Konrad Szwedo

Album: 98123

TeleInformatyka

**Narzędzie do tworzenia cech typu Haar’a.**

* SAT + RSAT **(R. Lienhart)**
* Aplikacja z GUI umożliwiająca tworzenie cech typu Haar’a (cechy tworzone poprzez zaznaczanie prostokątów przy użyciu myszki)

# Cechy typu Haar’a

Cechy typu Haar’a są cechami opierającymi się na prostokątnych obszarach obrazów cyfrowych. Ich nazwa wywodzi się z podobieństwa do falek Haar’a, dlatego też w literaturze przyjęło się używać tej nazwy. Podstawową zaletą operowania na cechach typu Haar’a jest ich zdolność do zawierania wiedzy o opisywanych przez nie obszarach, takiej jak np. występowanie krawędzi lub linii. Dodatkową zaletą jest szybkość obliczania ich wartości.

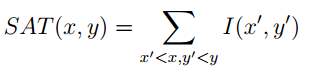
Cechy typu Haar’a definiowane są przez:

* Współrzędne cechy względem punktu startowego,
* Rozmiar cechy – szerokość oraz wysokość.

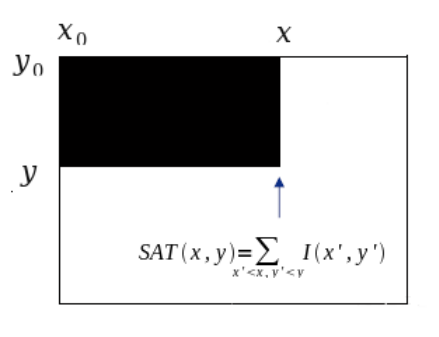
Wartość każdej cechy jest obliczana jako różnica sumy poziomów szarości pikseli pokrywanych przez biały oraz czarny prostokąt oraz sumy poziomów szarości pikseli pokrywanych przez czarny prostokąt. Składniki tej różnicy posiadają wagi odwrotnie proporcjonalne do rozmiarów, dzięki czemu różnice wielkości dwóch obszarów są kompensowane.

# Obraz scałkowany

Powtarzanie operacji obliczania wartości cechy typu Haar’a przez sumowanie poziomów szarości obrazu doprowadziłoby do znacznego wydłużenia procesu detekcji. Celem jej przyspieszenia Viola wprowadził pojęcie obrazu scałkowanego (ang. integral image). W tym celu obliczana jest wartość funkcji SAT (ang. Summed Area Table) oraz RSAT (ang. Rotated Summed Area Table) dla współrzędnych (x,y) każdego piksela, które przekształcają dany obraz w obraz scałkowany zgodnie ze wzorem:

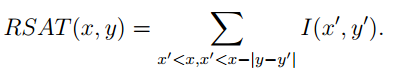


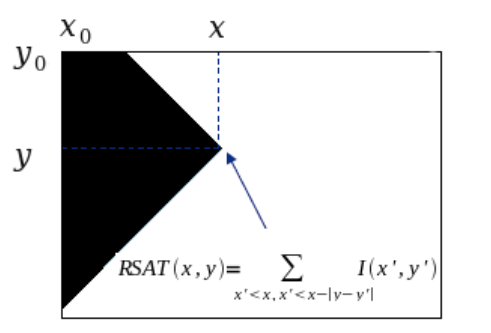
Gdzie:

 – jest wartością poziomu szarości piksela w punkcie .

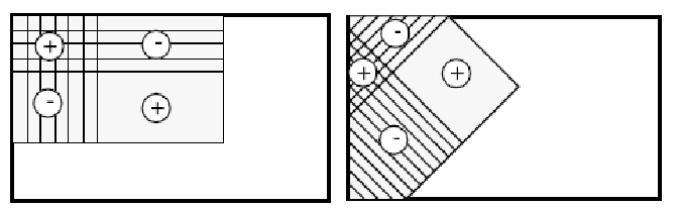
Funkcja SAT

Tablica RSAT wyznaczana jest za pomocą poniższego wzoru:



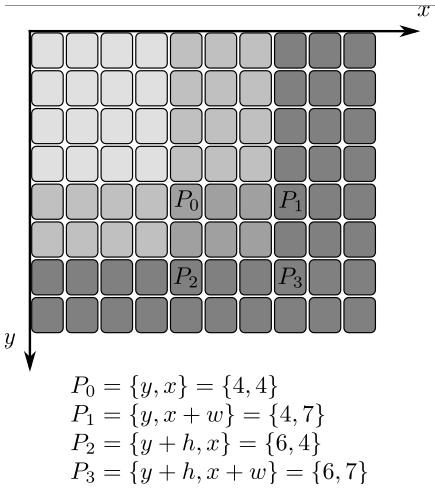
Funkcja RSAT

Dzięki zastosowaniu funkcji SAT oraz RSAT suma poziomów jasności pikseli obejmujących dowolny prostokątny obszar jest obliczana w stałym czasie. Wartość sumy pikseli powstaje poprzez proste dwie operacje dodawania oraz dwie operacje odejmowania.



