

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE**

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I ELEKTRONIKI



PRACA MAGISTERSKA

Mirosław Słysz

**„System śledzenia oczu, twarzy i ruchów
użytkownika komputera za pośrednictwem kamery
internetowej i pozycjonujący oczy cyberagenta
internetowego na oczach i akcjach użytkownika”**

Opiekun pracy: dr Adrian Horzyk

Kraków, 2009

Oświadczenie autora pracy

Oświadczam, świadomy odpowiedzialności karnej za
poświadczenie nieprawdy, że niniejszą pracę dyplomową
wykonałem osobiście i samodzielnie i nie korzystałem ze źródeł
innych niż wymienione w pracy.

.....
podpis

AGH

UNIVERSITY OF SCIENCE AND TECHNOLOGY IN KRAKOW

FACULTY OF ELECTRICAL ENGINEERING, AUTOMATICS, COMPUTER SCIENCE AND ELECTRONICS



MASTER OF SCIENCE THESIS

Mirosław Słysz

“System for eye, face and movement tracking using internet camera and positioning eyes of cyberagent on eyes and actions of computer user”

Supervisor: dr Adrian Horzyk

Krakow, 2009

Niniejszą pracę pragnę zadedykować moim wspaniałym, Rodzicom **Zofii i Edwardowi Słyszom**, dzięki którym miałem możliwość kształcić się i zdobywać cenną wiedzę, którzy stale mnie mobilizowali i wspierali przez okres trwania studiów na Akademii Górniczo-Hutniczej.

Za inspirację, wyrozumiałość oraz pomoc przy realizacji tej pracy pragnę złożyć serdecznie podziękowania mojemu promotorowi **dr Adrianowi Horzykowi**.

SPIS TREŚCI

1. WSTĘP	7
2. PRZEGLĄD ISTNIEJĄCYCH ROZWIĄZAŃ.....	10
2.1. METODY LOKALIZACJI TWARZY.....	10
2.1.1. <i>Metody oparte na porównywaniu kolorów.....</i>	10
2.1.2. <i>Metody oparte na dopasowywaniu schematów</i>	12
2.1.3. <i>Metody oparte na cechach.....</i>	14
2.1.3.1. Integral image.....	15
2.2. WYKRYWANIE MIMIKI.....	17
2.3. ANIMACJA POSTACI	18
3. OPIS ROZWIĄZANIA.....	20
3.1. LOKALIZACJA TWARZY.....	21
3.1.1. <i>Obraz w skali szarości.....</i>	21
3.1.2. <i>Obraz binarny.....</i>	21
3.1.3. <i>Schemat algorytmu lokalizacji twarzy.....</i>	24
3.1.4. <i>Szczegółowy opis klasyfikatorów</i>	26
3.1.4.1. Analiza wartości klatki (obraz binarny)	27
3.1.4.2. Analiza wartości klatki (obraz w skali szarości).....	28
3.1.4.3. Analiza dużego obszaru oczu oraz pod oczami (obraz binarny).....	29
3.1.4.4. Analiza małych obszarów oczu oraz pod oczami (obraz binarny)	30
3.1.4.5. Analiza małych obszarów oczu oraz pod oczami (obraz w skali szarości).....	32
3.1.4.6. Analiza lewych ćwiartek klatki (obraz w skali szarości)	32
3.1.4.7. Analiza prawych ćwiartek klatki (obraz w skali szarości)	33
3.1.4.8. Analiza pionowych połówek klatki (obraz w skali szarości)	34
3.1.4.9. Analiza poziomych połówek klatki (obraz w skali szarości)	34
3.1.4.10. Analiza wartości obszarów oczu i brwi (obraz binarny).....	35
3.1.4.11. Analiza obszarów policzków (obraz binarny).....	37
3.1.4.12. Klasyfikator sumaryczny nr 1.....	38
3.1.4.13. Analiza regionów oczu (obraz binarny)	40
3.1.4.14. Analiza regionu nosa (obraz binarny)	41
3.1.4.15. Analiza obszaru ust (obraz binarny).....	42
3.1.4.16. Analiza różnicy wartości obszaru ust i policzków (obraz binarny).....	43
3.1.4.17. Klasyfikator sumaryczny nr 2.....	43
3.1.4.18. Analiza zewnętrznych części obszarów oczu - w poziomie (obraz binarny)	44
3.1.4.19. Analiza zewnętrznych części obszarów oczu - w pionie (obraz binarny)	45
3.1.4.20. Analiza zewnętrznych części obszaru ust – w pionie (obraz binarny).....	45
3.1.4.21. Analiza bocznych krawędzi ramki (obraz binarny).....	46
3.1.4.22. Analiza obszaru oczu i pomiędzy oczami (obraz binarny).....	48
3.1.4.23. Analiza obszaru oczu oraz czoła (obraz binarny).....	49
3.1.4.24. Analiza obszaru oczu i brwi (obraz binarny).....	51
3.1.4.25. Klasyfikator sumaryczny nr 3.....	52
3.1.5. <i>Wybór optymalnej klatki dla pojedynczej twarzy</i>	53
3.2. LOKALIZACJA TWARZY W CZASIE RZECZYWISTYM	53
3.2.1. <i>Wykrywanie ruchu</i>	54
3.3. MIMIKA TWARZY	56

3.3.1.	<i>Lokalizacja ust</i>	56
3.3.2.	<i>Wykrywanie kształtu ust</i>	58
3.3.3.	<i>Lokalizacja oczu</i>	61
3.3.4.	<i>Określanie stanu oczu</i>	64
3.4.	ANIMACJA TWARZY.....	65
3.4.1.	<i>Ruch oczu</i>	66
3.4.2.	<i>Uśmiech zamknięty</i>	66
3.4.3.	<i>Uśmiech otwarty</i>	67
3.4.4.	<i>Animacja przejścia pomiędzy typami uśmiechu</i>	67
3.4.5.	<i>Mrugnięcie zwykłe</i>	68
3.4.6.	<i>Mrugnięcie specjalne</i>	68
3.5.	APLIKACJA.....	69
4.	TESTY OPRACOWANEGO ROZWIĄZANIA.....	69
4.1.	LOKALIZACJA TWARZY.....	69
4.2.	LOKALIZACJA UST I OCZU.....	71
4.3.	WYKRYWANIE MIMIKI.....	71
5.	PODSUMOWANIE.....	76
	BIBLIOGRAFIA.....	78

1. Wstęp

W czasach stałego rozwoju technologicznego maszyny stały się nieodzowną częścią życia człowieka. Większość wykonywanych przez nas czynności wspomagane jest przez wszelakie urządzenia. Człowiek z natury nie jest w stanie komunikować się z maszyną. Sposób pośrednictwa w relacjach człowiek-maszyna stał się więc bardzo ważnym zagadnieniem. Tak zwane interfejsy użytkownika czyli specjalnie stworzone oprogramowanie i zespół urządzeń, są stale urozmaicane i rozwijane. Interfejsy znane również jako HMI (ang. *Human-Machine Interface*) składają się z dwóch części – wejścia i wyjścia. Systemy wejścia umożliwiają kontrolę oraz manipulację odpowiednimi funkcjami lub procesami działania maszyny. Część wyjściowa odpowiada za informowanie użytkownika o efektach wydanych poleceń lub wykonanych na maszynie operacji.

Można zauważyć, że nowo powstające metody komunikacji z maszynami, a w tym z komputerem są coraz wygodniejsze i prostsze w użyciu. W wielu istniejących systemach wykorzystywane są elementy sztucznej inteligencji. Program nawiązuje pewien „kontakt” z użytkownikiem, ucząc się jego zwyczajów i często automatycznie sugeruje pewne rozwiązania.

Stale rozrastająca się globalna sieć Internet jest bardzo bogata w systemy wykorzystujące wszelkiego rodzaju algorytmy imitujące inteligencję. Przykładem są automatyczne systemy kontaktu z klientami CRM (ang. *Customer Relationship Management*). Programy danego typu oparte są na dużych bazach wiedzy. W przypadku obsługi klienta generowane są całe serie zapytań zależnych od wcześniej uzyskanych odpowiedzi. Metodologia danego typu ma zapewnić użytkownikowi jak najszybsze uzyskanie pożądanego efektu, a tym samym zapewnienie dużej satysfakcji.

Duży nacisk kładziony jest na badania nad rozwojem całej gamy oprogramowania symulującego zachowanie człowieka. Do takich systemów zaliczyć można chatboty (ang. chat – rozmowa) – programy umożliwiającego prowadzenie rozmowy z komputerem przy użyciu języka naturalnego lub tekstowego. Rozwój chatbotów zainspirowany został w 1950 roku przez Alana

Turinga, który chciał udowodnić inteligentne zdolności maszyny, potrafiące udawać człowieka w rozmowie za pośrednictwem języka tekstowego. Do dziś corocznie organizowane są konkursy, w których programy zostają poddane tzw. testowi Turinga. Aktualnie za najlepsze chatboty uważane są projekty Alice oraz Jabberwacky, nad którymi stale trwają prace rozwojowe. Coraz bardziej skuteczne stają się również programy rozpoznające mowę oraz gesty, co umożliwia komunikację z komputerem nie tylko za pomocą klawiatury ale również poprzez naturalny język. Takie systemy są niezwykle ważne dla osób niepełnosprawnych, gdyż dzięki nim są w stanie komunikować się z wieloma maszynami.

Równie istotnym zagadnieniem jak możliwość łatwego i przyjaznego dla użytkownika sposobu wydawania poleceń, jest sposób odreagowywania i prezentacji wyników działania urządzenia. Oprócz standardowych informacji wyświetlanych za monitorze komputera coraz częściej tekst przekazywany jest za pomocą głosu. Systemy wyposażane są w specjalne syntezaatory mowy umiejące wypowiadać dowolny tekst z odpowiednią intonacją. Jest to kolejny bardzo duży krok na drodze do stworzenia idealnej symulacji człowieka.

Podczas rozmowy dwóch osób bardzo ważnym elementem przekazu informacji jest język niewerbalny. Kluczowym sposobem nawiązywania dialogu jest kontakt wzrokowy. Poprzez uśmiech można bardzo łatwo okazać sympatię. Pochylenie głowy na bok jest oznaką zaciekawienia. Te i wiele innych prostych gestów umożliwiają ludziom na szybkie nawiązywanie kontaktu, a tym samym na swobodniejsze prowadzenie rozmowy. W sieci Internet istnieje obecnie kilka ciekawych rozwiązań prezentacji reakcji programu na uzyskiwane od użytkownika polecenia i odpowiedzi. Są to tzw. awatary – graficzna prezentacja sztucznej osobowości. Do najpopularniejszych należą trójwymiarowe postacie imitujące ludzkie ruchy. Na wielu stronach (głównie sklepach internetowych) można napotkać na krótkie filmy z osobą wyrażającą pewne gesty i reakcje. Wybrane sceny odtwarzane są w zależności od akcji użytkownika co daje wrażenie interakcji z żywą postacią.

Celem danej pracy jest stworzenie systemu śledzenia twarzy za pomocą kamery internetowej oraz reagowanie na ruch użytkownika komputera. Wykryte akcje

osoby siedzącej przed monitorem komputera oraz jej pozycja odwzorowane będą za pomocą odpowiednich reakcji i położenia oczu trójwymiarowej postaci tzw. cyberagenta. Stworzona postać ma za zadanie obserwować osobę przed komputerem oraz reagować na jej proste gesty mimiki.

Pierwszym kluczowym celem przy tworzeniu danego systemu jest opracowanie odpowiednich algorytmów, m.in. lokalizacji twarzy oraz wykrywania mimiki. Biometria czyli nauka zajmująca się m.in. pomiarami istot żywych, jest bardzo obszerną dziedziną. Istnieje bardzo wiele różnorodnych metod rozpoznawania i ustalania cech twarzy. Najistotniejszą cechą danych algorytmów w realizowanej pracy jest ich wydajność. Reakcja oczu na zmianę położenia twarzy jak również odpowiednie odreagowywanie na akcję rozmówcy jest zwykle natychmiastowa. Aby uzyskać podobne wrażenie system powinien działać w czasie rzeczywistym. Opracowane algorytmy muszą być zatem szybkie i wydajne. Kolejnym zagadnieniem, które należy zrealizować jest stworzenie odwzorowującego ludzkie odruchy cyberagenta.

W rozdziale drugim opisano przeanalizowane algorytmy lokalizacji twarzy, rozpoznawania emocji oraz rozwiązania z zakresu animacji postaci. Rozdział trzeci opisuje opracowane algorytmy wykorzystane w pracy. Rozdział czwarty zawiera prezentację testów działania opracowanych algorytmów.

2. Przegląd istniejących rozwiązań

2.1. Metody lokalizacji twarzy

Kluczowym zagadnieniem związanym z realizacją danej pracy jest lokalizacja twarzy na obrazie z kamery internetowej. Po wykryciu pozycji twarzy możliwa jest dalsza ekstrakcja jej elementów oraz wykrywanie stanu mimiki.

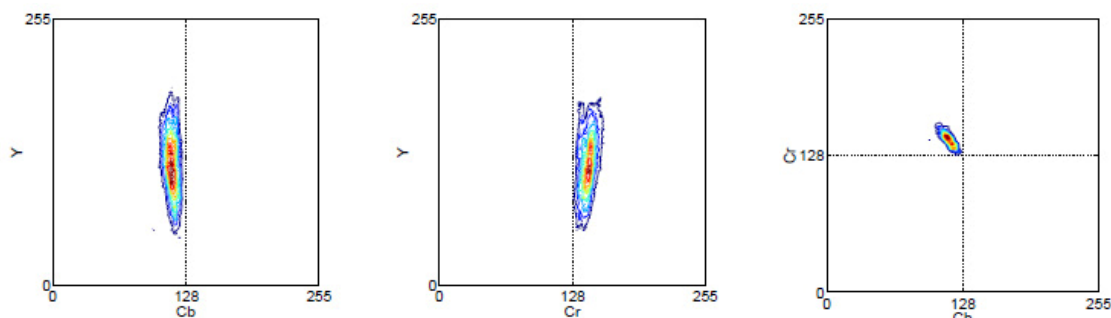
Obecnie działające systemy wykorzystujące lokalizację twarzy działają na bardzo różnorodnych algorytmach. Każdy z nich charakteryzuje się mniejszą lub większą szybkością działania i efektywnością poprawnej lokalizacji twarzy.

2.1.1. Metody oparte na porównywaniu kolorów

Do podstawowych algorytmów lokalizacji twarzy zalicza się całą gamę metod bazujących na kolorach. Ludzka skóra posiada charakterystyczne zabarwienie. Wyznaczając odpowiedni wzorec barwy można wskazać na obrazie regiony ciała człowieka. W pracy [1] wymienionych zostało kilka metod bazujących na różnych przekształceniach przestrzeni barw. Do najpopularniejszych przekształceń wykorzystywanych w algorytmach lokalizacji twarzy należą transformacje nieliniowe: HSV, I1I2I3 oraz transformacja liniowa YCbCr. Algorytm bazujący na przestrzeni barw YCbCr wykorzystano w pracy [2].

Model YCbCr opisywany jest przez składowe Y – luminancja, Cb – chrominancja różnicowa i Cr – chrominancja.

Analizując pewną ilość obrazów zawierających twarz ustalono, że średnia wartość składowych Cb oraz Cr dla ludzkiej skóry zamknięta jest w ściśle określonych granicach.



Rysunek 2.1 a, b, c: Wartości koloru skóry w przestrzeni barw YCbCr

Wykorzystując dany fakt możliwe jest ustalenie regionów obrazu, w których zlokalizowana jest skóra. Jeżeli istnieje obszar z dużą częścią punktów oznaczonych jako punkty skóry można przyjąć, że została odnaleziona twarz. Daną metodę trzeba uznać za mało skuteczną, jako, że istnieje duże prawdopodobieństwo wykrycia innych przedmiotów, których barwy odpowiadają przyjętym wartościom dla skóry. Jednocześnie jako twarz może zostać oznaczona również każda inna część ciała.

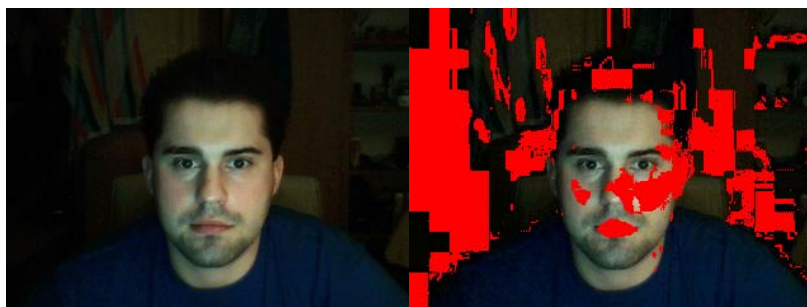
Implementacja algorytmu opisanego w pracy [2] okazała się skuteczna pod względem oznaczania punktów skóry na obrazach dobrej jakości, wykonanych w środowisku o równo rozłożonym oświetleniu. W przypadku obrazów z kamery internetowej, wykonanych w pomieszczeniach słabo lub nierównomiernie oświetlonych, jedynie mała część punktów zawierających skórę została oznaczona. Błędnie wykryta została również duża część punktów, niezawierających skóry. Przykłady poprawnego działania algorytmu opartego o porównywanie barw w przestrzeni YCbCr prezentują rysunki 2.2 a i b oraz 2.3 a i b. Rysunek 2.4 a i b jest obrazem uzyskanym z kamery internetowej działającej w pomieszczeniu o bardzo słabym oświetleniu. Widać, że większa część punktów twarzy nie została oznaczona, a jednocześnie duża ilość pikseli została oznaczona błędnie.



Rysunek 2.2 a i b: Zdjęcie dobrej jakości. Kolorem czerwony oznaczono punkty wykryte jako skóra.



Rysunek 2.3 a i b: Zdjęcie dobrej jakości. Kolorem czerwony oznaczono punkty wykryte jako skóra.



Rysunek 2.4 a i b: Zdjęcie niskiej jakości z kamery internetowej. Kolorem czerwony oznaczono punkty wykryte jako skóra.

2.1.2. Metody oparte na dopasowywaniu schematów

Kolejną popularną grupą algorytmów lokalizacji twarzy są metody oparte na porównywaniu wybranych regionów obrazu do pewnego wzorca (tzw. *template matching*). W pracach [1] i [3] opisano algorytmy bazujące na szablonie utworzonym jako średnia wypadkowa wartość obrazu w skali szarości dla różnych twarzy o zbliżonym rozmiarze. Uzyskany wzorec prezentuje Rysunek 2.5.



