Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

AkademiaWydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

PRACA MAGISTERSKA

Szymon Deja

System rozpoznawania i aktywnego śledzenia oczu użytkownika komputera za pośrednictwem kamery w czasie rzeczywistym.

Promotor:
dr Adrian Horzyk

Kraków 2010

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

SZYMON DEJA

REAL-TIME SYSTEM FOR EYE DETECTION AND TRACKING OF COMPUTER USER USING WEBCAM

SUPERVISOR:
dr Adrian Horzyk

Krakow 2010

Chciałbym złożyć najserdeczniejsze podziękowania Panu dr Adrianowi Horzykowi za okazaną życzliwość oraz zrozumienie w trakcie tworzenia niniejszej pracy.

Spis treści

Spis rysunków

Rozdział 1

Wstęp

Eye tracking (po polsku zwany również okulografią) jest zbiorem technik badawczych pozwalających na uzyskanie informacji odnośnie ruchu oka, jego położenia w danym przedziale czasowym oraz (ewentualnym) punkcie fiksacji wzroku. Dane pozyskane w ten sposób mogą zostać wykorzystane np. podczas prowadzenia badań z zakresu użyteczności interfejsów, czytania tekstu, czy skuteczności przekazu reklamowego.

Każdego dnia nasz wzrok jest intensywnie wykorzystywany do wielu różnych celów: do czytania, do oglądania filmów rozrywkowych, do postrzegania i uczenia się nowych rzeczy. Jednak nie wszyscy zdajemy sobie sprawy, jak bardzo skomplikowanym procesem jest funkcjonowanie układu wzrokowego człowieka.

Okulografia, czyli śledzenie ruchów gałek ocznych (ang. eye tracking), jest techniką stosowaną od ponad 100 lat w takich dziedzinach jak psychologia, medycyna, interakcja człowiek-komputer, marketing i wielu innych.

1.1 Postawienie problemu

Celem pracy magisterskiej jest opracowanie i implementacja systemu śledzącego oczy użytkownika komputera klasy PC w czasie rzeczywistym za pomocą kamery internetowej. Docelowym przeznaczeniem systemu ma być estymacja punktu aktualnej koncentracji osoby siedzącej przed monitorem.

Najistotniejszym założeniem pracy postawionym przez autora jest opracowanie uniwersalnej aplikacji niewymagającej żadnej specjalnej konfiguracji sprzętowej do działania. Idealne rozwiązanie powinno składać się z komputera osobistego oraz pojedynczej uniwersalnej kamery internetowej.

Na rynku istnieje wiele komercyjnych systemów umożliwiających śledzenie wzroku. Największą wadą istniejących rozwiązań jest konieczność posiadania specjalnego sprzętu, zaczynając od dedykowanych urządzeń montowanych na głowie do kamer działających na podczerwień. Łatwo dostrzec

zalety systemu, który by nie wymagał specjalistycznego sprzętu. Umożliwiłoby to łatwe spopularyzowanie rozwiązania. Jednak stosowanie kamery internetowej sprawia, że dokładność śledzenia oczu będzie znacznie mniejsza w porównaniu z systemami komercyjnymi.

Kolejną istotną kwestią jest wymaganie, aby system był w stanie pracować w czasie rzeczywistym, co umożliwi śledzenie na bieżąco punkty koncentracji. Pociąga to za sobą konieczność doboru algorytmów o niezbyt dużej złożoności obliczeniowej. Duży nacisk został położony na oporność algorytmów przetwarzania obrazu na zmienne warunki oświetlenia.

Stworzenie systemu, który byłby w stanie sprostać postawionym wymagania, nie jest rzeczą łatwą. Świadczy o tym choćby fakt, że nie istnieje jeszcze żadne komercyjne rozwiązanie tego problemu.

Rozdział 2

Istniejące rozwiązania

Istnieje wiele metod umożliwiających rejestrację aktywności wzrokowej człowieka, począwszy od zwykłej bezpośredniej obserwacji poprzez inwazyjne metody mechaniczne, a skończywszy na badaniu różnicy potencjałów elektrycznych pomiędzy dwiema stronami gałki ocznej.

W dziedzinie śledzenia oczu zostało wykonanych wiele badań, przez co istnieje wiele rozwiązań znacząco różniących się od siebie. Wybór optymalnej metody zależy od celu, w jakim są wykonywane badania. Przeznaczenie danego systemu określa wymaganą dokładność, rozdzielczość, częstotliwość pomiarów, łatwość i wygodę używania, a także cenę.

Systemy do eye-trackingu można podzielić na kilka grup: ze względu na położenia urządzenia względem głowy (mobilne i niemobilne), rodzaju uzyskiwanych danych, metody wyznaczania punktu fiksacji (lub samego ruchu oka).