Do utworzenia 3 tablic Integral Image’ów została wykorzystana konwencja z biblioteki OpenCV. Oznacza to że, każda z trzech tablic jest rozmiaru (w + 1, h + 1). Gdzie pierwszy wiersz, oraz pierwszy element każdej kolumny wypełniane są zerami. W przypadku RSAT pierwszy wiersz wypełniany jest zerami a pierwszy element kolumny jest wypełniany wartością RSAT(y-1, x+1). Dzięki takiemu podejściu obraz scałkowany jest ściślejszy z matematycznego punktu widzenia, dla przykładu: Gdy mamy wiersz posiadający 3 pixele z wartościami: 1, liczba skumulowanych sum wynosi 4, ponieważ możemy uzyskać sumę pierwszych 0, 1, 2, lub wszystkich 3 pixeli. Wobec tego liczba różnych sum wynosi: 4 (0,1,2,3).

Drugą przyczyną dla którego tworzymy integralImage z paddingiem jest uproszczenie oraz przyśpieszenie obliczeń. Obraz scałkowany jest używany do obliczenia wartości sumy każdego możliwego prostokąta (jest to suma dwóch punktów narożnych minus suma dwóch innych punktów narożnych). Odległość pomiędzy punktami narożnymi jest dokładnie wielkością prostokąta, dla którego chcemy obliczyć sumę. Dla przykładu, jeżeli rozpatrujemy prostokąt o szerokości 5 pixeli wtedy odwołujemy się do punktów SAT[y][x] oraz SAT[y][x+5]. W przypadku, gdy prostokąt zajmuję dokładnie cały obraz, SAT bez paddingu może powodować wykroczenie indexów o 1, żeby temu zapobiec metoda pobierająca wartość sumy dla prostokąta musi sprawdzać indeksy a co za tym idzie będzie wolniejsza, ponieważ musi używać warunków if które są sprawdzają czy indeks jest w przedziale co jest kosztowniejsze niż 4 pobrania z tablicy.



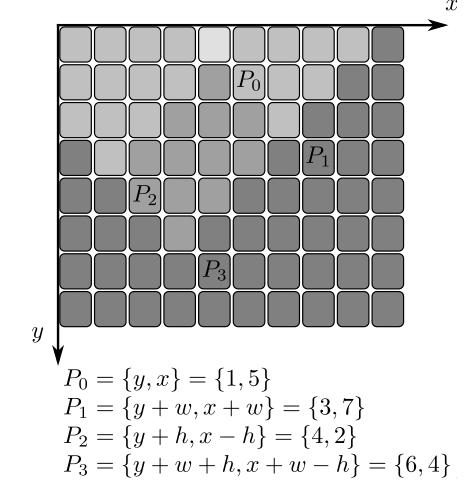
Wartość sumy pikseli dla danego prostokąta to:

**+**SAT(P0)

**+**SAT(P3)

**-**SAT(P1)

**-**SAT(P2)

****

Wartość sumy pikseli dla danego obróconego prostokąta to:

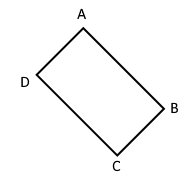
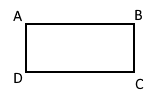
**+**RSAT(P0)

**+**RSAT(P3)

**-**RSAT(P1)

**-**RSAT(P2)

**Oznaczenia przyjęte w programie:**

Oznaczenia punktów dla nieobróconego prostokąta. Oznaczenia punktów dla obróconego prostokąta.

Punkt startowy:

P0 (P0x, P0Y)

Zestaw opisujący cechę typu Haar’a składa się z pozycji wspólnego punktu startowego P0 dla wszystkich prostokątów oraz:

Parametry opisu prostokątów:

* **x** [Element] Odległość na osi X od punktu startowego wartość (Ax - P0x)
* **y** [Element] Odległość na osi Y od punktu startowego wartość (Ay - P0y)
* **width** [Element] Szerokość prostokąta (wartość Bx – Ax)
* **height** [Element] Wysokość prostokąta (wartość Ay – Dy)
* **isRotated** [Atrybut] Przyjmuje wartość „true” dla obróconych prostokątów

# Funkcjonalność

Pierwszym krokiem działania programu jest otworzenie obrazu wejściowego (domyślnie otwiera obraz sample.jpg znajdujący się w katalogu startowym programu). Następnie program konwertuje obraz do skali szarości, tak aby pojedynczy piksel opisany był wartością z przedziału [0..255], gdzie 0 oznacza kolor czarny a 255 kolor biały. Dla wczytanego obrazu, aplikacja oblicza 3 obrazy scałkowane (suma pikseli, suma kwadratów pikseli, suma obróconych pikseli). Rysowany jest prostokąt o wymiarach (w,h : 100, 50) oraz punkt startowy i umieszczane są na obszarze rysowania.