Rysunek 2.5: Przykład stosowanego schematu w metodach *template matching*

Dana metoda polega na „skanowaniu” obrazu od lewego górnego rogu do przeciwległego krańca w poszukiwaniu części obrazu odpowiadającej wykorzystywanemu schematowi. Dla każdego badanego obszaru wyliczana jest różnicowa funkcja błędu. Wiele odmian tej metody korzysta z różnych wersji wzoru danej funkcji. W pracy [1] funkcja opisana jest wzorem:

$$\Delta(x,y) = \sqrt{\frac{1}{IJ} \sum_{i=1}^I \sum_{j=1}^J (O(x+i-1,y+j-1) - W(i,j))^2} \quad (2.1)$$

gdzie:

$$x = 1,2,3, \dots W-I+1$$

$$y = 1,2,3, \dots H-J+1$$

W, H – wysokość i szerokość obrazu wejściowego

$\Delta(x,y)$ – wartość funkcji błędu w punkcie o współrzędnych (x,y)

$O(i,j)$ – wartość piksela (i,j) obrazu wejściowego

$W(i,j)$ – wartość piksela (i,j) obrazu wzorca

I, J – wysokość i szerokość wzorca

Wyznaczenie wartości danej funkcji wymaga przeliczenia różnicy wszystkich odpowiadających sobie pikseli w badanym rejonie z obrazu wejściowego oraz na schemacie. Przy braku informacji na temat rozmiaru poszukiwanej twarzy, obraz należy przeszukać kilkakrotnie porównując jego części do schematów twarzy różnych wielkości.

Operacje wykonywane na pojedynczych pikselach są bardzo czasochłonne, dlatego też pomimo zadowalającej skuteczności, algorytmy danego typu nie nadają się do zastosowania w systemie czasu rzeczywistego.

W celu zwiększenia wydajności algorytmu w pracy [6] wykorzystano fakt symetrii twarzy. Na obrazie wyszukiwany jest schemat zawierający połowę twarzy. Dwukrotnie mniejsza ilość pikseli wymagająca sprawdzenia znacznie poprawia wydajność systemów typu *template matching*, lecz działanie danego algorytmu nadal wymaga wykonywania bardzo wielu obliczeń.

2.1.3. Metody oparte na cechach

Jedną z popularniejszych metod, bardzo często wykorzystywaną w komercyjnych systemach jest opublikowana w 2001 roku przez Poula Viole oraz Michaela Jones'a [5] metoda oparta na cechach. Charakteryzuje się bardzo dużą wydajnością i skutecznością wykrywania twarzy. Wykorzystano metodę [6], wg której dla każdego obiektu charakteryzującego się specyficznym kształtem można wskazać cały zespół cech na obrazie w skali szarości do utworzenia wzorca danego obiektu.

Przykładem cechy specyficznej dla obrazów zawierających twarz jest fakt, że część pomiędzy oczami na obrazie w skali szarości jest jaśniejsza od części zawierających oczy (rysunek 2.6b). Podobnym przykładem jest cecha zaprezentowana na rysunku 2.6c - obszar pod oczami jest znacznie jaśniejszy od obszaru oczu.



Rysunek 2.6 a, b, c: Obraz w skali szarości oraz przykłady cech twarzy.

Poszukując twarz, podobnie jak algorytmy bazujące na schematach, dana metoda sprawdza części obrazu o pewnym zakresie wymiarów. Dla każdej z takich części (klatek) sprawdza się czy zawiera cechy o ściśle określonym położeniu i rozmiarze, które ustalone zostają przez algorytm uczący się - *Ada Boost* [7]. Bazując na serii zdjęć algorytm ustalił zestaw cech o kształtach przedstawionych na rysunku 2.7, w różnym rozmiarze, które zostały odnalezione

na obrazach z twarzą i nie występują na obrazach bez twarzy. W dany sposób tworzony jest pewnego rodzaju schemat porównania.



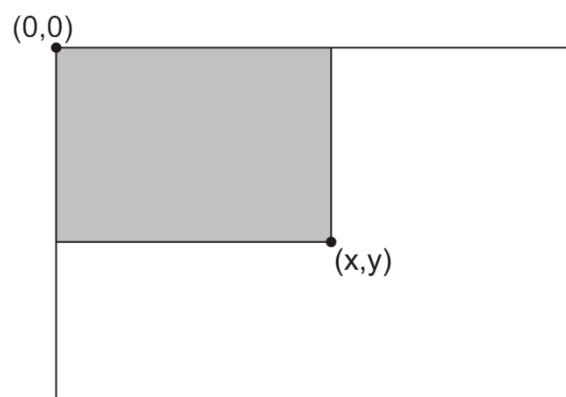
Rysunek 2.7: Zastosowane cechy w algorytmie Viola-Jones'a

Algorytm posiada formę kaskadową tzn. w przypadku stwierdzenia braku jednej z cech wymaganych dla obrazu twarzy badana klatka zostaje uznana jako niezawierająca twarzy. Dana metoda zapewnia bardzo dużą wydajność.

Dodatkowo w algorytmie wykorzystany został tak zwany obraz zintegrowany (ang. *integral image*), co pozwala na bardzo szybkie wyliczanie sum wartości dowolnych obszarów obrazu.

2.1.3.1. Integral image

Integral image jest formą obrazu, która w punkcie współrzędnych (x,y) przechowuje wartość równą sumie wartości pikseli w obszarze, którego lewy górny róg wyznaczany jest przez punkt o współrzędnych $(0,0)$ ¹, a prawym dolnym rogiem jest punkt (x,y) .



Rysunek 2.8: Przykład oznaczeń punktów na obrazie

Jeśli wartość obrazu w skali szarości w punkcie o współrzędnych (i,j) określana jest jako $gray(i,j)$ to wartość integral image w punkcie (x,y) równa jest:

¹Punkt początkowy obrazu może mieć współrzędne $(0,0)$ lub $(1,1)$ w zależności od wykorzystywanego systemu. Przykładowo w systemie Matlab tablice indeksowane są począwszy od 1, a w systemie Java lub C++ od 0.

$$\text{integral_image}(x, y) = \sum_{i=0}^x \sum_{j=0}^y \text{gray}(i, j) \quad (2.1)$$

Obraz integral image można szybko wyliczyć w jednym przejściu korzystając ze wzorów:

$$s(x, y) = s(x-1, y) + \text{gray}(x, y) \quad (2.2)$$

$$\text{integral_image}(x, y) = \text{integral_image}(x, y-1) + s(x, y) \quad (2.3)$$

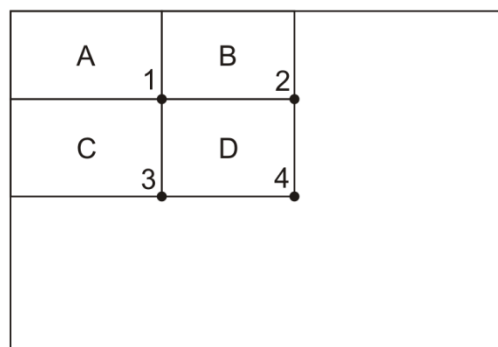
przyjmując

$$s(-1, y) = 0$$

oraz

$$\text{integral_image}(x, -1) = 0$$

Używając integral image, suma każdego obszaru może być obliczona za pomocą maksymalnie trzech operacji.



Rysunek 2.9: Sposób oznaczeń obszarów na *integral image*

Jeśli A, B, C i D określimy jako sumy wartości pikseli w oznaczonych przez nie obszarach to:

- wartość integral image w punkcie 1 równa jest wartości obszaru A
- w punkcie 2, sumie A+B,
- w punkcie 3 sumie A+C
- w punkcie 4 sumie A+B+C+D

Wartość obszaru D może zostać obliczona ze wzoru:

$$D = \text{integral_image}(4) - \text{integral_image}(3) - \text{integral_image}(2) + \text{integral_image}(1) \quad (2.4)$$

Metoda lokalizacji obiektów opartej o cechy została rozwinięta przez Rainera Lienharta. Opisany w pracy [8] algorytm oprócz wymienionych cech wykorzystuje dodatkowe, przedstawione na rysunku 2.10.



Rysunek 2.10: Cechy zastosowane w algorytmie Lienharta

Wprowadzenie dodatkowych cech znacznie zwiększa wydajność danego algorytmu. Podobnie jak w algorytmie Viola-Jones, wartości dla cech algorytmu Lienharta obliczane są korzystając z obrazu *integral image* w odmienionej formie (opisany w [8]).

Pomimo konieczności walidacji obrazu jako zawierającego twarz przez kilkaset cech jest to zdecydowanie jedna z bardziej skutecznych, a zarazem wydajnych metod lokalizacji twarzy. Przyczyniło się to do zastosowania tej metody w bardzo popularnym projekcie typu open source o nazwie Open Computer Vision Library (OpenCV) [9] stworzonej przez firmę Intel. Jest to biblioteka napisana w języku C++ zawierająca cały szereg algorytmów związanych z przetwarzaniem obrazów. Stale rozwijana oraz wzbogacana o nowe możliwości jest często stosowana w komercyjnych systemach wykorzystujących metody biometryczne.

2.2. Wykrywanie mimiki

Rozpoznawanie mimiki jest bardzo bogatym w metody działem biometrii. Każdą z cech mimicznych można opisać zależnościami pomiędzy położeniem brwi, oczu i ust oraz ich kształtem. W celu ekstrakcji części twarzy wykorzystywany jest cały szereg metod – najczęściej wykorzystywane są algorytmy bazujące na:

- schematach
- cechach
- ruchu
- maszynach wektorów nośnych SVN (ang. *support vector machines*)

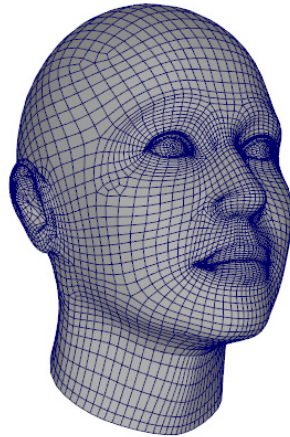
W pracy [10] do lokalizacji oczu oraz ust wykorzystano informacje o kolorach. Obszary oczu na obrazie w skali szarości, jako rejony najciemniejsze posiadają małą wartość. Usta posiadają odmienne od skóry zabarwienie, dzięki czemu stosując metody oznaczania regionów wg koloru można określić ich dokładne położenie.

W pracy [14] do lokalizacji oczu wykorzystano algorytm bazujący na fakcie okresowego mrugania oczami. Regiony twarzy w okolicach odpowiadających położeniu oczu, w których wykryty zostaje ruch uznawane są za oczy. Dana metoda jest skuteczna lecz ze względu na zwykle bardzo szybko wykonywane mrugnięcia istnieje konieczność szybkiej rejestracji zmian na obrazie.

2.3. Animacja postaci

Pierwsze animacje sięgają początku XIX wieku, kiedy to wynaleziono tzw. zdjęcia poklatkowe. Jako, że ludzkie oko rejestruje ok. 25 klatek na sekundę, seria szybko zmienianych obrazów zawierających ujęcia o małym przesunięciu obiektów oddaje wrażenie ich ciągłego ruchu.

Obecnie animacja postaci jest bardzo rozwiniętym działem z dziedziny przetwarzania i generowania grafiki komputerowej. Zwykle wykorzystywane w grach komputerowych animacje ludzi, w tym również mimiki twarzy stają się coraz bardziej realistyczne. Szybko rozwijające się technologie kart graficznych dają bardzo bogate możliwości. Większość animacji trójwymiarowych bazuje na specjalnych modelach obiektów 3D. Przykładowy model 3D twarzy składa się nawet z kilkuset charakterystycznych punktów. Przemieszczanie wybranych partii punktów umożliwia animację mimiki twarzy.



Rysunek 2.11: Trójwymiarowy model twarzy

Na rynku istnieje kilka programów umożliwiających w prosty sposób stworzenie realistycznie wyglądającej postaci.

W chwili obecnej najlepszym ze względu na bogate możliwości programem do kreowania modeli postaci oraz tworzenia zaawansowanych animacji jest Poser. Dany program umożliwia zaprojektowanie drobnych szczegółów wyglądu oraz stworzenie dowolnej animacji. Predefiniowane funkcje pozwalają na ustawienie nawet takich szczegółów jak sposób poruszania się i wykonywania gestów.

Bardzo bogatym w funkcje jest program do modelowania twarzy FaceGen Modeller. Aplikacja ta umożliwia ustalenie niemal każdego szczegółu wyglądu – kształtu, barwy jak i wyrazu twarzy.

3. Opis rozwiązania

Ze względu na specyficzny charakter systemu jego realizację zdecydowano się wykonać jako aplikację internetową, współpracującą ze standardowymi kamerami internetowymi oraz niezależną sprzętowo. Jako środowisko programistyczne wybrano język Java, który spełnia wymienione wymogi. Dodatkowo aplet Java, którego kod wykonywany jest w całości po stronie użytkownika strony internetowej umożliwia szybkie, ciągłe przetwarzanie obrazu pobieranego z kamery internetowej.

Najbardziej rozbudowanym aspektem pracy są algorytmy akwizycji i przetwarzania obrazów z kamery internetowej. Głównym problemem związanym z wykorzystaniem kamer internetowych jest otrzymywana niska jakość obrazu. Obecnie większość użytkowników komputera korzysta z kamer wbudowanych w komputer przenośny typu laptop lub prostych kamer internetowych, których rozdzielczość nie przekracza 1.3 Mpx. Jest to dość przyzwoita rozdzielczość lecz obraz często bywa zaszumiony i niewyraźny. Należy również zwrócić uwagę na fakt, iż często komputer umieszczony jest w pomieszczeniu o nierównym oświetleniu. Tym samym twarz użytkownika będzie mocniej oświetlona z jednej strony.

W celu uzyskania szybkiej lokalizacji oraz odpowiedzi cyberbota na zarejestrowane zmiany na obrazie, algorytm musi być wykonywany w bardzo krótkim czasie. Obecnie standardem użytkowanych komputerów są jednostki wyposażone w dwa, a nawet cztery rdzenie procesora. Jednak nawet tak wydajne maszyny nie będą w stanie wykonać zaawansowanych algorytmów na obrazach o dużej rozdzielczości w krótkim czasie. W celu uzyskania szybkiego algorytmu lokalizacja będzie realizowana na obrazach o rozdzielczości 320x240 pikseli. Jest to wielkość oferowana przez większość znajdujących się na rynku urządzeń przechwytyjących obraz.

3.1. Lokalizacja twarzy

Pierwszym etapem przetwarzania obrazu jest wykrycie twarzy. Po jej zlokalizowaniu możliwe jest ustalenie położenia poszczególnych partii twarzy oraz dalsze przetwarzanie mające na celu określenie cech mimiki.

W danej pracy podjęto próbę utworzenia algorytmu bazującego na cechach – analogicznie do metody opisanej w [5] i [8]. Cechy w przeciwieństwie to danych algorytmów zostaną wybrane ręcznie na podstawie przeprowadzonych doświadczeń.

W opracowanej metodzie wykorzystano obraz w skali szarości oraz obraz binarny uzyskany autorskim algorytmem.

3.1.1. Obraz w skali szarości

Wartość skali szarości dla punktu obrazu o współrzędnych (i,j) wyliczany jest ze wzoru:

$$\text{gray}(x,y)=0.3*\text{red}(x,y)+0.59*\text{green}(x,y)+0.11*\text{blue}(x,y) \quad (3.1)$$

gdzie:

gray(x,y) – wartość obrazu w skali szarości

red(x,y) – wartość koloru czerwonego obrazu w formacie RGB

green(x,y) - wartość koloru zielonego obrazu w formacie RGB

blue(x,y) - wartość koloru niebieskiego obrazu w formacie RGB

3.1.2. Obraz binarny

W obrazie binarnym piksele, których wartość na obrazie w skali szarości jest niższa od pewnego progu τ przyjmują wartość 1, a pozostałe 0.

W przypadku równomiernego oświetlenia na twarzy można wyłonić wiele cech, m.in. zaprezentowane na rysunkach 2.6b oraz 2.6c.

W przypadku nierównomiernie oświetlonego miejsca w jakim znajduje się komputer i kamera internetowa, dane cechy mogą nie zostać odnalezione.

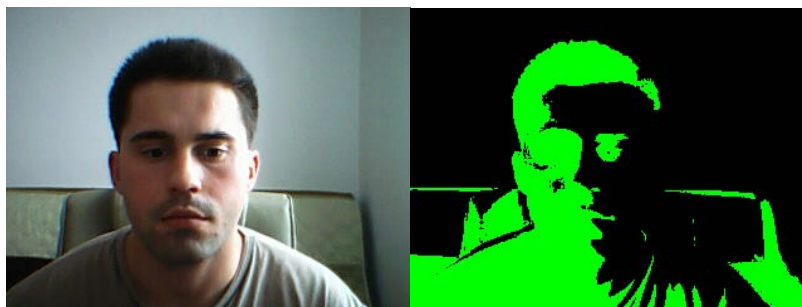
Przeprowadzenie standardowej binaryzacji obrazu z odpowiednio dobranym progiem w zależności od oświetlenia „wyłania” główne cechy twarzy.



Rysunek 3.1 a i b: Obraz dobrej jakości w formacie RGB oraz po binaryzacji ze stałym progiem

Na obrazie widocznie zarysowane są oczy, nos usta oraz sam kształt głowy.

Dla zdjęcia wykonanego w pomieszczeniu o nierównomiernym oświetleniu dane cechy twarzy nie zostaną uwidocznione.



Rysunek 3.2 a i b: Obraz słabej jakości w formacie RGB oraz po binaryzacji ze stałym progiem

Zauważono, że pomimo nierównomiernego oświetlenia, pewne części twarzy np. obszar oka w swoim lokalnym otoczeniu na obrazie w skali szarości posiada wartość najniższą (jest najciemniejszy).

Po przeprowadzeniu testów na obrazie o nierównomiernym oświetleniu ustalono, że dokonując binaryzacji wyliczając osobny próg dla każdego punktu można wyłonić cechy pomijane przy zwykłej binaryzacji.

Nowy obraz binarny wyliczany jest wg wzoru:

$$\text{binary}(x,y) = \begin{cases} 1 & \text{gdy } \text{gray}(x,y) \leq \text{threshold}(x,y) \\ 0 & \text{w innym przypadku} \end{cases} \quad (3.2)$$

gdzie:

$binary(x,y)$ – wartość obrazu binarnego w punkcie o współrzędnych (x,y)

$gray(x,y)$ – wartość obrazu w skali szarości w punkcie o współrzędnych (x,y)

Próg binaryzacji wyliczany jest dla każdego pojedynczego punktu. Jego wartość zależy od wartości sumy pikseli obrazu w skali szarości na obszarze 7×7 o środku w punkcie (x,y) , dla którego obliczany jest próg.

$$threshold(x,y) = w_bin * average_val(x,y) \quad (3.3)$$

oraz:

$$average_value(x,y) = \frac{\sum_{x=x1}^{x2} \sum_{y=y1}^{y2} gray(x,y)}{w * h} \quad (3.4)$$

$$x1=x-3, x2=x+3, y1=y-3, y2=y+3, w_bin=0.95, w=7, h=7$$

gdzie:

x, y – współrzędne punktu, dla którego wyliczany jest próg $threshold$

$(x1,y1)$ i $(x2,y2)$ – współrzędne odpowiednio lewego-górnego oraz prawego-dolnego rogu obszaru, dla którego wyliczana jest suma wartości pikseli

w_bin – empirycznie dobrany współczynnik

w, h – szerokość i wysokość wyliczanego obszaru

$average_value(x,y)$ – średnia wartość obrazu w skali szarości dla obszaru o rozmiarze 7×7 , o środku w punkcie (x,y)

Obraz 3.2a poddano binaryzacji opisanym powyżej algorytmem. Wynik można zaobserwować na rysunku 3.3.



