ponieważ obraz to struktura dwuwymiarowa, zamiast uśredniania wzdłuż wiersza, a następnie, wyznaczania pochodnej wzdłuż wiersza, w celu określenia gradientu poziomego, najpierw dokonywane jest uśrednianie obrazu wzdłuż kolumny, a później - obliczenie pochodnej wzdłuż wiersza (zakłócenia są losowe, więc uśrednienie wzdłuż kolumny jest tak samo dobrze uzasadnione jak uśrednienie wzdłuż wiersza). Ponieważ splot jest operacją liniową, obowiązuje dla niego prawo łączności operacji, co oznacza, że procedura przekształcenia obrazu może być zapisana jako:

$$\mathbf{y} = \mathbf{H}_{AVG_Y} * \mathbf{H}_{\Delta_X} * \mathbf{f} = (\mathbf{H}_{AVG_Y} * \mathbf{H}_{\Delta_X}) * \mathbf{f} = \mathbf{S}_H * \mathbf{f}$$

a więc jest to przekształcenie obrazu, wykorzystujące jedną maskę (tym razem 2D), nazywaną poziomym operatorem Sobela, o postaci:

$$\mathbf{S}_H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Wynik przekształcenia obrazu za pomocą poziomej maski Sobela to obliczenie pochodnej w kierunku poziomym, połączone z uśrednianiem treści obrazu w kierunku pionowym. Analogicznie, dla obliczenia pochodnej funkcji w kierunku pionowym stosuje się operator Sobela gradientu pionowego:

$$\mathbf{S}_V = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

4.1 Detekcja krawędzi - Canny

Jednym z najczęściej stosowanych obecnie algorytmów detekcji krawędzi jest metoda zaproponowana przez Canny'ego, która składa się z dwóch zasadniczych faz. Pierwsza z nich to ekstrakcja z obrazu punktów, których moduł gradientu przekracza przyjętą wartość progową θ_1 . Komponenty pionowy i poziomy gradientu obrazu są obliczane z wykorzystaniem przedstawionych

masek Sobela. Druga faza algorytmu to 'uzupełnianie konturów' - ponieważ wzdłuż granic obiektów gradienty mogą się istotnie zmieniać, zastosowanie wyłącznie pierwszej fazy detekcji może skutkować wydzieleniem niekompletnych konturów. Uzupełnianie konturów polega na poszukiwaniu punktów obrazu, dla których gradienty przekraczają wartość mniejszej, drugiej wartości progowej θ_2 , przy czym punktami startowymi poszukiwania są tylko końce konturu wydzielonego w pierwszej fazie. Pozwala to na zmniejszenie wrażliwości procesu detekcji konturu na wspomniane niejednorodności kontrastu obszarów znajdujących się na granicy obiektów.

Narzędzie OpenCV pozwalające na realizację detekcji konturu metodą Canny'ego to funkcja `out = cv2.Canny(in, th1, th2 [, r[, L2/L1]])`, gdzie parametry to odpowiednio: analizowany obraz, progi θ_1 i θ_2 , oraz opcjonalnie, rozmiar operatora Sobela i wybór normy dla obliczania gradientu.

Literatura

- [1] <https://msdn.microsoft.com>