Użytkownik może zmienić jego pozycję ruchem przeciągania elementu znanym z różnych programów komputerowych. Najeżdżając na krawędź może zmienić szerokość, wysokość danego prostokąta. Istnieje także możliwość zmiany pozycji punktu początkowego poprzez analogiczny ruch myszką. Użytkownik może także przypisać kolor wypełnienia obszaru prostokąta w celu szybkiej identyfikacji wzrokowej. Dla zaznaczonego prostokąta na liście prostokątów znajdującej się po prawej stronie interfejsu, aktualizowane są na bieżąco położenia poszczególnych punktów prostokąta oraz jego parametry takie jak ilość pikseli w obszarze, suma pikseli, szerokość wysokość.

Po skończonej pracy dopasowania punktu startowego i prostokątów użytkownik może zapisać parametry do pliku w formacie XML.

<?xml version=**"1.0"** encoding=**"UTF-8"** standalone=**"yes"**?>

<HaarFeature>

<initialPointX>**132**</initialPointX>

<getInitialPointY>**110**</getInitialPointY>

<rect isRotated=**"false"**>

<x>**163**</x>

<y>**-93**</y>

<width>**100**</width>

<height>**50**</height>

</rect>

<rect isRotated=**"true"**>

<x>**-86**</x>

<y>**14**</y>

<width>**100**</width>

<height>**50**</height>

</rect>

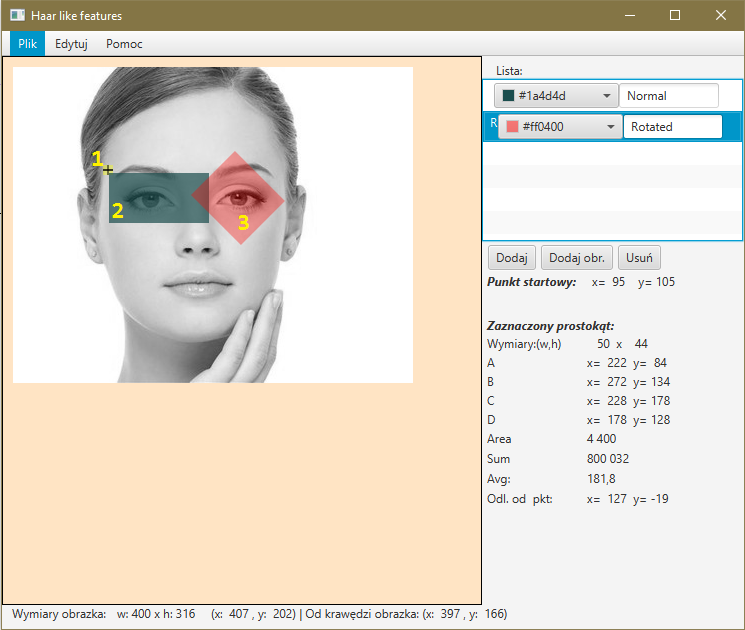
</HaarFeature>

Przykładowy plik wynikowy XML

# Wykorzystane narzędzia

Aplikacja została stworzona przy użyciu platformy Java w wersji 8. Framework tworzący interfejs użytkownika to JavaFX 8. Do obliczania Integral image’ów została stworzona metoda obliczająca 3 obrazy scałkowane, opisane w punkcie 3. Tworzy ona wartości integral image’ów dla sumy pikseli, kwadratów sum pikseli, sumy obróconych pikseli o kąt 45.

# Interfejs programu



Na zrzucie ekranu zaznaczone są:

1. Punkt startowy
2. Prostokąt
3. Prostokąt obrócony

Interfejs użytkownika składa się z 5 głównych części:

1. Obszar obrazu. Posiada wymiary takie jak maksymalna wielkość rysowanego obrazu. Na nim użytkownik przemieszcza utworzone prostokąty.
2. Lista stworzonych prostokątów. Ta część posiada kontrolki do tworzenia zwykłych i obróconych prostokątów, nadawania im nazwy oraz koloru który umożliwia szybszą identyfikację wzrokową.
3. Panel informacji o prostokątach. Wyświetlane dane to:
   1. Dane o punkcie startowym współrzędne X, Y
   2. Dane prostokąta
      1. Opis wymiarów (w, h) (Szerokość, Wysokość) w px
      2. Powierzchnia prostokąta (Area)
      3. Suma wartości pikseli dla całego obszaru (Sum)
      4. Średnia jasności pikseli dla obszaru (Avg)
      5. Odległość punktu prostokąta A od punktu startowego.
4. StatusBar. Wyświetla wymiary obrazka oraz pomocnicze wartości położenia kursora w obszarze obrazu.
5. Górna belka menu. Zawiera akcję operacji:
   1. Plik
      1. Otwórz obraz. (Otwiera nowy obraz, tworzy z niego obraz 8 bitowy w skali szarości a następnie wyświetla w obszarze rysowania)
      2. Zapisz listę cech. (Zapisuje listę cech jako plik xml, wybrany przez użytkownika)
      3. Wyjście
   2. Edycja
      1. Tworzy nową domyślny prostokąt.
      2. Tworzy nowy obrócony prostokąt.
      3. Usuwa prostokąt z listy.

Bibliografia:

1. An Extended Set of Haar-like Features for Rapid Object Detection – **R. Lienhart, J. Maydt** (2002)
2. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object  
   Detection – **R. Lienhart, A. Kuranov, V. Pisarevsky** (2002)