Rysunek 3.3: Obraz słabej jakości po binaryzacji ze zmiennym progamiem

Pomimo nierównomiernego oświetlenia główne elementy twarzy zostały wyeksponowane, dzięki czemu można wskazać wiele cech, które mogą posłużyć lokalizacji twarzy.

3.1.3. Schemat algorytmu lokalizacji twarzy

Algorytm lokalizacji twarzy składa się z 25 etapów - klasyfikatorów. Na każdym z nich sprawdzane jest czy obszar (klatka) będący „kandydatem na twarz” zawiera określone cechy.

Cechy na poszczególnych etapach posiadają ściśle określone położenie lub obszar, w którym dana cecha musi wystąpić. Po wyliczeniu cechy, jej wartość porównywana jest z pewnym progiem. Jeśli spełnia ona nierówność określoną dla tego etapu, klatka zostaje „przepuszczona” przez klasyfikator.

Cechy, wartości progu dla klasyfikatorów oraz kolejność sprawdzanych cech wykorzystanych w algorytmie lokalizacji twarzy zostały dobrane empirycznie. Eksperymenty i doświadczenia przeprowadzone na wielu obrazach zawierających twarze oraz obrazach bez twarzy pozwoliły na dobranie optymalnych parametrów.

W celu lokalizacji twarzy poszukuje się klatki obrazu (obszaru w kształcie kwadratu) o rozmiarze z obranego zakresu, badając obraz wejściowy kolejno od lewej do prawej strony i od góry do dołu z krokiem 2 pikseli.

Dla przyśpieszenia obliczeń obraz z kamery pobierany jest w rozdzielczości 320x240 pikseli. Rozmiar klatki, będącej „kandydatem na twarz” zwiększana jest od wielkości 50x50 co 5 pikseli do wielkości 160x160. Dane wielkości zostały dobrane empirycznie mając na uwadze, że osoba znajduje się przed monitorem, w pobliżu którego usytuowana jest kamera internetowa.

Etapy złożone z cech występujących we wszystkich obrazach zawierających twarz zostały zakwalifikowane jako mocne klasyfikatory. W celu poprawienia szybkości działania algorytmu, tak jak w pracy [5] i [8] kolejne etapy zostały połączone w formę drzewa. Każda klatka sprawdzana jest przez kolejne klasyfikatory. Jeśli na jednym z etapów klatka nie posiada lub wyliczona wartość wybranej cechy nie

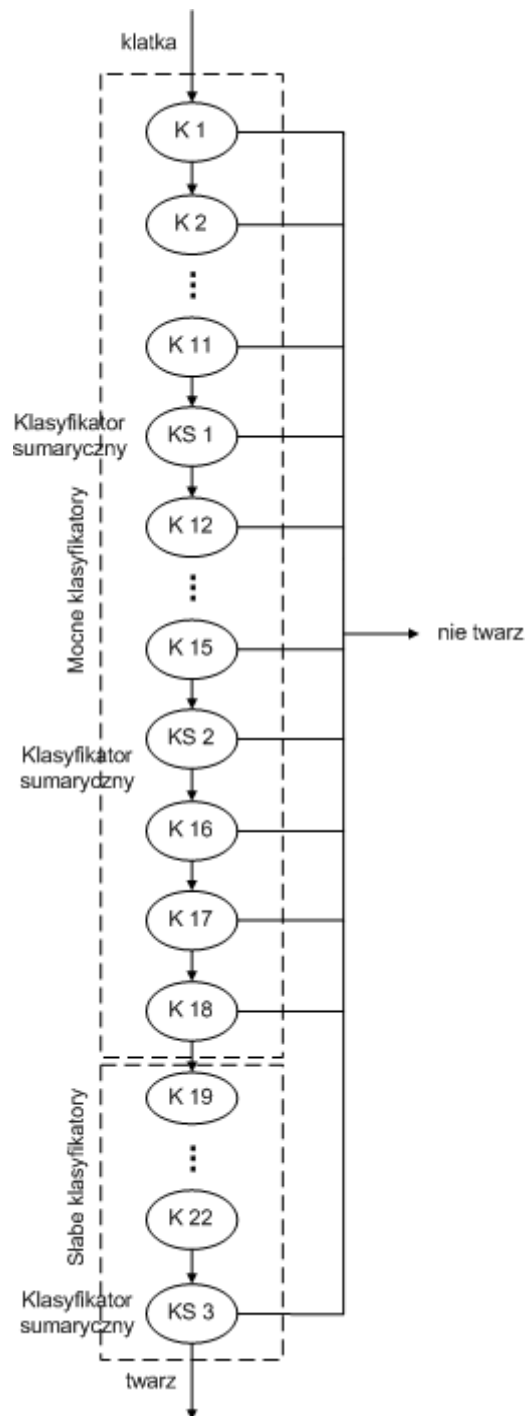
spełnia określonego warunku, zostaje ona odrzucona i nie jest sprawdzana na kolejnych etapach.

W algorytmie zastosowano również słabe klasyfikatory. Składają się one z cech, które nie występują na wszystkich obrazach z twarzami. Pojedynczo nie wyznaczają przynależności klatek do twarzy lub nie-twarzy. Wyliczona wartość cech danych klasyfikatorów wykorzystana jest na końcu algorytmu.

Dla poprawienia szybkości działania klasyfikatory zostały uporządkowane według złożoności obliczeniowej oraz otrzymanych w doświadczeniach informacji o ilości odrzucanych klatek. Jako pierwsze zostały użyte mocne klasyfikatory, o niskiej złożoności obliczeniowej, odrzucające dużą ilość klatek nie-twarzy. Na końcu algorytmu wyliczane są słabe klasyfikatory.

Po przeprowadzeniu szeregu doświadczeń ustalono, że można dołączyć dodatkowe klasyfikatory pomiędzy etapami oraz na końcu algorytmu. Obliczane są one jako sumy iloczynów wartości wybranych, wcześniej wyliczonych cech oraz odpowiednich współczynników. Progi dla danych klasyfikatorów oraz współczynniki dla wartości poszczególnych cech dobrane zostały empirycznie.

Wartość wyliczona przez ostatni klasyfikator sprawdzający sumę cech wcześniej obliczonych etapów przemnożonych przez odpowiednie współczynniki zostaje zapisana jako „wartość jakości twarzy”. Jeżeli algorytm dla jednej twarzy znajduje wiele klatek różniących się położeniem lub wielkością zostaje wybrana klatka posiadająca największą wartość jakości twarzy.



Rysunek 3.4: Schemat połączenia klasyfikatorów

3.1.4. Szczegółowy opis klasyfikatorów

Położenie cech jest ściśle określone względem położenia klatki i jej rozmiarów. Poniższe parametry wykorzystane są do opisu położenia poszczególnych cech.

frame_x – współrzędna x położenia lewego górnego rogu klatki

frame_y – współrzędna y położenia lewego górnego rogu klatki

frame_size – rozmiar klatki (wartość szerokości i wysokości)

Dla obrazu w skali szarości oraz obrazu binarnego wyliczony został obraz zintegrowany (*integral image*) opisany w 2.1.3.1.

integral_gray – zintegrowany obraz w skali szarości

integral_binary – zintegrowany obraz po lokalnej binaryzacji

3.1.4.1. Analiza wartości klatki (obraz binarny)

Rysunek 3.5a przedstawia zdjęcie zawierające twarz. Rysunek 3.5b prezentuje obraz po lokalnej binaryzacji wraz z zaznaczoną wykrytą twarzą. Na rysunku wyraźnie zaznaczają się rysy twarzy. Po przeprowadzeniu eksperymentów ustalono, że średnia wartość ilości pikseli o wartości 1 na obrazie binarnym dla klatki zawierającej twarz mieści się w przedziale [0.07, 0.4]. Klasyfikator sprawdza czy dla klatki spełnione są nierówności 3.5 i 3.6.

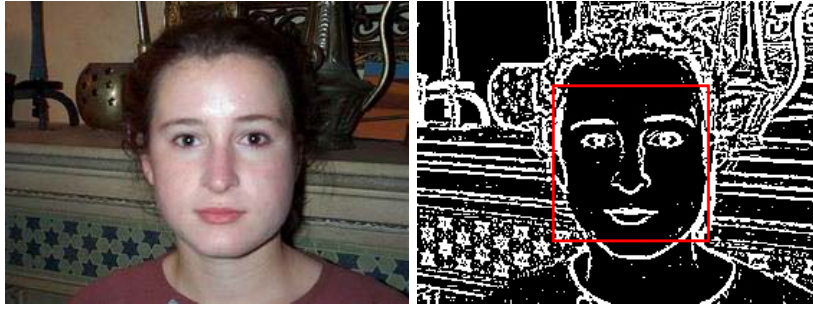
$$\text{value1} \geq 0,07 \quad (3.5)$$

$$\text{value1} \leq 0,4 \quad (3.6)$$

gdzie:

$$\text{value1} = \frac{\text{frame_val}}{\text{frame_size} * \text{frame_size}} \quad (3.7)$$

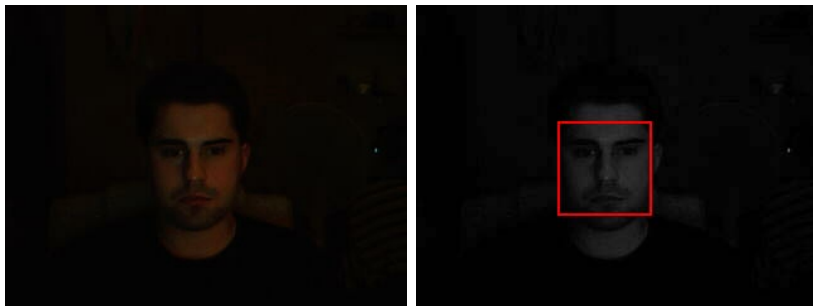
frame_val – wartość obszaru ograniczonego przez klatkę, na obrazie integral_binary



Rysunek 3.5 a i b: Zdjęcie w formacie RGB oraz po lokalnej binaryzacji.

3.1.4.2. Analiza wartości klatki (obraz w skali szarości)

Kolejny klasyfikator wylicza średnią wartość skali szarości dla klatki. Dana wartość musi być większa od zadanego progu równego 20. Ten etap pozwala odrzucić klatki o zbyt niskiej łącznej wartości pikseli obrazu w skali szarości.



Rysunek 3.6 a i b: Przykład zdjęcia twarzy w pomieszczeniu o niskim oświetleniu.

Nierówność sprawdzana przez klasyfikator:

$$\text{value2} > 20 \quad (3.8)$$

gdzie:

$$\text{value2} = \frac{\text{frame_val}}{\text{frame_size} * \text{frame_size}} \quad (3.9)$$

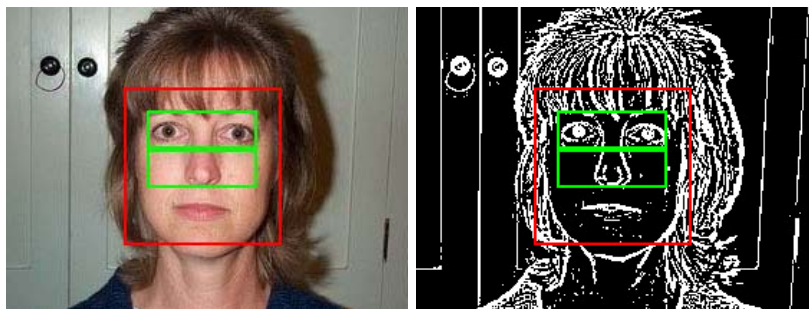
frame_val – wartość obszaru ograniczonego przez klatkę, na obrazie integral_gray

Doświadczenia pokazały, że w bardzo niskim oświetleniu średnia wartość skali szarości dla twarzy jest większa od wyznaczonego progu. Na rysunku 3.6 a i b zaprezentowano zdjęcie wykonane w pomieszczeniu o niskim oświetleniu. Na danym etapie wszystkie klatki nie obejmujące powierzchnią widocznej twarzy

zostały odrzucone, dzięki czemu twarz została odnaleziona w bardzo krótkim czasie.

3.1.4.3. Analiza dużego obszaru oczu oraz pod oczami (obraz binarny)

W danym etapie algorytmu cecha obliczana jest jako stosunek wartości powierzchni dolnej do powierzchni górnej (zaznaczonej kolorem zielonym na rysunku 3.7 a i b) na obrazie binarnym.



Rysunek 3.7 a i b: Zdjęcie w formacie RGB oraz po lokalnej binaryzacji.

Współrzędne lewego górnego rogu cechy (x,y) wyznaczone są wzorem:

$$x = \text{frame_x} + 0,15 * \text{frame_size} \quad (3.10)$$

$$y = \text{frame_y} + 0,15 * \text{frame_size} \quad (3.11)$$

Oba prostokąty posiadają jednakowe wymiary. Szerokość oznaczana jako w oraz wysokość h wyliczane są ze wzorów:

$$w = 0,7 * \text{frame_size} \quad (3.12)$$

$$h = 0,25 * \text{frame_size} \quad (3.13)$$

Klasyfikator sprawdza czy został spełniony warunek:

$$\text{value3} < 0,8 \quad (3.14)$$

gdzie:

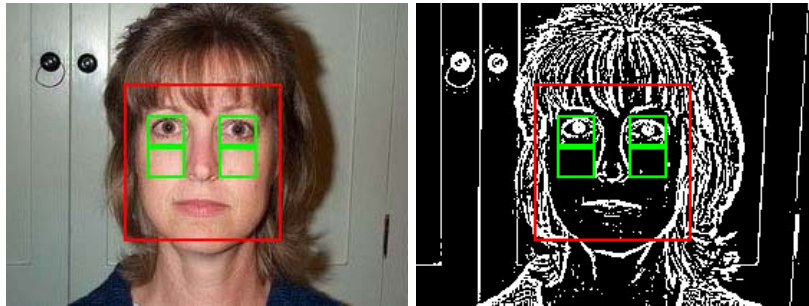
$$\text{value3} = \frac{\text{top_rect}}{\text{bottom_rect}} \quad (3.15)$$

top_rect – wartość górnego obszaru na obrazie integral_binary

bottom_rect – wartość dolnego obszaru na obrazie integral_binary

3.1.4.4. Analiza małych obszarów oczu oraz pod oczami (obraz binarny)

Na danym etapie wyliczane są dwie cechy określone jako stosunek wartości powierzchni dolnej do górnej dla obrazu binarnego, przedstawionych na rysunku 3.8.



Rysunek 3.8 a i b: Zdjęcie w formacie RGB oraz po lokalnej binaryzacji z zaznaczoną wykrytą klatką zawierającą twarz.

Wymiary powierzchni wyznaczane są wzorami:

$$w=0,25*frame_size \quad (3.16)$$

$$h=0,2*frame_size \quad (3.17)$$

$$y=frame_y+0,2*frame_size \quad (3.18)$$

Współrzędne x położenia lewej górnej krawędzi cechy określają wzory:

- lewy obszar

$$x_left=frame_x+0,15*frame_size \quad (3.19)$$

- prawy obszar

$$x_right=frame_x+0,6*frame_size \quad (3.20)$$

Klasyfikator sprawdza czy są spełnione warunki:

$$\begin{cases} value4 \leq 0,65 \\ value4_1 < 0,8 \\ value4_2 < 0,8 \end{cases} \quad (3.21)$$

gdzie:

$$\text{value4} = \begin{cases} \frac{\text{left_bottom_rect} + \text{right_bottom_rect}}{\text{left_top_rect} + \text{right_top_rect}} & \text{jeśli } \frac{\text{left_top_rect}}{w * h} > 0,15 \cap \frac{\text{right_top_rect}}{w * h} > 0,15 \\ 1 & \text{jeśli inaczej} \end{cases} \quad (3.22)$$

$$\text{value4_1} = \frac{\text{left_bottom_rect}}{\text{left_top_rect}} \quad (3.23)$$

$$\text{value4_2} = \frac{\text{right_bottom_rect}}{\text{right_top_rect}} \quad (3.24)$$

left_bottom_rect - wartość lewego dolnego obszaru na obrazie integral_binary

left_top_rect - wartość lewego górnego obszaru na obrazie integral_binary

oraz analogicznie dla prawej strony

right_bottom_rect - wartość dolnego obszaru na obrazie integral_binary

right_top_rect - wartość górnego obszaru na obrazie integral_binary

Sprawdza się czy wartość obu cech jest mniejsza od progu o wartości 0,8. Dodatkowo wyliczana jest wartość value4 (3.22). Jej wartość musi być mniejsza bądź równa 0,65. Jeżeli średnia wartość obrazu binarnego dla obu oznaczonych górnych obszarów jest większa od 15% to value4 jest równe stosunkowi sumy wartości górnych obszarów do sumy wartości dolnych obszarów. W przeciwnym wypadku value4 równe jest 1, a tym samym klatka nie spełnia jednego z warunków klasyfikatora i zostaje odrzucona.

Dany klasyfikator można określić jako dokładniejszą wersję klasyfikatora z etapu trzeciego. Wszystkie klatki odrzucone na etapie trzecim zostałyby odrzucone również w czwartym etapie. Jednak ze względu na większą złożoność obliczeniową danego etapu został wprowadzony do algorytmu również klasyfikator zastosowany w etapie 3 (część 3.1.4.3), który niskim kosztem obliczeń odrzuca dużą ilość klatek nie-twarzy.

3.1.4.5. Analiza małych obszarów oczu oraz pod oczami (obraz w skali szarości)

Obszary wykorzystywane przez klasyfikator na etapie piątym posiadają jednakowe położenie oraz wymiary co w etapie czwartym. Wartość obszarów obliczana jest dla obrazu w skali szarości. Wartość cechy wyliczana jest jako stosunek sumy wartości obu powierzchni górnych do sumy powierzchni dolnych.

Klasyfikator sprawdza czy spełniony jest warunek:

$$\text{value5} < 0,95 \quad (3.25)$$

gdzie:

$$\text{value5} = \frac{\text{left_top_rect} + \text{right_top_rect}}{\text{left_bottom_rect} + \text{right_bottom_rect}} \quad (3.26)$$

Kolejne klasyfikatory na etapach od 6 do 9 sprawdzają czy wartość skali szarości dla określonych obszarów klatki nie jest zbyt zróżnicowana. Progi dla danych klasyfikatorów zostały dobrane po przeprowadzeniu szeregu doświadczeń. Obrazy wykorzystane w eksperymentach stanowiły zdjęcia zarówno wykonane w dobrych warunkach oświetlenia jak również wykonanych w pomieszczeniu o nierównym oświetleniu.

3.1.4.6. Analiza lewych ćwiartek klatki (obraz w skali szarości)

Na etapie szóstym obliczane są dwie powierzchnie o wysokości równej rozmiarowi klatki i szerokości równej $\frac{1}{4}$ rozmiaru klatki (oznaczone kolorem zielonym na rysunku 3.9)

Cecha obliczana jest jako stosunek mniejszej do większej wartości obliczonych powierzchni (w celu otrzymania wartości < 1).

Dopuszczalna różnica wyliczonych wartości wynosi 60% względem większej wartości. Klasyfikator weryfikuje czy dany warunek jest spełniony sprawdzając nierówność:

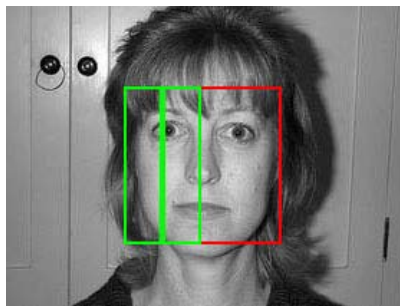
$$\text{value6} > 0,4 \quad (3.27)$$

gdzie:

$$\text{value6} = \begin{cases} \text{left_rect} & \text{jeśli } \text{left_rect} < \text{right_rect} \\ \text{right_rect} & \text{jeśli } \text{left_rect} > \text{right_rect} \end{cases} \quad (3.28)$$

left_rect – wartość lewego obszaru na obrazie integral_gray

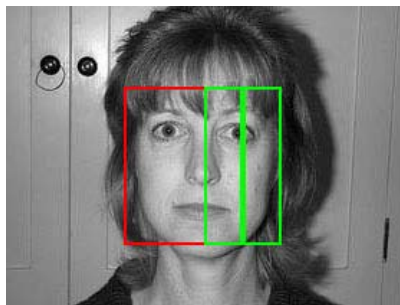
right_rect – wartość prawego obszaru na obrazie integral_gray



Rysunek 3.9: Obraz w skali szarości

3.1.4.7. Analiza prawych ćwiartek klatki (obraz w skali szarości)

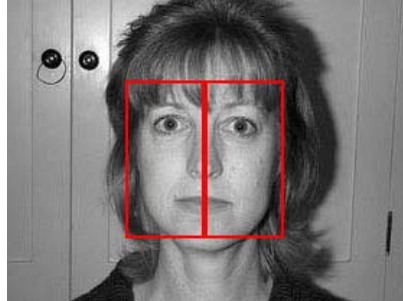
Klasyfikator na etapie siódmym jest analogiczny do poprzedniego, lecz wyliczane powierzchnie położone są po prawej stronie klatki (oznaczone kolorem zielonym na rysunku 3.10).



Rysunek 3.10: Obraz w skali szarości

3.1.4.8. Analiza pionowych połówek klatki (obraz w skali szarości)

Na etapie ósmym sprawdzana jest różnica wartości z obrazu w skali szarości pionowo zorientowanych połówek klatki (Rysunek 3.11).



Rysunek 3.11: Obraz w skali szarości.

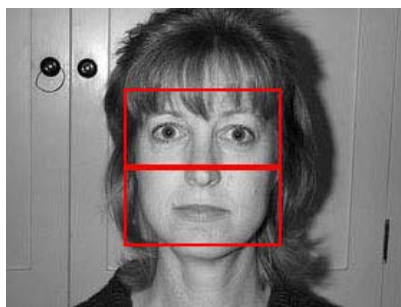
Dane wartości można otrzymać jako sumy wyliczonych wartości w etapach szóstym (dla lewej połowy klatki) i siódmym (dla prawej połowy klatki).

Podobnie jak dla poprzednich klasyfikatorów wartość cechy obliczana jest jako stosunek wartości mniejszej do większej. Dopuszczalna różnica wynosi 75% czyli wyznaczona wartość cechy musi być większa od progów równego 0,25. Tak duża dozwolona różnica wartości obu obszarów została wprowadzona ze względu na obrazy wykonane w pomieszczeniach, w których źródło oświetlenia znajdowało się po jednej stronie twarzy. W takich warunkach różnica intensywności pikseli po obu stronach twarzy może być dość duża.

3.1.4.9. Analiza poziomych połówek klatki (obraz w skali szarości)

Klasyfikator etapu dziewiątego sprawdza czy różnica wartości obrazu skali szarości dla powierzchni równych połowie wielkości klatki, zorientowanych poziomo (oznaczone na rysunku 3.12) jest mniejsza od dopuszczalnej wartości 40%. Dany warunek jest testowany sprawdzając czy wartość wyliczonej cechy jest większa od progów o wartości 0,6 .

Cecha analogicznie do poprzednich etapów 6-8, wyliczana jest jako stosunek mniejszej do większej wartości opisanych powyżej obszarów.



Rysunek 3.12: Obraz w skali szarości

W algorytmie nie zastosowano klasyfikatorów sprawdzających różnicę wartości na obrazie w skali szarości ćwierci klatki zorientowanych poziomo. Po przeprowadzeniu eksperymentów ustalono, że maksymalna różnica wartości dla takich obszarów wynosi nawet 90%. Powodem takich dużych rozbieżności był występujący zarost w dolnej części twarzy lub ciemne włosy przysłaniające czoło.

3.1.4.10. Analiza wartości obszarów oczu i brwi (obraz binarny)

Na etapie dziesiątym wykorzystano fakt, że na obrazie binarnym powierzchnie zawierające oczy i brwi po obu stronach twarzy posiadają zbliżoną wartość. Po przeprowadzeniu badań ustalono, że maksymalna różnica wielkości dla tych powierzchni wynosi 30%². Cecha obliczana jest jako stosunek ilości pikseli o wartości 1 z obrazu po lokalnej binaryzacji na obszarach zaznaczonych na Rys. Wymiary i położenie powierzchni wyliczane są ze wzorów³:

$$w=0,25*frame_size \quad (3.29)$$

$$h=0,2*frame_size \quad (3.30)$$

$$y=frame_y+0,1*frame_size \quad (3.31)$$

- dla powierzchni po lewej stronie

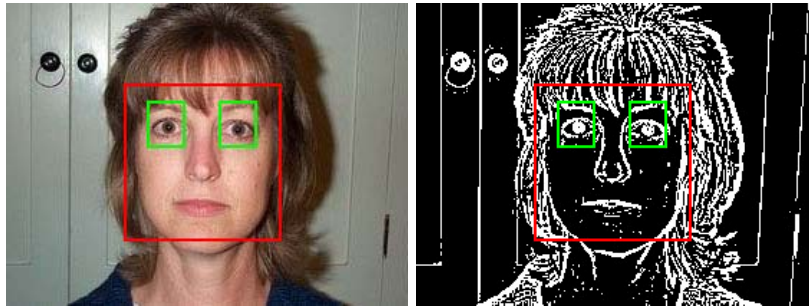
$$x_left=frame_x+0,15*frame_size \quad (3.32)$$

- dla powierzchni po prawej stronie

² Dla klatek usytuowanych centralnie na twarzy. Wykorzystane obrazy stanowiły zdjęcia wykonane w idealnych jak i przeciętnych warunkach oświetlenia.

³ Wartości współrzędnej x oraz szerokość dla danych powierzchni są równe tym wielkościom wyliczonym dla powierzchni w etapie czwartym. W celu redukcji wykonywanych obliczeń można wykorzystać dane obliczone w poprzednim etapie.

$$x_right=frame_x+0.6*frame_size \quad (3.33)$$



Rysunek 3.13: Obraz w formacie RGB oraz po lokalnej binaryzacji

Klatka zostaje sprawdzona jeśli spełniony jest jeden z warunków:

$$value10 \geq 0.7 \text{ lub } \begin{cases} \frac{left_rect}{w * h} < 0.05 \\ \frac{right_rect}{w * h} < 0.05 \end{cases} \quad (3.34)$$

gdzie:

$$value10 = \begin{cases} \frac{left_rect}{right_rect} & \text{jeśli } left_rect < right_rect \\ \frac{right_rect}{left_rect} & \text{jeśli } left_rect > right_rect \end{cases} \quad (3.35)$$

left_rect – wartość lewego obszaru na obrazie integral_binary

right_rect – wartość prawego obszaru na obrazie integral_binary

Dopuszczalna minimalna wartość cechy oznaczonej value10 wynosi 0,7. Dodatkowo obliczane są średnie wartości ilości pikseli o wartości 1 na obrazie binarnym dla obu powierzchni. Jeżeli obie otrzymane wielkości są mniejsze od progu wynoszącego 0,05 to pomimo nie spełnienia głównego warunku dla cechy z danego etapu klatka zostaje przepuszczona.

Na obrazach binarnych zawierających twarz osoby o słabo zarysowujących się brwiach i jasnych oczach ilość pikseli o wartości 1 w wyznaczonych obszarach może być niewielka. W takim przypadku nieznaczna różnica wielkości kilku, kilkunastu pikseli może stanowić dużą różnicę procentową pomiędzy tymi

obszarami. Dodatkowy warunek został wprowadzony w celu zlikwidowania opisanego problemu.

3.1.4.11. Analiza obszarów policzków (obraz binarny)

Na etapie jedenastym sprawdzana jest ilość pikseli oznaczonych wartością 1 na obrazie binarnym w obszarach, które w klatce z twarzą znajdują się w rejonach policzków.

Można zauważyć, że dla obrazów wykonanych w idealnym oświetleniu piksele leżące w obszarze policzków nie otrzymają wartości 1 przy lokalnej binaryzacji obrazu. Nawet w obrazach o nierównym oświetleniu procentowa wartość ilości pikseli o wartości 1 na całym obszarze stanowi maksymalnie 25% powierzchni. Ten fakt wykorzystany jest na etapie jedenastym jako kolejny warunek, który musi zostać spełniony przez klatkę zawierającą twarz.

Położenie oraz wymiary wykorzystanych powierzchni wyliczane są ze wzorów:

$$y = \text{frame_y} + 0,4 * \text{frame_size}^4 \quad (3.36)$$

$$w = 0,225 * \text{frame_size} \quad (3.37)$$

$$h = 0,2 * \text{frame_size} \quad (3.38)$$

- lewy obszar

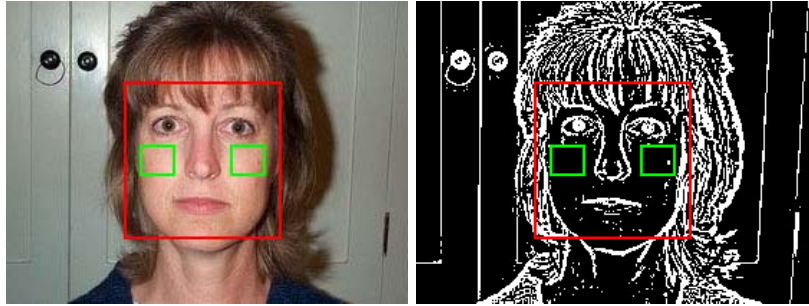
$$x_{\text{left}} = \text{frame_x} + 0,1 * \text{frame_size} \quad (3.39)$$

- prawy obszar

$$x_{\text{right}} = \text{frame_x} + \text{frame_size} - 0,1 * \text{frame_size} - s_8_w^5 = \text{frame_x} + 0,675 * \text{frame_size} \quad (3.40)$$

⁴ Powierzchnie zawierające policzki usytuowane są pod obszarami zawierającymi oczy, określonymi w etapie czwartym.

⁵ Wzór został wyprowadzony w celu pokazania, że powierzchnie po obu stronach znajdują się w jednakowej odległości od odpowiednich boków klatki.



Rysunek 3.14 a i b: Obraz w formacie RGB oraz po lokalnej binaryzacji

Klasyfikator sprawdza czy zachodzi następująca nierówność:

$$\text{value11} \leq 0,25 \quad (3.41)$$

gdzie:

$$\text{value11} = \frac{\frac{\text{left_rect}}{w * h} + \frac{\text{right_rect}}{w * h}}{2} \quad (3.42)$$

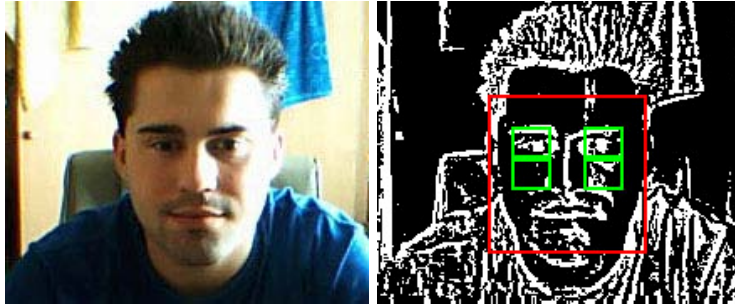
left_rect – wartość lewego obszaru z obrazu binarnego

right_rect - wartość prawego obszaru z obrazu binarnego

3.1.4.12. Klasyfikator sumaryczny nr 1

Dany klasyfikator sprawdza czy suma iloczynów wybranych wartości wyliczonych w poprzednich etapach i odpowiednich współczynników jest większa od określonego progu.

Pewne cechy wyliczone w poprzednich etapach można uznać za wyznaczniki jakości twarzy. Przykładowo cechy wyliczone na etapie czwartym dla klatki z twarzą obrazu przedstawionego na Rys będą posiadać wartość bliską zeru. Górne obszary posiadają dużo większą wartość od dolnych, w których znajduje się niewiele pikseli o wartości 1. Dla twarzy przedstawionej na rysunku 3.15, który stanowi przykład obrazu o gorszej jakości, wartość cech klasyfikatora czwartego (część 3.1.4.4) będą większe niżeli dla twarzy z rysunku 3.8.



Rysunek 3.15 a i b: Przykład obrazu wykonanego w pomieszczeniu z bocznym źródłem światła

Jako wielkości wyznaczające jakość twarzy uznano wartości wyliczone w etapach 4, 5, 10 i 11 oznaczone odpowiednio $value_4$, $value_5$, $value_{10}$ oraz $value_{11}$. Jako, że w części etapów mniejsza wyliczona wartość oznaczała twarz o lepszej jakości wprowadzono normalizację. Wprowadzone nowe wartości określone są wzorami:

$$v_4 = 1 - value_4 \quad (3.43)$$

$$v_5 = 1 - value_5 \quad (3.44)$$

$$v_{10} = value_{10} \quad (3.45)$$

$$v_{11} = 1 - value_{11} \quad (3.46)$$

Klasyfikator sprawdza nierówność:

$$phase1 \geq 2,9 \quad (3.47)$$

$$phase1 = w_4 * v_4 + w_5 * v_5 + w_{10} * v_{10} + w_{11} * v_{11} \quad (3.48)$$

Współczynniki w zostały dobrane empirycznie. Ich wartości wynoszą:

$$w_4 = 1,4$$

$$w_5 = 2$$

$$w_{10} = 1$$

$$w_{11} = 1$$

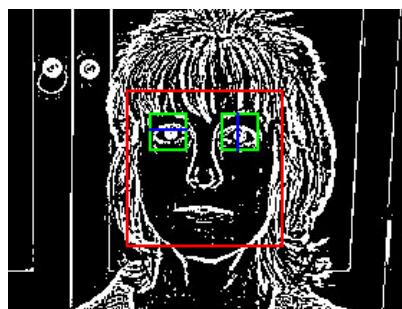
3.1.4.13. Analiza regionów oczu (obraz binarny)

Na etapie 12 sprawdzana jest procentowa ilość pikseli o wartości 1 w rejonach oczu. Na całej wysokości, dla każdego z obszarów oczu musi znajdować się określona minimalna ilość prostokątów o wysokości 1 i szerokości obszaru, w którym ilość pikseli o wartości 1 jest większa od empirycznie ustalonego progu. Analogicznie na całej szerokości musi znajdować się minimalna ilość prostokątów o szerokości 1 i wysokości obszaru, w których ilość pikseli o wartości 1 jest większa od pewnego progu. Dodatkowo różnica ilości wymienionych obszarów pionowych i poziomych w obszarach obu oczu nie może przekroczyć pewnego progu. Dane sprawdzenie zapewnia wyszukiwanie klatek z obszarami oczu o zbliżonym kształcie i wielkości ciemnych obszarów odpowiadających oczom.

Wartość progu dla pojedynczego prostokąta o wysokości 1 i szerokości obszaru równa jest $0,4 * \text{szerokosc_obszaru}$. W każdym z obszarów, minimum 5% prostokątów musi spełnić dany warunek.

Wartość progu dla prostokąta o szerokości 1 i wysokości obszaru równa jest $0,01 * \text{wysokosc_obszaru}$. W każdym z obszarów, minimum 70% prostokątów musi spełnić dany warunek.

Różnica pomiędzy ilością poziomych prostokątów o wysokości 1 spełniających powyższe warunki, w obu obszarach nie może przekroczyć 22%. Analogicznie dla prostokątów pionowych.



Rysunek 3.16: Obraz binarny.

Dla danego etapu wyliczana jest wartość

$$\text{value13} = \text{v13_h_left} + \text{v13_h_right} + \text{v13_v_left} + \text{v13_v_right} \quad (3.49)$$

gdzie:

v13_h_left i v13_h_right –procentowa wartość prostokątów poziomych spełniających opisane warunki dla odpowiednio lewego i prawego obszaru

v13_v_left i v13_v_right –procentowa wartość prostokątów pionowych spełniających opisane warunki dla odpowiednio lewego i prawego obszaru

3.1.4.14. Analiza regionu nosa (obraz binarny)

Operacje etapu trzynastego mają na celu sprawdzenie czy w określonej klatce znajduje się obszar zawierający kształt odpowiadający kształtowi nosa.

Wymiary badanego obszaru:

Współrzędne lewego górnego rogu

$$x = \text{frame_x} + 0,4 * \text{frame_size} \quad (3.50)$$

$$y = \text{frame_x} + 0,25 * \text{frame_size} \quad (3.51)$$

-wysokość

$$h = 0,4 * \text{frame_size} \quad (3.52)$$

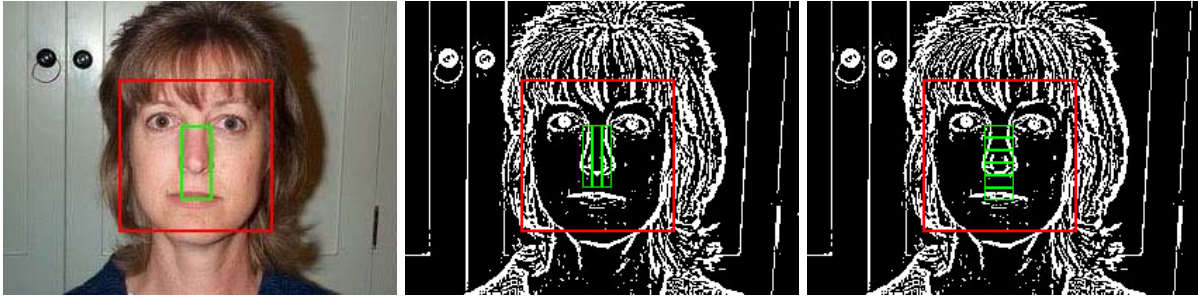
-szerokość

$$w = 0,2 * \text{frame_size} \quad (3.53)$$

Szerokość i wysokość badanego obszaru została podzielona odpowiednio na 3 i 5 części.

W pierwszej kolejności badane są prostokąty poziome. Jeżeli w czterech kolejnych prostokątach minimum 10% pikseli danego obszaru posiada wartość 1 oraz ostatni z danych prostokątów w 3 częściach utworzonych przez jego podzielenie względem szerokości posiada piksele o wartości 1 to wartość etapu 14 value14 zostaje zwiększona o 1.

Jeżeli zewnętrzne pionowe prostokąty – lewy i prawy posiadają minimum 30% pikseli o wartości 1 na obrazie binarnym oraz prostokąt środkowy posiada mniej niż 30% pikseli o wartości 1 to wartość danego etapu value13 zostaje zwiększona o 1. Dana część etapu nie musi zostać spełniona.



Rysunek 3.17 a, b i c: Kolejno rysunki w kolorze RGB oraz po binaryzacji. Kolorem zielonym oznaczono badane obszary w etapie 14.

3.1.4.15. Analiza obszaru ust (obraz binarny)

W etapie 15 poszukiwany jest obszar odpowiadający ustom, zawierający określoną ilość pikseli o wartości 1 na obrazie binarnym.

W celu ustalenia pozycji ust przeszukiwany jest obszar, którego współrzędne lewego górnego rogu wynoszą:

$$x = \text{frame_x} + 0,25 * \text{frame_size} \quad (3.54)$$

y – równa jest wartość dolnej granicy odnalezionego w etapie 14 nosa.

- szerokość

$$w = 0,5 * \text{frame_size} \quad (3.55)$$

- wysokość (od początku przeszukiwanego obszaru do dolnej krawędzi całej badanej klatki)

$$h = \text{frame_y} + \text{frame_size} - y \quad (3.56)$$

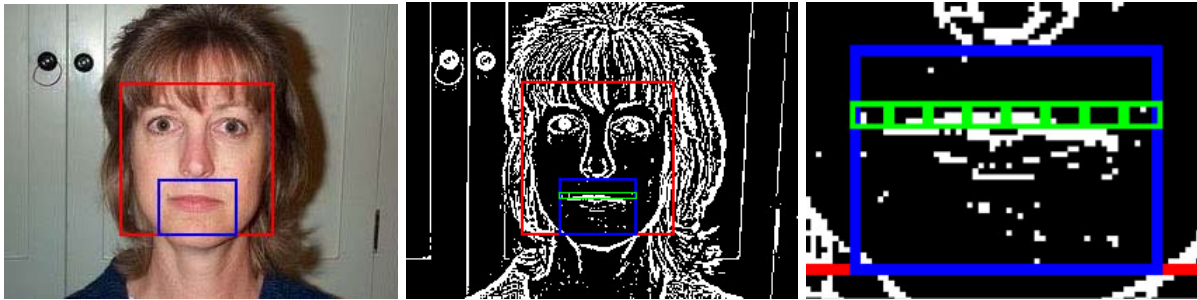
W obszarze o opisanych wymiarach poszukiwany jest prostokąt o wysokości równej

$$h1 = 0,05 * \text{frame_size} \quad (3.57)$$

którego procentowa wartość pikseli o wartości 1 na obrazie binarnym wynosi minimum 20% oraz jest większa od $1,4 * \text{srednia_z_obszaru}$.

Gdzie srednia_z_obszaru jest średnią wartością pikseli o wartości 1 w całym przeszukiwanym obszarze.

Znaleziony prostokąt dzielony jest na 8 mniejszych części. Jeżeli przynajmniej w 3 kolejnych znajduje się powyżej 20% pikseli o wartości 1 to sprawdzane jest czy powyżej oraz poniżej danego prostokąta można odnaleźć prostokąt w którym odpowiadające małe części będą posiadały mniej niż 67% białych pikseli w stosunku do wartości w badanym prostokącie. Dane sprawdzenie zapobiega wskazaniu jako ust obszaru zarostu, który jako bardzo ciemny po binaryzacji.



Rysunek 3.18 a, b i c: Obraz w formacie RGB oraz obraz binarny

3.1.4.16. Analiza różnicy wartości obszaru ust i policzków (obraz binarny)

W danym etapie obliczana jest różnica pomiędzy średnią ilością pikseli o wartości 1 z obrazu binarnego w obszarze etapu 15 (usta) i w obszarze etapu 11 (policzki).

$$\text{value16} = v16_1 - v16_2 \quad (3.58)$$

gdzie:

$v16_1$ i $v16_2$ równe są odpowiednio średniej ilości pikseli o wartości 1 w obszarach ust i policzków

Wyliczona wartość musi być spełniać nierówność

$$\text{value16} \leq 0,4 \quad (3.59)$$

3.1.4.17. Klasyfikator sumaryczny nr 2

W drugim etapie sumarycznym sprawdzana jest nierówność

$$\text{phase2} \geq 5,2 \quad (3.60)$$

$$\text{phase2} = \text{phase1} + w_{13} * v_{13} + w_{14} * v_{14} + w_{16} * v_{16} \geq 2,9 \quad (3.61)$$

Współczynniki wynoszą

$$w_{13}=0,35$$

$$w_{14}=1$$

$$w_{16}=1,2$$

Wartości v są równe:

$$v_{13}=\text{value13} \quad (3.62)$$

$$v_{14}=\text{value14} \quad (3.63)$$

$$v_{16}=1-\text{value16} \quad (3.64)$$

3.1.4.18. Analiza zewnętrznych części obszarów oczu - w poziomie (obraz binarny)

Na danym etapie poszukiwane są prostokąty po zewnętrznych stronach obszarów zawierających oczy (położenie oraz wymiary opisane w etapie 4 – 3.1.4.4) zawierających maksymalnie 20% pikseli o wartości 1 na obrazie binarnym.

Wymiary poszukiwanych prostokątów:

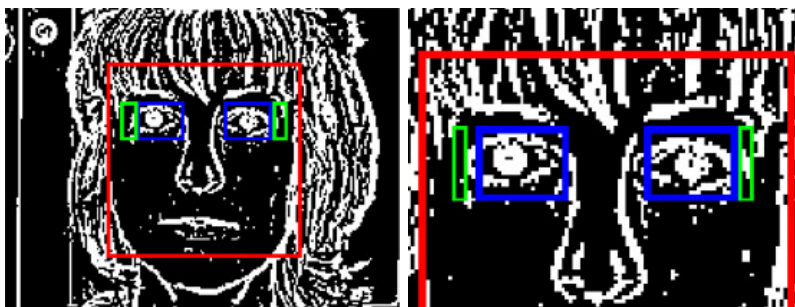
- szerokość

$$w=0,04*\text{frame_size} \quad (3.65)$$

- wysokość (równa wysokości obszarów zawierających oczy)

$$h=0,2*\text{frame_size} \quad (3.66)$$

Dane cechy (prostokąty) poszukiwane są na połowie obszaru - po lewej stronie, od lewej krawędzi lewego obszaru zawierającego oko do lewej krawędzi ramki z twarzą, analogicznie po prawej, od prawej krawędzi prawego obszaru zawierającego oko do prawej krawędzi ramki z twarzą (obszary oznaczono na rysunku 3.19 a i b).

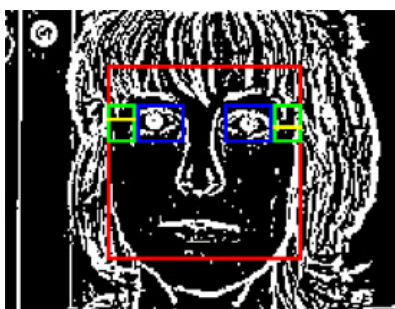


Rysunek 3.19 a i b: Obraz binarny.

3.1.4.19. Analiza zewnętrznych części obszarów oczu - w pionie (obraz binarny)

W etapie 19 całe obszary po zewnętrznej stronie obszarów zawierających oczy do krawędzi ramki z twarzą „skanowane” są w pionie prostokątem o szerokości obszaru i wysokości 1. Na rysunku 3.20 kolorem niebieskim oznaczono obszary zawierające oczy. Kolorem zielonym obszary skanowane prostokątem o wysokości 1 – przykładowe oznaczono kolorem żółtym.

Dla obu obszarów wyliczana jest średnia ilość pikseli o wartości 1 na obrazie binarnym. Jeżeli w skanowanym obszarze znajduje się prostokąt o wysokości 1, w którym ilość pikseli o wartości 1 jest większa od 90% średniej wartości danego obszaru to ramka zostaje odrzucona.

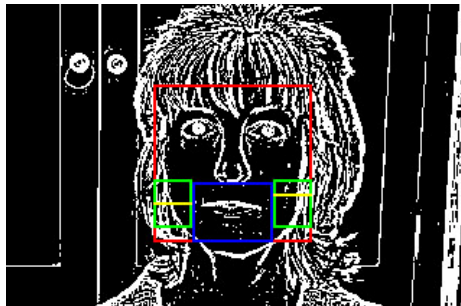


Rysunek 3.20: Obraz binarny

3.1.4.20. Analiza zewnętrznych części obszaru ust – w pionie (obraz binarny)

Na etapie 20 analogicznie do procedury etapu 19 „skanowane” są obszary po zewnętrznej części obszaru zawierającego usta. Jeżeli wartość jednego

z prostokątów o wysokości 1 zawierającego się w danych obszarach jest większa od 70% wartości całego obszaru, klatka zostaje odrzucona.



Rysunek 3.21: Obraz binarny

3.1.4.21. Analiza bocznych krawędzi ramki (obraz binarny)

Dany klasyfikator jest pierwszym z serii słabych klasyfikatorów. Opisane w kolejnych etapach warunki nie muszą zostać spełnione. Dla każdego z tych etapów wyliczone zostają pewne wartości, których wartość sprawdzana jest przez ostatni klasyfikator sumaryczny.

Na danym etapie sprawdzane czy w pobliżu bocznych krawędzi ramki znajdują się rysy kształtu twarzy. W tym celu badane są cztery regiony oznaczone kolorami niebieskim i zielonym na rysunku.

Wymiary danych obszarów równe są:

- szerokość

$$w = 0,1 * \text{frame_size} \quad (3.67)$$

$$h = 0,3 * \text{frame_size} \quad (3.68)$$

Wartość współrzędnej y górnej krawędzi obszarów równe jest

$$y = \text{frame_y} + 0,4 * \text{frame_size} \quad (3.69)$$

Wartość współrzędnej x lewego górnego rogu obszarów oznaczonych kolorem niebieskim na rysunku 3.22.:

- dla lewego obszaru

$$x = \text{frame_x} \quad (3.70)$$

- dla prawego obszaru

$$x = \text{frame_x} + \text{frame_size} - w \quad (3.71)$$

Wartość współrzędnej x lewego górnego rogu obszarów oznaczonych kolorem zielonym na rysunku 3.22.:

- dla lewego obszaru

$$x = \text{frame_x} + w \quad (3.72)$$

- dla prawego obszaru

$$x = \text{frame_x} + \text{frame_size} - 2 * w \quad (3.73)$$

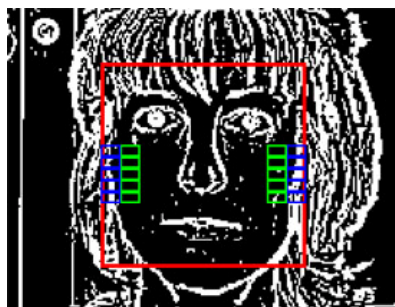
Każdy z czterech badanych obszarów podzielony został na 6 równych prostokątów. Dla każdego z nich wyliczana jest wartość na obrazie binarnym.

Dla każdej ze stron sprawdzany jest warunek:

$$\begin{cases} \text{blue_counter} > 3 \\ \text{green_counter} == 0 \end{cases} \quad (3.74)$$

Gdzie blue_counter i gree_counter – ilość prostokątów zawierających się w pojedynczym regionie oznaczonym na rysunku 3.22. odpowiednio kolorem niebieskim i zielonym zawierających więcej niż 15% pikseli o wartości 1

Jeżeli spełniony są warunki opisane wzorem (3.74) to wartość etapu zwiększana jest o 1⁶.



Rysunek 3.22: Obraz binarny

⁶ Warunek sprawdzany jest dla każdej ze stron, przez co wartość danego klasyfikatora może wynosić maksymalnie 2.

3.1.4.22. Analiza obszaru oczu i pomiędzy oczami (obraz binarny)

Wartość danego klasyfikatora określana jest w zależności od wartości obszarów zawierających oczy oraz pomiędzy oczami, oznaczonymi kolorem zielonym na rysunku 3.23. Sprawdzane jest czy obszary zawierające oczy posiadają większą wartość na obrazie binarnym większą od zera oraz czy jest większa od wartości obszaru pomiędzy oczami.

Wyliczane są wartości obszaru obejmującego wszystkie trzy prostokąty oraz wartość prostokąta środkowego.

Wymiary i współrzędne lewego górnego rogu całego obszaru oznaczonego na rysunku 3.23. kolorem zielonym:

$$x = \text{frame_x} + 0,22 * \text{frame_size} \quad (3.75)$$

$$y = \text{frame_y} + 0,15 * \text{frame_size} \quad (3.76)$$

$$w = 0,56 * \text{frame_size} \quad (3.77)$$

$$h = 0,25 * \text{frame_size} \quad (3.78)$$

Wymiary oraz współrzędna x środkowego obszaru:

$$w1 = w/3 \quad (3.79)$$

$$x1 = x + w1 \quad (3.80)$$

Sprawdzany jest warunek:

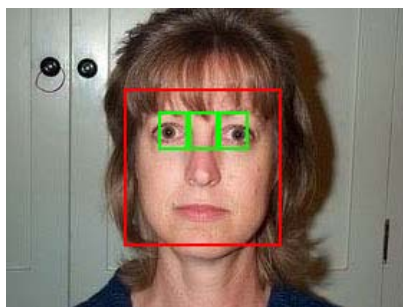
$$\begin{cases} \text{rect_eyes} \geq \text{rect_center} \\ \text{rect_eyes} > 0 \end{cases} \quad (3.81)$$

gdzie:

$$\text{rect_eyes} = (\text{rect_all} - \text{rect_center}) / 2 \quad (3.82)$$

rect_center – wartość środkowego obszaru

rect_all – wartość całego obszaru



Rysunek 3.23: Obraz w formacie RGB.

Jeżeli wyliczone wartości obszarów spełniają warunek (3.80) to wartość klasyfikatora⁷ obliczana jest ze wzoru:

$$\text{value22} = 1 - \text{rect_center} / \text{rect_eyes} \quad (3.83)$$

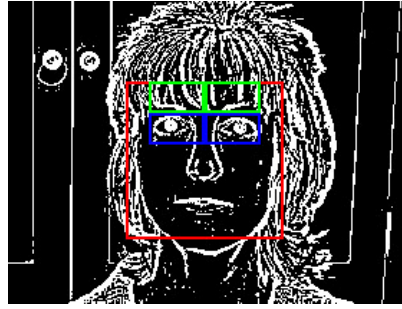
3.1.4.23. Analiza obszaru oczu oraz czoła (obraz binarny)

Na danym etapie klasyfikator sprawdzane są obszary oczu oraz nad oczami. W przypadku odśloniętego czoła, regiony usytuowane nad oczami na obrazie binarnym będą posiadały wartości bliskie zero i jednocześnie dużo mniejsze od wartości obszarów zawierających oczy. Na rysunkach 3.24. a i b widać, że w opisanych obszarach górnych na obrazie binarnym znajduje się bardzo niewielka ilość pikseli o wartości 1 (oznaczonych kolorem białym). Na rysunku 3.25. zaprezentowano przypadek, w którym dane warunki nie zostaną spełnione.



Rysunek 3.24 a i b: Obraz w formacie RGB oraz obraz binarny.

⁷ Jeżeli warunek etapu jest niespełniony to wartość klasyfikatora równa jest 0



Rysunek 3.25: Obraz binarny.

Wymiary obszarów:

$$h=0,2*frame_size \quad (3.84)$$

$$w=0,35*frame_size \quad (3.85)$$

- lewy górny obszar

$$x1=frame_x+0,15*frame_size \quad (3.86)$$

$$y1=frame_y \quad (3.87)$$

- lewy dolny obszar

$$x2=x1 \quad (3.88)$$

$$y2=frame_y+h \quad (3.89)$$

- prawy górny obszar

$$x3=x1+w \quad (3.90)$$

$$y3=frame_y \quad (3.91)$$

- prawy dolny obszar

$$x4=x3 \quad (3.92)$$

$$y4=y2 \quad (3.93)$$

Wyliczane są wartości danych obszarów na obrazie binarnym.

Dla obu stron sprawdzany jest trzy warunki opisane wzorem (3.94):

- czy wartość dolnego obszaru jest większa od zera
- czy wartość górnego obszaru jest mniejsza od wartości dolnego obszaru
- czy średnia wartość górnego obszaru jest mniejsza od 30%

$$\begin{cases} \text{rect_bottom} > 0 \\ \text{rect_top} < \text{rect_bottom} \\ \text{rect_top}/(w * h) < 0.3 \end{cases} \quad (3.94)$$

gdzie:

rect_bottom – wartość obszaru dolnego dla każdej ze stron

rect_top – wartość obszaru górnego dla każdej ze stron

Jeżeli dla obu stron spełnione są warunki opisane wzorem 3.94 to wartość etapu wyliczana jest ze wzoru

$$\text{value23} = (1 - \text{rect_top1}/\text{rect_bottom1}) + (1 - \text{rect_top2}/\text{rect_bottom2}) \quad (3.95)$$

Gdzie rect_top1 i rect_top2 równe są wartości obszarów górnym odpowiednio lewego i prawego. Analogicznie wartości rect_bottom1 i rect_bottom2 równe są wartościom obszarów dolnych.

3.1.4.24. Analiza obszaru oczu i brwi (obraz binarny)

Klasyfikator danego etapu sprawdza czy pomiędzy obszarami odpowiadającymi brwiom oraz oczom znajduje się przestrzeń o małej wartości na obrazie binarnym.

Przeszukiwane są obszary o wymiarach:

$$w = 0,2 * \text{frame_size} \quad (3.96)$$

$$h = 0,3 * \text{frame_size} \quad (3.97)$$

i współrzędnych

$$y = \text{frame_y} + 0,1 * \text{frame_size} \quad (3.98)$$

- dla lewego obszaru

$$x = \text{frame_x} + 0,15 * \text{frame_size} \quad (3.99)$$

- dla prawego obszaru

$$x = \text{frame_x} + 0,65 * \text{frame_size} \quad (3.100)$$

Od góry do dołu każdego z obszarów wymiarach opisanych powyższymi wzorami, oznaczonych na rysunku 3.26a kolorem zielonym wyliczane są wartości 3

prostokątów położonych jeden pod drugim. Przykładowe prostokąty zostały oznaczone na rysunku 3.26b.

Wysokość prostokątów górnego i dolnego równa jest

$$h1=0,08*frame_size \quad (3.101)$$

Wysokość prostokąta środkowego wyliczana jest ze wzoru

$$h2=0,01*frame_size \quad (3.102)$$

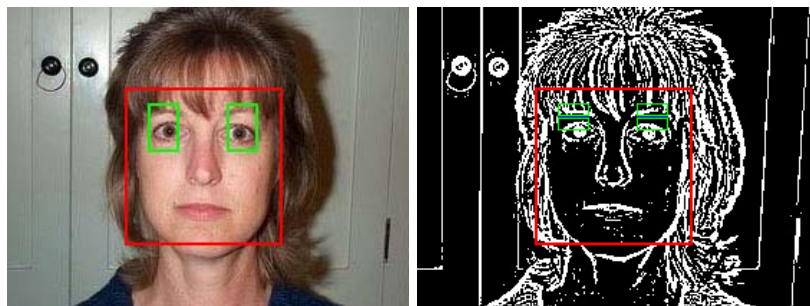
Dla obu obszarów sprawdzane jest czy istnieje dany zestaw prostokątów, których wartości na obrazie binarnym spełniają warunki

$$\begin{cases} \text{rect_top_av} > 0,2 \\ \text{rect_bottom_av} > 0,2 \\ \text{rect_center_av} / \text{rect_top_av} \leq 0,6 \\ \text{rect_center_av} / \text{rect_top_av} \leq 0,6 \end{cases} \quad (3.103)$$

gdzie:

rect_top_av, rect_bottom_av i rect_center_av równe są średnim wartościom prostokątów odpowiednio górnego, dolnego i środkowego

Jeżeli na badanym obszarze znajdują się prostokąty spełniające warunki 3.103 to wartość etapu zostaje zwiększona o 1.



Rysunek 3.26 a i b: Obraz w formacie RGB oraz binarny.

3.1.4.25. Klasyfikator sumaryczny nr 3

Wartość klasyfikatora sumarycznego wyliczana jest ze wzoru

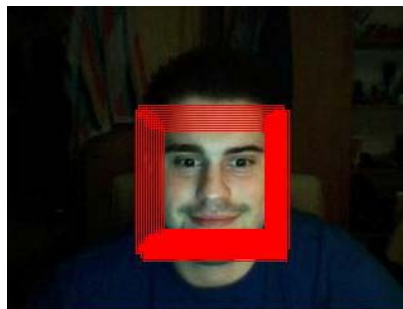
$$\text{phase3}=\text{phase2}+w21*\text{value21}+w22*\text{value22}+w23*\text{value23}+w24*\text{value24} \quad (3.104)$$

Jeżeli wartość $phase3$ spełnia nierówność opisaną wzorem 3.105 klatka zostaje zachowana jako zawierająca twarz.

$$phase3 \geq 9,3 \quad (3.105)$$

3.1.5. Wybór optymalnej klatki dla pojedynczej twarzy

Często dla pojedynczej twarzy zostaje odnalezione nawet do 15 klatek różniących się nieznacznie położeniem i rozmiarem, które spełniają wymagane, opisane powyżej warunki. Wszystkie z odnalezionych klatek są porównywane. Jeżeli posiadają minimum 60% wspólnej powierzchni to zostaje zatrzymana ramka, której wyliczona wartość jest wyższa. Daną metodą uzyskuje się pojedyncze klatki dla wszystkich znalezionych twarzy. Jako, że system korzysta z informacji o jednej twarzy do dalszej części zostaje przekazana ramka, dla której odległość środka do środka obrazu jest najmniejsza.



Rysunek 3.27: Wykryte klatki spełniające warunek (3.105)

3.2. Lokalizacja twarzy w czasie rzeczywistym

Wydajność opracowanego algorytmu lokalizacji twarzy dla obrazu o rozdzielczości 320x240 pikseli na komputerze klasy PC z dwurdzeniowym procesorem 2GHz oraz 2GB pamięci RAM wynosi średnio 8 klatek / s. Jest to zadowalający wynik umożliwiający stałe śledzenie twarzy oraz szybką reakcję na ruch oraz zmianę mimiki poprzez ruch cyberagenta.

Stale wykonywane operacje lokalizacji twarzy są dużym obciążeniem dla procesora. Użytkownik komputera utrzymuje zwykle stałą pozycję głowy przed monitorem wykonując jedynie nieznaczne ruchy. W pracy [11] do algorytmu

lokalizacji twarzy dołączono wykrywanie ruchu, co znacznie odciąża pracę procesora. Jeśli nie zostanie zarejestrowany ruch, algorytm lokalizacji twarzy nie zostaje wykonany. W przypadku wcześniejszego wykrycia twarzy, zostaje uznane, że znajduje się ona w stałym, pierwotnym położeniu.

3.2.1. Wykrywanie ruchu

W przypadku obrazów ruch można zdefiniować, jako różnicę pikseli na obrazach zarejestrowanych w dwóch jednostkach czasu. Zgodnie z algorytmem opisanym w pracy [11], pojedynczy punkt obrazu zostaje uznany za punkt, który uległ przemieszczeniu, w przypadku gdy bezwzględna różnica pomiędzy wartościami danego punktu na obrazach w skali szarości zarejestrowanymi w pewnym odstępie czasu jest większa od pewnego progu – tzw. progu detekcji ruchu. Dany warunek został wprowadzony ze względu na zaszumiony obraz uzyskiwany z kamery internetowej. Często uzyskiwane kolejne klatki obrazu różnią się w pewnym stopniu wartością natężenia koloru oraz jasnością. Po serii doświadczeń za optymalną wartość progu przyjęto 20% średniej wartości aktualnego obrazu w skali szarości.

Dany warunek wyraża się wzorem

$$|\text{prev_gray}(x,y) - \text{gray}(x,y)| > 0,2 * \frac{\sum_{i=1}^w \sum_{j=1}^h \text{gray}(i,j)}{w * h} \quad (3.106)$$

gdzie

$\text{gray}(x,y)$ – wartość obrazu w skali szarości dla punktu o współrzędnych (x,y)

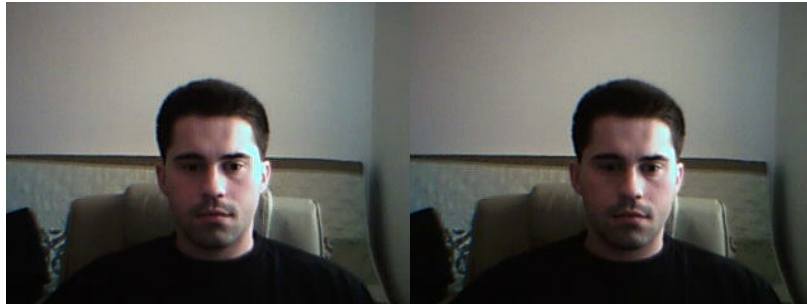
$\text{prev_gray}(x,y)$ – wartość poprzednio zarejestrowanego obrazu w skali szarości dla punktu o współrzędnych (x,y)

w – szerokość obrazu

h – wysokość obrazu

Wykrywanie ruchu oparte na różnicy wartości pikseli pomiędzy dwoma ostatnio zarejestrowanymi obrazami nie daje oczekiwanej skuteczności. Zrealizowany w dany sposób algorytm nie wykrywał bardzo wolno przemieszczających się

przedmiotów. Wykorzystując do porównania, jako poprzedni, obraz zapisany w czasie ostatniej rejestracji ruchu można uzyskać skuteczny algorytm.



Rysunek 3.28 a i b: Obrazy, dla których wyznaczano ruch.



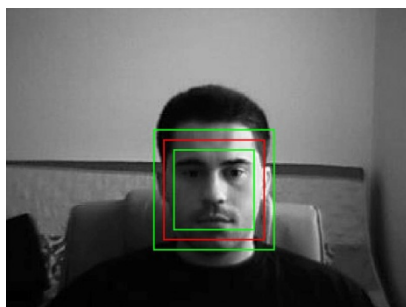
Rysunek 3.29: Obraz z wykrytym ruchem. Piksele uznane, za przemieszczone zostały oznaczone kolorem białym.

Obraz z oznaczonymi pikselami powyżej progu detekcji ruchu zapisywany jest w postaci binarnej.

Na rysunku 3.30 czerwonym kolorem oznaczona jest wykryta klatka z twarzą. Kolorem zielonym oznaczono wewnętrzną oraz zewnętrzną krawędź ramki, względem wartości której sprawdzane jest czy twarz uległa przemieszczeniu. Wewnętrzna oraz zewnętrzna krawędź danej ramki znajduje się w odległości od krawędzi klatki z twarzą równej

$$r = 0,1 * \text{frame_size} \quad (3.107)$$

Jeżeli suma pikseli o wartości 1 na obrazie z oznaczonym ruchem, zawartych w danej ramce stanowi więcej niż 17% powierzchni ramki to twarz zostaje uznana za przemieszczoną. W takim przypadku następuje ponowne przeszukiwanie obrazu w celu lokalizacji twarzy.



Rysunek 3.30: Ramka detekcji przemieszczenia twarzy

W przypadku braku zlokalizowanej na obrazie twarzy przeszukiwany jest cały obraz z uwzględnieniem ruchu. Dla każdej klatki obliczana jest procentowa ilość pikseli, które zostały oznaczone na binarnym obrazie ruchu. Jeżeli dana wartość przekracza 17% to klatka – „kandydat na twarz” sprawdzana jest algorytmem opisanym w części 3.1.

3.3. Mimika twarzy

Stan emocjonalny twarzy określany jest za pomocą wielu charakterystycznych cech kształtu i położenia oczu, ust oraz brwi. W pracy zrealizowano proste rozpoznawanie stanu mimiki twarzy na podstawie informacji o położeniu i kształcie ust oraz oczu.

Do najważniejszych gestów twarzy należy uśmiech. Jest to prosty, niewerbalny sposób na zwrócenie uwagi drugiej osoby przy pierwszym kontakcie. Dodatkowym gestem, często wykorzystywanym przy kontaktach międzyludzkich jest mrugnięcie. Zwykle kojarzone jest jako wyraz sympatii oraz zaufania.

3.3.1. Lokalizacja ust

W pierwszym etapie należy określić dokładne położenie ust. Na etapie 14 algorytmu lokalizacji twarzy ustalono przybliżoną wartość współrzędnych ust. W celu dokładnego ustalenia pozycji ust przeszukiwany jest obszar o rozmiarach:

- szerokość

$$w = 0,4 * \text{frame_size} \quad (3.108)$$

- wysokość

$$h=0,2*frame_size \quad (3.109)$$

i współrzędnych

$$x = frame_x+0,3*frame_size \quad (3.110)$$

$$y=y0-0,1*frame_size \quad (3.111)$$

gdzie

y_0 – wartość współrzędnej y wykrytej w etapie 15 pozycji ust

Na danym obszarze poszukiwany jest prostokąt (cecha) o wymiarach

- szerokość

$$w1=w \quad (3.112)$$

- wysokość

$$h1=0,02*frame_size \quad (3.113)$$

o największej wartości wyrażonej wzorem

$$value = 0,5 * (rect_top - rect) + 0,5 * (rect_bottom - rect) \quad (3.114)$$

gdzie

$rect$ – wartość sumy pikseli zawierających się w sprawdzanym środkowym prostokącie na obrazie w skali szarości

$rect_top$ - wartość sumy pikseli zawierających się w prostokącie górnym

$rect_bottom$ - wartość sumy pikseli zawierających się w prostokącie dolnym



Rysunek 3.31: Obraz w skali szarości z wykrytymi ustami.

3.3.2. Wykrywanie kształtu ust

Po lokalizacji ciemnej przestrzeni pomiędzy wargami wyliczana jest wartość kwadratu w pozycji środkowej ust, o współrzędnych

$$x = \text{frame_x} + 0,5 * \text{frame_size} - 0,5 * \text{edge_size} \quad (3.115)$$

y – równe współrzędnej y zlokalizowanej pozycji ust metodą opisaną w części 3.3.1.

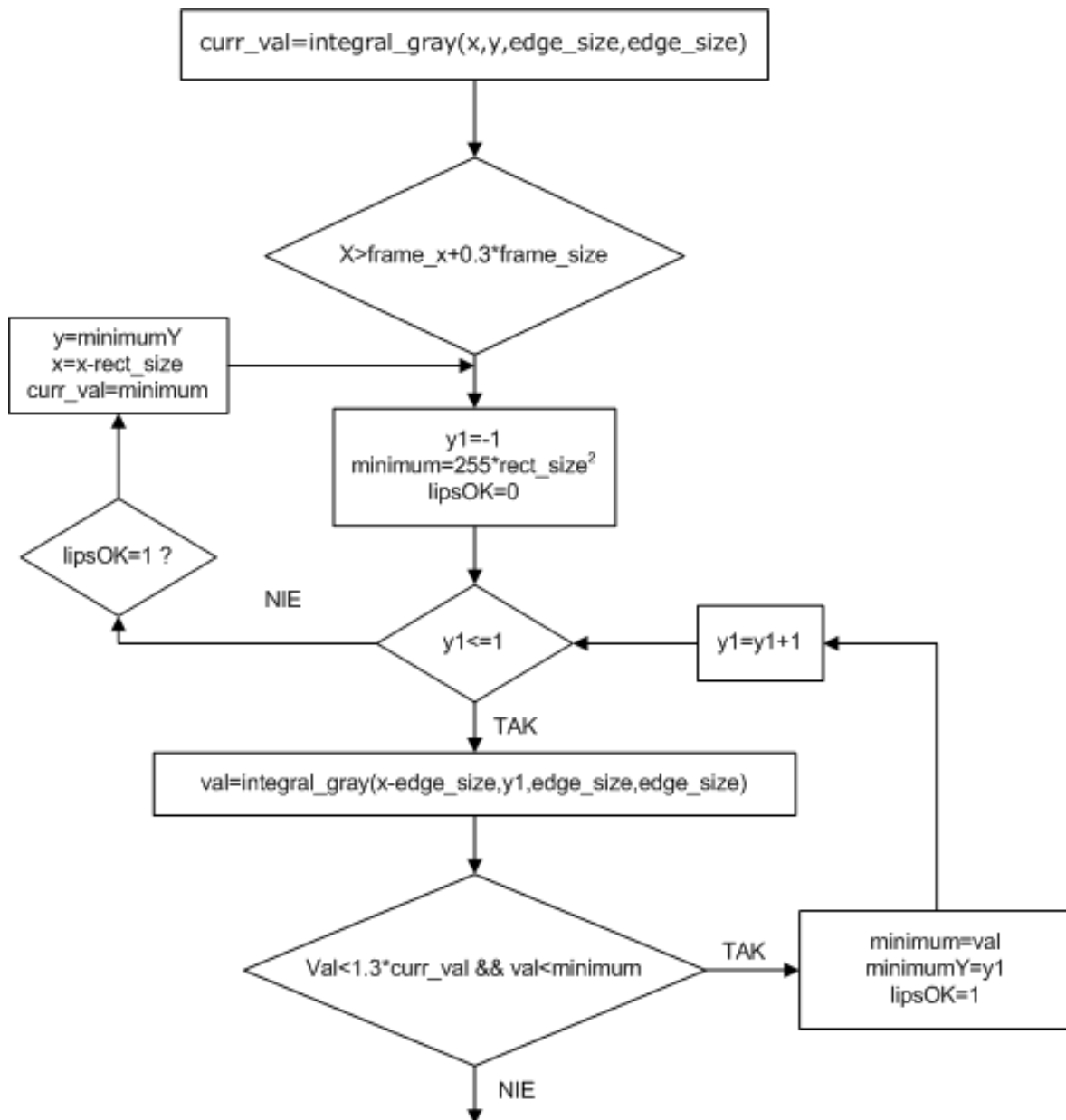
gdzie :

edge_size – wielkość boku kwadratu równa

$$\text{edge_size} = 0,02 * \text{frame_size} \quad (3.116)$$

Począwszy od pozycji startowej kolejno w kierunku lewym i prawym badany jest kształt ust, „podążając” w kierunku kwadratu o najmniejszej wartości na obrazie w skali szarości (najciemniejszym).

Schemat prezentuje szczęść algorytmu mającą na celu ustalenie lewego krańca ust.



Rysunek 3.32: Schemat algorytmu wykrywającego kształt ust.

Po lewej stronie od aktualnego położenia wyliczane są wartości dla kwadratów o współrzędnej y z przedziału $\{-1,0,1\}$. Jeżeli wartość wyliczonego kwadratu jest mniejsza od 1,3 wartości kwadratu w aktualnej pozycji oraz posiada najmniejszą wartość z kwadratów w danej pozycji współrzędnej x to następuje przesunięcie do nowej pozycji.

Na podstawie współrzędnych 3 punktów – środka ust oraz bocznych krańców ustalany jest ich kształt.

$$\begin{cases} \text{center_y} - \text{left_y} > 0,1 * (\text{center_x} - \text{left_x}) \\ \text{center_y} - \text{right_y} > 0,1 * (\text{right_x} - \text{center_x}) \end{cases} \quad \text{8} \quad (3.117)$$

Dla uśmiechu musi zostać spełniona para warunków:

gdzie :

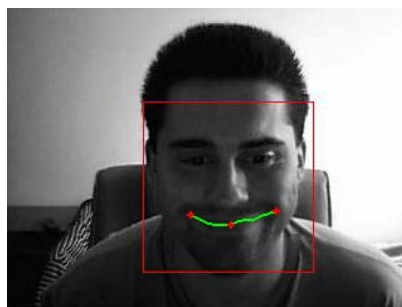
(center_x, center_y) – współrzędne środka ust

(left_x, left_y) – współrzędne lewej krawędzi ust

(right_x, right_y) – współrzędne prawej krawędzi ust

Dany warunek sprawdza czy różnica wysokości od punktów krańcowych i środka ust jest większa niż 10% szerokości ust w każdym z tych kierunków. Badane jest w ten sposób zaokrąglenie ust.

Na rysunku wykryty opisany algorytm kształt ust oznaczono kolorem zielonym. Punkty krańcowe oraz środkowy oznaczono czerwonymi krzyżykami.



Rysunek 3.33: Obraz w skali szarości z oznaczonym kształtem ust.

Na podstawie wyliczonych w 3.3.1 wartości prostokątów określany jest rodzaj uśmiechu. Jeżeli wartość prostokąta leżącego powyżej lub poniżej prostokąta ustalonego jako przestrzeń pomiędzy ustami spełnia warunek 3.118 to uśmiech zostaje uznany jako otwarty. Wykorzystano fakt, że widoczne podczas szerokiego, otwartego uśmiechu zęby są najjaśniejszym obszarem.

$$\text{rect_top} > 1,4 * \text{rect} \vee \text{rect_bottom} > 1,4 * \text{rect} \quad (3.118)$$

⁸ Forma wzoru dla obrazu, którego początek znajduje się w lewym górnym rogu

gdzie:

rect_top i rect_bottom – wartości odpowiednio górnego i dolnego prostokąta wyliczonego w 3.3.1

rect – wartość środkowego prostokąta ustalonego jako punkt centralny ust – wyliczony w 3.3.1

3.3.3. Lokalizacja oczu

W celu określenia stanu oczu konieczna jest znajomość ich dokładnego położenia. W tym celu wyliczona zostaje wartość prostokątów o wysokości 1 położonych od góry do dołu w obszarach o współrzędnych lewego górnego rogu

- dla lewej strony

$$x = \text{frame_size} + 0.1 * \text{frame_size} \quad (3.119)$$

- dla prawej strony

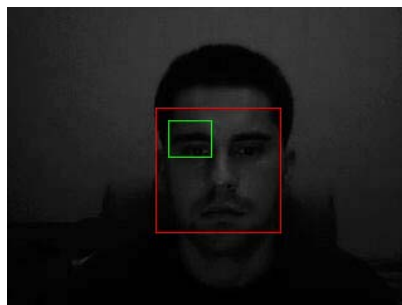
$$x = \text{frame_size} + 0.55 * \text{frame_size} \quad (3.120)$$

$$y = \text{frame_y} + 0.1 * \text{frame_size} \quad (3.121)$$

i wymiarach

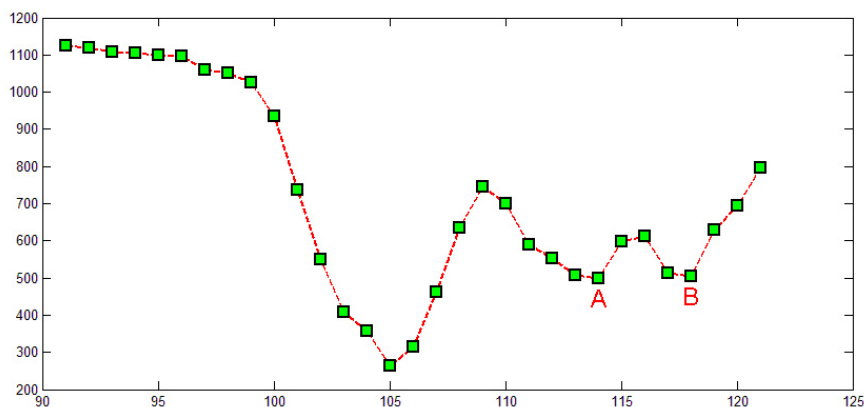
$$w = 0.45 * \text{frame_size} \quad (3.122)$$

$$h = 0.3 * \text{frame_size} \quad (3.123)$$



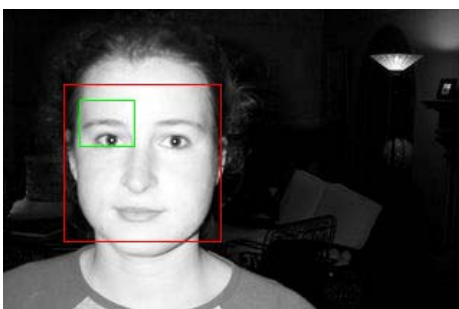
Rysunek 3.34: Zaznaczony obszar

Rysunek 3.35 prezentuje wartości wyliczonych prostokątów o wysokości 1 wzdłuż oznaczonego na zielono obszaru na rysunku 3.34. Wyraźnie zaznaczają się trzy minima. W badanym obszarze oko znajduje się w stanie otwartym. Oznaczone na rysunku 3.35 punktami A i B wartości odpowiadają dolnej i górnej krawędzi oka. Najmniejsza wartość na całej wysokości w danym przypadku wyznacza pozycję brwi.

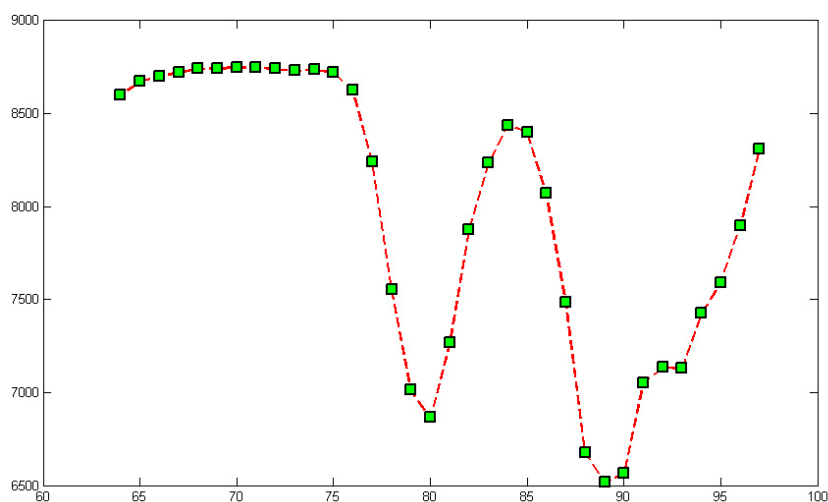


Rysunek 3.35: Wartości poprzecznych prostokątów lewego oka dla twarzy z rysunku 3.34

W przypadku jasno oświetlonego otoczenia lub osoby posiadającej mało zaznaczające się brwi (rysunek 3.36) analiza opisanego obszaru pokazuje, że najciemniejszym obszarem będzie okolica źrenicy.



Rysunek 3.36: Przykład obrazu o dużej jasności.



Rysunek 3.37: Wartości poprzecznych prostokątów lewego oka dla twarzy z rysunku 3.36

Niezmiennym faktem jest występowanie w danych obszarach dwóch lub więcej minimów lokalnych. Pierwsze napotkane od góry w badanym obszarze minimum zostaje uznane na położenie brwi. Kolejne napotkane minimum położone jest na górnej krawędzi źrenicy, co ustalono badając serie przykładowych obrazów.

Korzystając z faktu, iż okolice oka – źrenica oraz boczne krańce stanowią najciemniejsze obszary w swoim otoczeniu można ustalić położenie oczu w pozycji poziomej.

Rozmiary poszukiwanego obszaru zawierającego oko dobrane zostały empirycznie i wynoszą:

- szerokość

$$w = 0.25 * \text{frame_size} \quad (3.124)$$

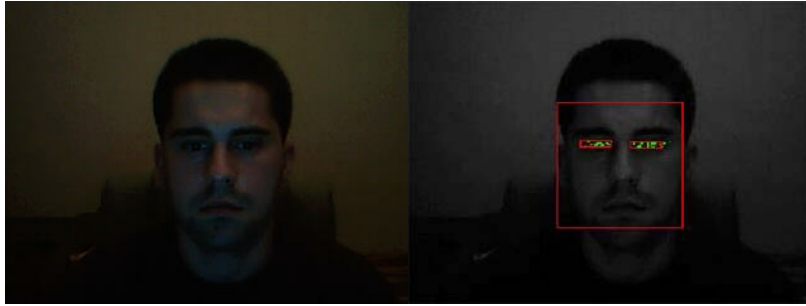
- wysokość

$$h = 0.06 * \text{frame_size} \quad (3.125)$$

Znając położenie górnych krawędzi oczu na całej szerokości obszaru o wysokości h , dla każdego punktu wyliczana jest wartość obszaru 3x3 piksele, którego środek stanowi badany piksel. Jeżeli wartość danego punktu na obrazie w skali szarości spełnia warunek opisany wzorem WZÓR to punkt zostaje oznaczony.

$$\text{gray}(x, y) < 0.9 * \frac{\sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} \text{gray}(i, j)}{3 * 3} \quad (3.126)$$

Po wyliczeniu danych wartości na szerokości całego badanego obszaru poszukiwany jest region zawierający oko o rozmiarach określonych wzorami (3.124) i (3.125), który zawiera największą ilość oznaczonych punktów. Rysunek 3.38a prezentuje badany obraz w formacie RGB. Na rysunku 3.38b zapisanym w skali szarości oznaczono kolorem zielonym punkty spełniające nierówność 3.128. Czerwonymi ramkami zaznaczono obszary zawierające największą ilość danych punktów.



Rysunek 3.38 a i b: Obraz RGB oraz w skali szarości z oznaczonymi oczami.

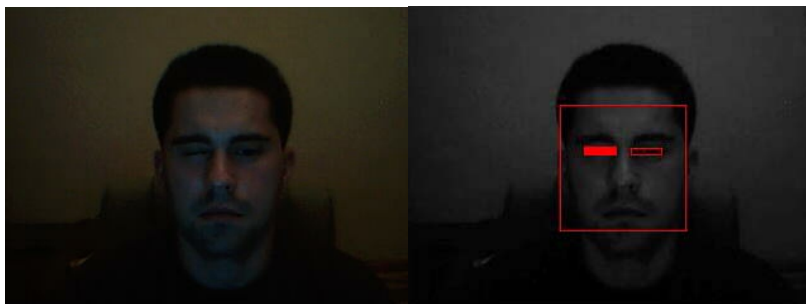
3.3.4. Określanie stanu oczu

Analogicznie do metod lokalizacji twarzy określenia stanu oczu opiera się na badaniu pewnych cech.

Wyliczane są wartości na obrazie w skali szarości dla wyznaczonych obszarów zawierających oczy oraz obszarów znajdującymi się nad i pod nimi. Wysokość przylegających do obszarów oczu regionów równa jest połowie wysokości obszaru zawierającego oko.

Jeżeli spełnione zostaną warunki opisane wzorem (3.127) oko uznane zostaje za zamknięte.

$$\begin{cases} 0.5 * \text{rect_center} < 0.7 * \text{rect_top} \\ 0.5 * \text{rect_center} < 0.7 * \text{rect_bottom} \end{cases} \quad (3.127)$$



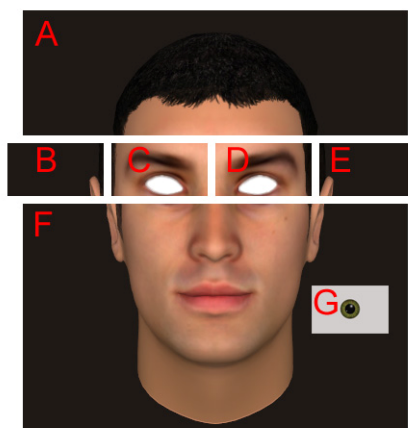
Rysunek 3.39 a i b: Obraz z oznaczonymi oczami i ich stanem.

3.4. Animacja twarzy

Ostatnim zagadnieniem związanym z daną pracą jest animacja trójwymiarowej twarzy. Istnieje wiele systemów i języków programowania modeli 3D. Wykonanie w danych systemach prostego przekształcenia kształtu obiektu wymaga dość dużej ilości obliczeń i przekształceń całej serii punktów, z których zbudowany jest model 3D. W danej pracy animację zrealizowano jako szybko zmieniane zdjęcia z serii ujęć twarzy o różnych „stopniach natężenia” wyrazu twarzy.

Animacja cyberagenta zrealizowana jest na serii szybko zmienianych obrazów (w empirycznie ustalonym okresie czasu) zawierających jego twarz w różnym położeniu. Obrazy z twarzą zostały wygenerowane przez program FaceGen Modeller w wersji 3.1.

W celu ograniczenia ilości klatek, twarz cyberagenta została podzielona na kilka części.



Rysunek 3.40: Sposób podziału obrazu z animowaną twarzą.

Elementami stałymi, są części obrazu A, B oraz E, oznaczone na rysunku 3.40. Literą G oznaczono bitmapę zawierającą obraz oka. Obrazy zapisywane są w formacie PNG, co daje możliwość wykorzystania przezroczystości. Bitmapa zawierająca oko (G) umieszczona za C i D widoczna jest tylko w przezroczystych regionach.

Przedstawione powyżej obrazy zawierają części twarzy w tzw. pozycji neutralnej.

W pracy wykonano animacje opisane poniżej.

3.4.1. Ruch oczu

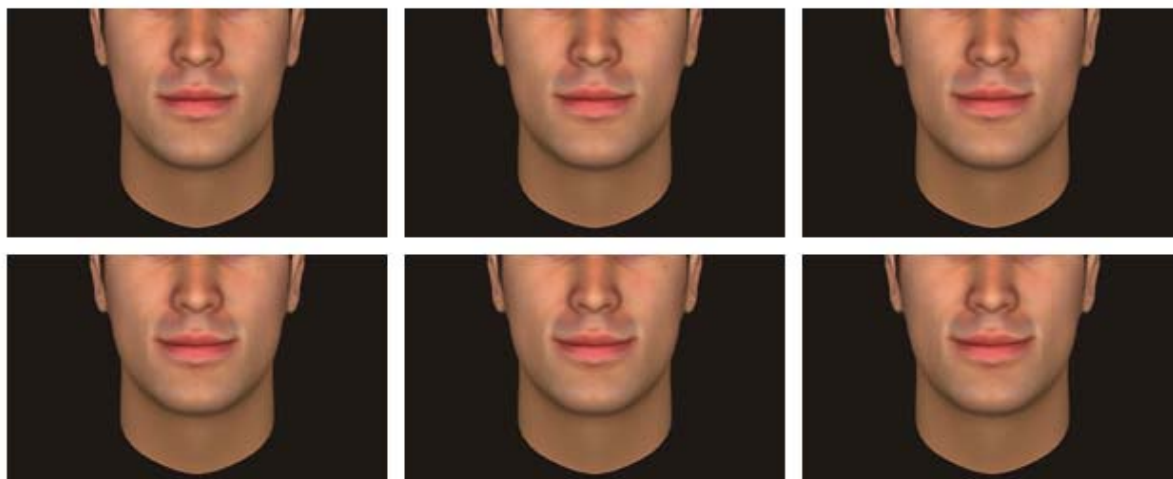
Ruch oczu odwzorowuje przemieszczanie się użytkownika komputera przed monitorem. Zrealizowany został poprzez przemieszczanie bitmapy zawierającej obraz oka, oznaczonej przez G na rysunku 3.40. Docelowa pozycja danej bitmapy określana jest w zależności od:

- pozycji oczu na obrazie z kamery internetowej
- pozycji aplikacji na ekranie monitora

Po ustaleniu nowej pozycji źrenicy, obraz przesuwany jest pomiędzy punktami znajdującymi się na linii od położenia aktualnego do docelowego, w odstępie czasu 20 ms.

3.4.2. Uśmiech zamknięty

Animacja zrealizowana została na 6 klatkach obrazu zaprezentowanych na rysunku 3.41.

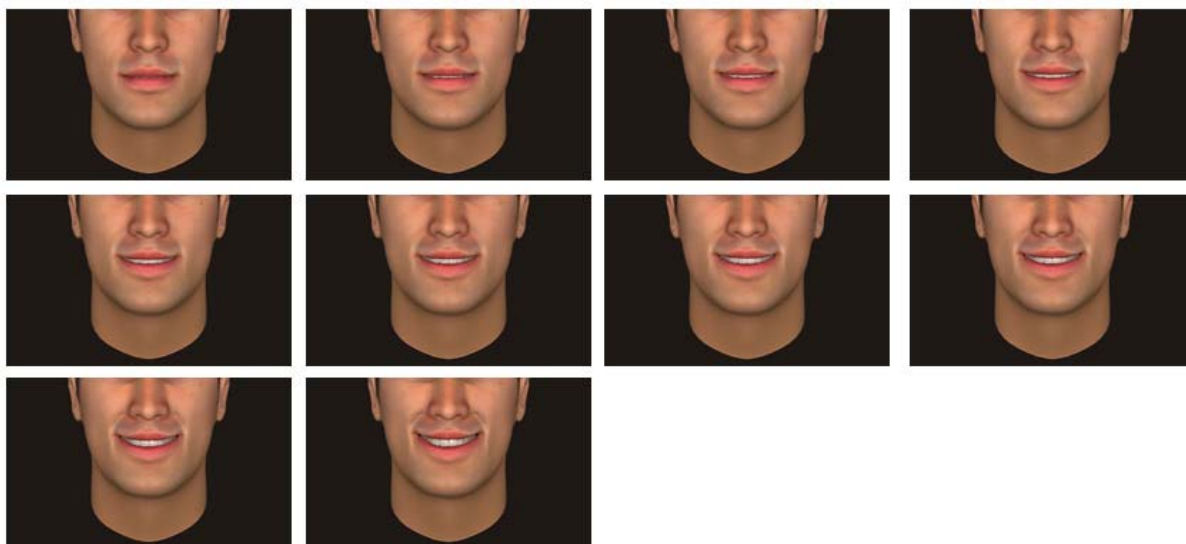


Rysunek 3.41: Seria klatek wykorzystywana w animacji uśmiechu.

Zwykły uśmiech zostaje animowany i utrzymywany, po zlokalizowaniu uśmiechu na twarzy użytkownika komputera. Kolejne klatki przerysowywane są w odstępie czasu równym 20 ms.

3.4.3. Uśmiech otwarty

Animacja zrealizowana została na 10 klatkach obrazu widocznych na rysunku 3.42 zmieniających w odstępie czasu równym 20 ms.

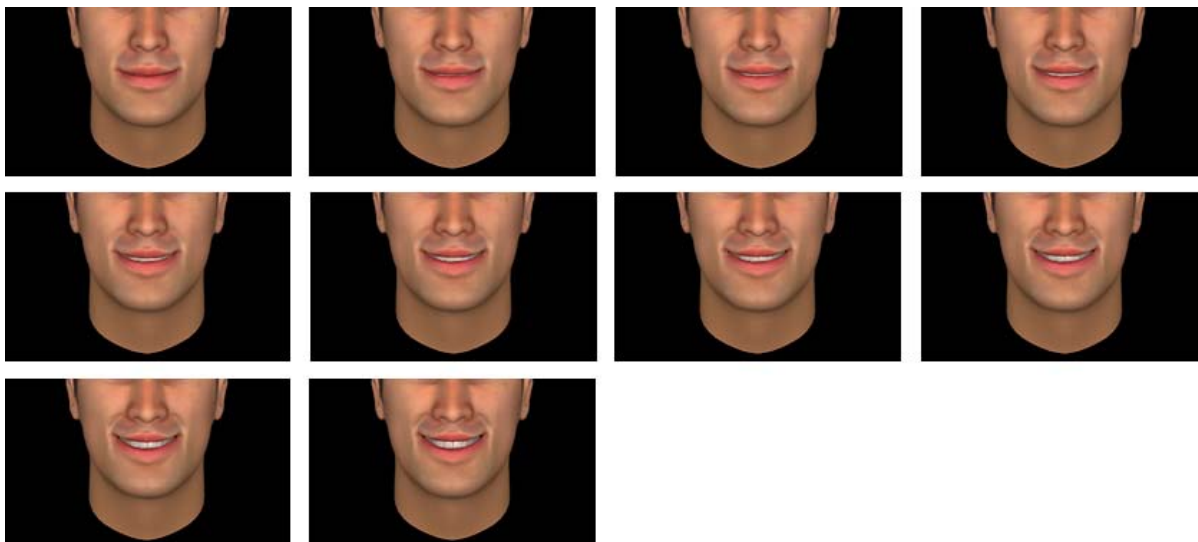


Rysunek 3.42: Seria klatek wykorzystywana w animacji otwartego uśmiechu.

Uśmiech otwarty zostaje animowany i zatrzymywany w momencie zarejestrowania uśmiechu otwartego na twarzy użytkownika komputera.

3.4.4. Animacja przejścia pomiędzy typami uśmiechu

Animacja przejścia pomiędzy uśmiechem zamkniętym, a otwartym zrealizowana została na 10 klatkach obrazu widocznych na rysunku 3.43 zmieniających w odstępie czasu równym 20 ms.



Rysunek 3.43: Seria klatek wykorzystywana w animacji przejścia pomiędzy uśmiechem zamkniętym a otwartym

3.4.5. Mrugnięcie zwykłe

W celu nadanie pewnych naturalnych odruchów ludzkich, w losowo wybranym czasie następuje zwykłe mrugnięcie obu oczu cyberbota. Animacja zrealizowana jest na 5 klatkach obrazu przedstawionych na rysunku 3.44 zmieniających w odstępie czasu równym 20 ms.



Rysunek 3.44: Seria klatek wykorzystywana w animacji mrugnięcia

3.4.6. Mrugnięcie specjalne

Mrugnięcie specjalne, tzw. „oczko” zrealizowane jest na 5 klatkach obrazu widocznych na rysunku 3.45 zmieniających w odstępie czasu równym 40 ms.



Rysunek 3.45: Seria klatek wykorzystywana w animacji mrugnięcia specjalnego

W przypadku specjalnego mrugnięcia dodatkowo zostaje opuszczona brew co symuluje tożsame ruchy człowieka w przypadku mrugania do drugiej osoby.

3.5. Aplikacja

Aplikacja zaimplementowana została jako applet Java. Dla potrzeb prezentacji zrealizowana została również strona internetowa widoczna na rysunku 3.46.



Rysunek 3.46: Widok aplikacji włączonej do strony internetowej

W celu uruchomienia programu konieczna jest instalacja JRE (Java Runtime Environment) w wersji min. 1.6 oraz JMF (Java Media Framework).

Po uruchomieniu system automatycznie wyszukuje dostępne kamery i uruchamia pierwszą z odnalezionych. Format obrazu zostaje automatycznie ustawiony na 320x240. W przypadku, gdy kamera internetowa nie obsługuje danego formatu zostaje wybrana rozdzielczość, której szerokość jest najbliższa wartości 320.

4. Testy opracowanego rozwiązania

4.1. Lokalizacja twarzy

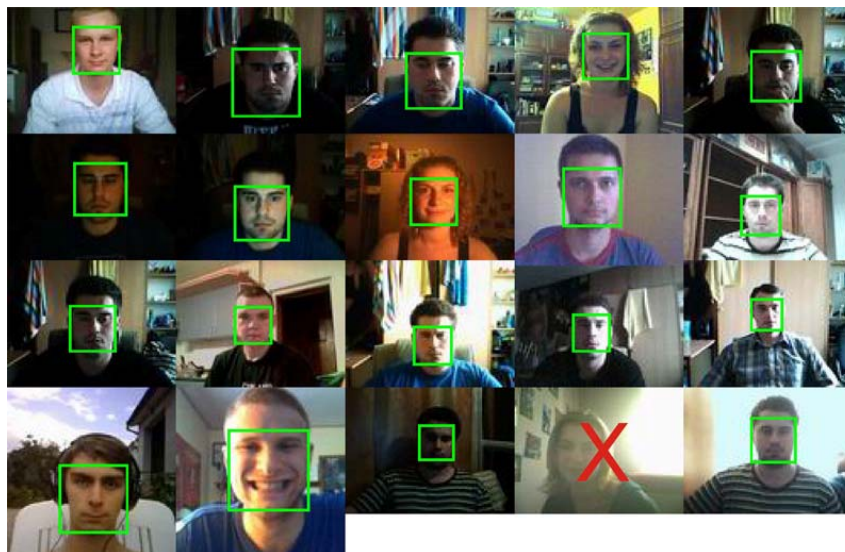
Skuteczność opracowanej metody lokalizacji twarzy została przetestowana dla:

- 78 dobrej jakości zdjęć zawierających twarze 26 różnych osób, pochodzą ze źródła internetowego [13]
- 20 zdjęć 8 osób z różnych kamer internetowych
- 40 zdjęć nie zawierających twarzy

Tabela 4.1: Wyniki przeprowadzonych testów lokalizacji twarzy

	Ilość	Wartość w %
Wykryte twarze (zdjęcia dobrej jakości)	76/78	97 %
Wykryte twarze (zdjęcia słabej jakości)	19/20	95%
Błędnie wykryte twarze	6/138	4 %

Przeprowadzone testy pokazują, że opracowany algorytm charakteryzuje się dobrą skutecznością klasyfikacji obrazów zawierających i nie zawierających twarzy. Metoda sprawdza się dobrze, poprawnie lokalizując twarze nawet dla obrazu wykonanego w pomieszczeniu o nierównym oświetleniu. Na rysunku 4.1 przedstawiono wykorzystane do testów obrazu z kamer internetowych. Kolorem zielonym oznaczono wykrytą twarz.

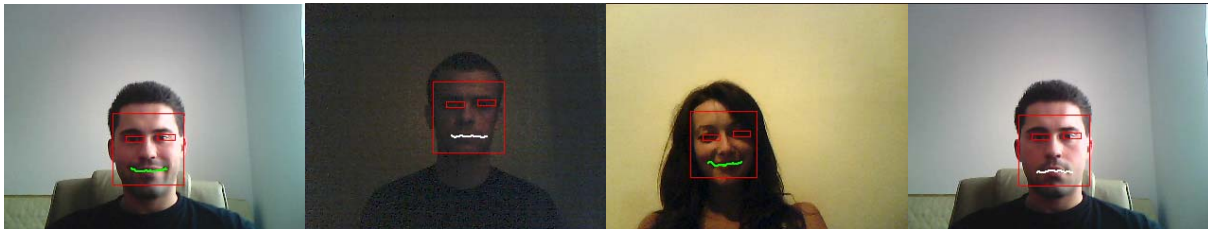


Rysunek 4.1: Obrazy z kamery internetowej wykorzystane w testach lokalizacji twarzy

Pomimo prób uniewrażliwienia skuteczności algorytmu na złe warunki oświetlenia, w przypadku zdjęcia z rysunku 4.1 oznaczonego czerwonym krzyżykiem twarz nie została odnaleziona.

4.2. Lokalizacja ust i oczu

Opracowana metoda lokalizacji ust oraz oczu charakteryzuje się dużą skutecznością. Na zdjęciach 4.2, 4.3, 4.4 oraz 4.5 zaprezentowano obrazy twarzy, na których poprawnie zlokalizowano oczy i usta.



Rysunek 4.2 a, b, c, d: Obrazy z oznaczonymi regionami oczu oraz kształtem ust.

4.3. Wykrywanie mimiki

Testy poprawności wykrywania określonych cech mimiki przeprowadzono w kilku pomieszczeniach z wykorzystaniem dwóch kamer internetowych. Dla wykazania poprawności wykrywanych stanów oczu oraz ust wprowadzono oznaczenia poprzez kolory.

W przypadku wykrycia zamkniętego oka, jego obszar zostanie zapełniony kolorem czerwonym.

Dla ust w przypadku rozpoznania określonego stanu, piksele wykryte jako ciemna przestrzeń pomiędzy wargami oznaczane są kolorem:

- stan neutralny – kolor biały
- uśmiech zamknięty – kolor zielony
- uśmiech otwarty – kolor czerwony

Na rysunku 4.3 zaprezentowano przykład, na którym użytkownik komputera uśmiecha się, co jak widać odwzorowane jest poprzez uśmiech cyberbota. Kolorem zielonym oznaczono usta, których kształt został poprawnie zakwalifikowany jako uśmiech.



Rysunek 4.3: Przykład działającej aplikacji.

Na rysunku 4.4 zaprezentowano twarz, na której zarejestrowano mrugnięcie, co odzwierciedlone jest poprzez mrugnięcie cyberbota.



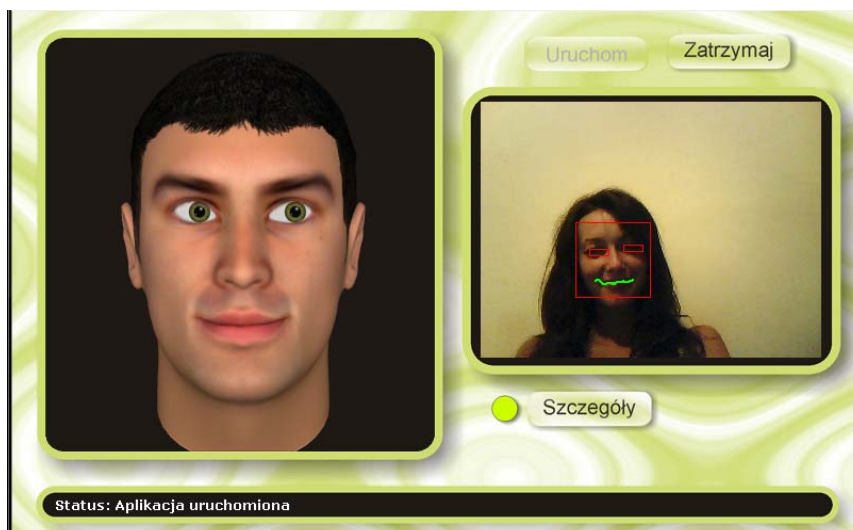
Rysunek 4.4: Przykład działającej aplikacji.

Dla twarzy na rysunku 4.5 zostało wykryte mrugnięcie przy jednoczesnym uśmiechu.



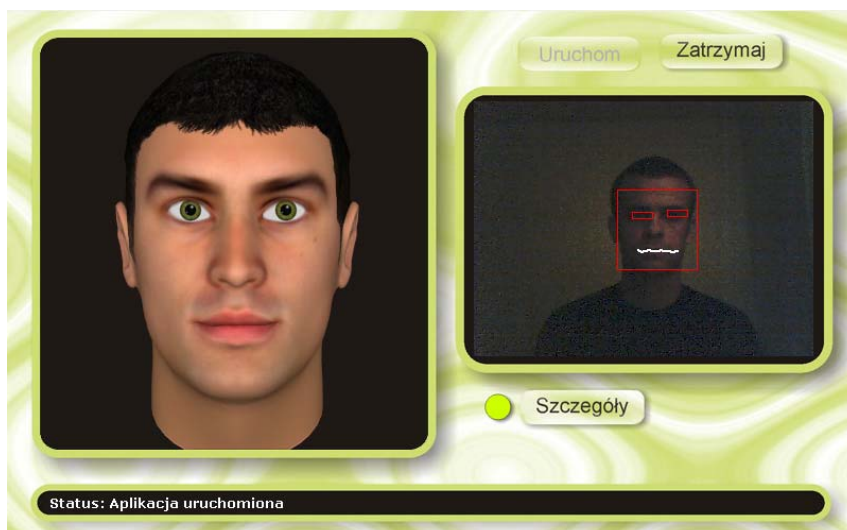
Rysunek 4.5: Przykład działającej aplikacji.

Rysunek 4.6 prezentuje kolejny przykład obrazu z twarzą, na której wykryto uśmiech



Rysunek 4.6: Przykład działającej aplikacji.

Testy przeprowadzane w ciemnym pomieszczeniu pokazały, że algorytmy wykrywania kształtu ust oraz położenia oczy są równie skuteczne, co można zaobserwować na rysunku 4.7 i 4.8.

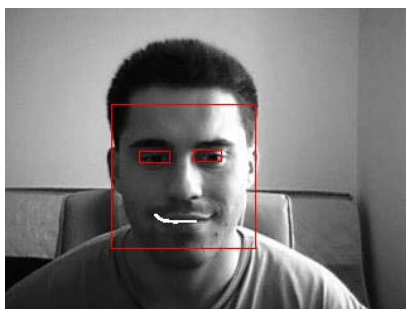


Rysunek 4.7: Przykład działającej aplikacji.



Rysunek 4.8: Przykład działającej aplikacji.

W całej serii testów przeprowadzonych dla trzech osób wystąpiły pewne błędy w sposobie oceny stanu mimiki. Do najczęstszych należy zła ocena kształtu ust, co spowodowane było nierównym oświetleniem części twarzy (rysunek 4.9). Na dużo jaśniejszej stronie twarzy kształt ust nie został poprawnie wykryty ze względu na wartość pikseli większą niżeli dopuszczalny próg różnicy wynoszący 30% (rysunek 3.32).



Rysunek 4.9: Przykład błędnie określonego kształtu ust.

5. Podsumowanie

Stale rozwijana technologia przynosi szybki rozwój mikroprocesorów, a tym samym mocy obliczeniowej maszyn i komputerów. Daje to coraz większe możliwości tworzenia zaawansowanych systemów. Pomimo istniejących już obecnie bardzo wielu metod biometrycznych, dążąc do uzyskania perfekcyjnych algorytmów nadal opracowywane są nowe. Większość z istniejących obecnie metod lokalizacji twarzy, rozpoznawania mimiki sprawdza się w specyficznych warunkach.

Jednym z celów danej pracy było opracowanie algorytmów działających poprawnie w systemie wykorzystującym jako źródło wejściowe obrazu kamerę internetową. Obrazy uzyskiwane z tradycyjnych, najczęściej używanych kamer internetowych są przeciętnej jakości. Przeanalizowane algorytmy lokalizacji twarzy oparte na kolorach, schematach i cechach charakteryzują się dobrą skutecznością w przypadku obrazów wysokiej jakości. Proste testy wykazały, że informacje, na których opiera się działanie wymienionych metod ulegają zatraceniu w przypadku obrazów z kamer internetowych. Algorytm oparty na kolorach nie sprawdza, jako że obrazy te często pozbawione są intensywności koloru, co prowadzi do błędnego wykrywania pikseli, na których nie ma widocznej skóry człowieka. Algorytmy oparte na schematach i cechach bazują na informacji z obrazów w skali szarości. Przy nierównym oświetleniu takie informacje zostają utracone. Podstawowe cechy (rysy twarzy) można łatwo wyłonić z obrazu binarnego o zmiennym progu binaryzacji, zależnego od wartości punktów sąsiednich (dział 3.1.2). Wykorzystanie tak utworzonego obrazu binarnego oraz obrazu w skali szarości do ustalenia cech, jakie powinien spełniać obszar obrazu zawierający twarz pozwoliło stworzyć wydajny i skuteczny algorytm lokalizacji twarzy. W opracowanej metodzie cechy dobrano empirycznie, po przeprowadzeniu wielu testów na obrazach zawierających twarz oraz obrazach bez twarzy. Cechy oraz wartości powiązane z poszczególnymi etapami algorytmu wybrano tak, aby wyeliminować możliwość błędnego określenia lokalizacji twarzy oraz skutecznie poprawne wykrywanie twarzy. Uzyskane rezultaty są zadowalające.

Metody lokalizacji części twarzy oraz ekstrakcji cech mimiki również zostały oparte o pewne cechy. Jak widać na przedstawionych testach, opracowane algorytmy wykrywania mimiki również charakteryzują się dużą skutecznością. Niestety w pewnych warunkach oświetleniowych skuteczność metody wykrywania kształtu ust ulega pogorszeniu.

Wysoka moc obliczeniowa współczesnych kart graficznych daje bardzo duże możliwości kreowania animacji trójwymiarowych o bardzo realistycznym wyglądzie. W danej pracy starano skupić się głównie na odwzorowaniu naturalnych odruchów towarzyszących rozmowie dwojga ludzi. Utrzymywanie stałego kontaktu wzrokowego jest wyrazem zainteresowania i świadczy o stanie skupionej uwagi na drugiej osobie. Animacja zrealizowana za pomocą szybko zmieniającej się serii zdjęć sprawdziła się w danym przypadku dobrze dając wrażenie płynnego ruchu cyberagenta. Rozwiązanie takie dało oczekiwane rezultaty bez stosowania skomplikowanych metod animacji modeli trójwymiarowych. Tym samym pozwoliło to ograniczyć zapotrzebowania na moc obliczeniową.

Aplikacja przetestowana przez kilka osób wzbudziła oczekiwane wrażenia. Stale utrzymujące się na użytkowniku oczy cyberagenta wzbudzają poczucie ciągłego zainteresowania. Odpowiedź cyberagenta na uśmiech wzbudza bardzo pozytywne emocje.

Systemy danego typu mogą doskonale sprawdzić się na stronach sklepów internetowych lub innych, gdzie bardzo ważne jest zaciekawienie potencjalnego klienta.

Dana praca jest jedynie prezentacją możliwości wykorzystania metod biometrycznych i prostych animacji w celu zainteresowania użytkowników stron www. Realizując rozbudowane aplikacje wykorzystujące większą ilość animacji wyrazów mimiki oraz połączenie takiej prezentacji graficznej z inteligentnymi systemami typu czatbot można uzyskać bardzo ciekawe oprogramowanie. Takie rozwiązanie z pewnością znajdzie wszechstronne zastosowanie w nowoczesnych inteligentnych systemach *Human-Machine Interface*.

Bibliografia

- [1] Kukharev Georgy, Kuźmiński Adam "Techniki Biometryczne. Część 1. Metody Rozpoznawania Twarzy" Wydział Informatyki, Politechnika Szczecińska, 2003
- [2] Alberto Albioly, Luis Torres, Charles A. Bouman, Edward J. Delp "A simple and efficient face detection algorithm for video database applications"
- [3] Yi Zhu and Florin Cutu, "Face Detection using Half-Face Templates", Computer Science Dept., Indiana University, USA
- [4] Strona o rozpoznawaniu twarzy <http://face-recognition.eu/>
- [5] Paul Viola, Michael Jones "Robust Real-time Object Detection", Vancouver, Canada, July 13, 2001
- [6] Constantine P. Papageorgiou, Michael Oren, Tomaso Poggio "A General Framework for Object Detection"
- [7] Yoav Freund, Robert E. Schapire "A decision-theoretic generalization of on-line learning and an application to boosting"
- [8] Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection"
- [9] <http://sourceforge.net/projects/opencvlibrary/>
- [10] Tomasz Kotlinski, „Automatyczne śledzenie twarzy i rozpoznawanie gestów na twarzy użytkownika komputera siedzącego przed kamerą"
- [11] Ya-Feng Deng, Guang-Da Su, Jun Zhou, Bo Fu , "Fast and robust face detection in video"
- [12] T. N. Bhaskar, Foo Tun Keat, Surendra Ranganath, Y. V. Venkatesh "Blink Detection and Eye Tracking for Eye Localization"
- [13] <http://www.robots.ox.ac.uk/~vgg/data/data-cats.html>