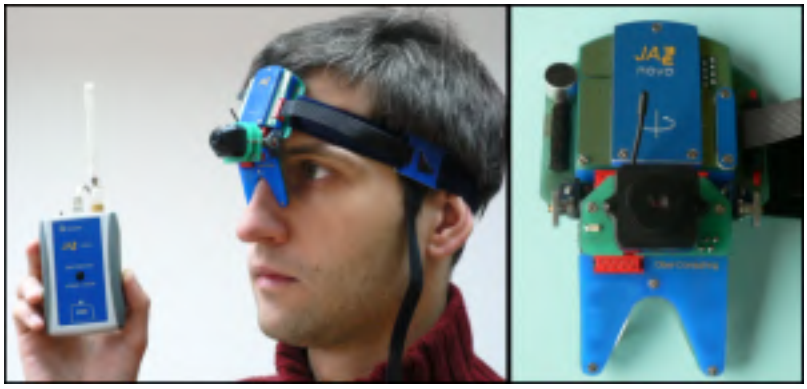
Poniżej został przedstawiony przegląd stosowanych metod do pomiaru pozycji gałki ocznej.

2.1 Rodzaj uzyskiwanych danych

Bezwzględny punkt odniesienia

Niektórzy badacze zajmujący się eye trackingiem są zainteresowani wyłącznie informacją o ruchu oka, ale nie w odniesieniu do punktu padania fiksacji w przestrzeni (punkt fiksacji jest miejscem w polu widzenia osoby badanej, na który patrzy się ona w danym momencie). Dla tego rodzaju badań najważniejsza jest informacja o położeniu oka w odniesieniu do punktu startowego. Takie dane mają charakter wartości bezwzględnych (ang. "absolute values"). Technika znana jest również pod nazwą eye trackingiu oczodołowego (ang. "orbital eye tracking").

Przykładem wyżej wymienionego urządzenia jest polski wynalazek JAZZ-novo rys. 2.1, powstały przy współpracy Instytutu Biocybernetyki i Inżynierii Biomedycznej PAN oraz firmy Ober



Rysunek 2.1: System pomiarowy JAZZ-novo

Consulting Poland.

Informacje zebrane tą metodą są często wykorzystywane w badaniach narządu przedsionkowego (ang. "vestibular research"), neurologicznych badaniach nad wzrokiem oraz w diagnozowaniu dysleksji. Praktyczne zastosowanie znajdują również w lotnictwie wojskowym, dostarczając informacji o interakcji pilota z kokpitem samolotu.

Względny punkt odniesienia

Znacznie większą gamę zastosowań mają systemy zwracające dane o względnym położeniu oka, czyli w odniesieniu do punktu padania fiksacji na monitorze bądź w przestrzeni. Wyniki testów są zazwyczaj prezentowane w postaci map ciepłych miejsc (ang. "heat maps") rys. 2.2.

Jest to możliwe dzięki zastosowaniu zaawansowanych algorytmów, które są w stanie monitorować położenie oka, łącząc następnie punkt fiksacji osoby badanej z daną sceną wzrokową. Wszystkie środowiska dedykowane owemu zadaniu (ang. "gaze tracking systems") muszą zostać wcześniej skalibrowane. Liderem na rynku światowym dostarczającym takie rozwiązanie jest firma Tobii.

Trójwymiarowy kierunek wzroku

Trzecim rodzajem eye trackingu jest tak zwana "okulografia 3D". Oprócz danych o stopniu horyzontalnego i wertykalnego odchylenia oka dostarcza informacji o stopniu rotacji (skrętu) gałki ocznej wokół własnej osi. Niektóre urządzenia pozwalają nawet na monitorowanie ruchów oraz położenia głowy, co można później zestawić z danymi o rotacji oka. Rozwiązania tego rodzaju dostarcza niemiecka firma SensoMotoric Instruments GmbH. Jest nim nagłowne urządzenie 3D Video-Oculography® (3D VOG) rys. 2.3.

Rysunek 2.2: Wyniki badania wzroku reprezentowane przez mapę ciepła

Rysunek 2.3: System 3D Video-Oculography®

Rysunek 2.4: Urządzenia służące do pomiaru potencjału skóry [2]

Uzyskane dane są najczęściej wykorzystywane do badań nad widzeniem obuocznym (ang. "binocular vision research") oraz badaniach narządu przedsionkowego.

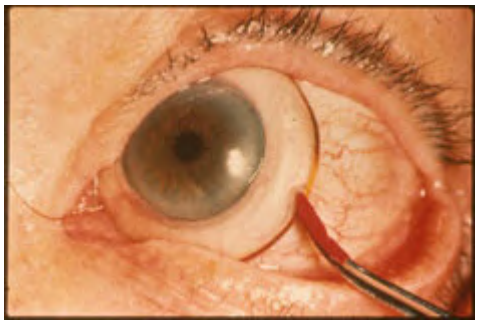
2.2 Metody pomiaru pozycji oka

2.2.1 Elektro-okulografia (EOG)

Technika polega na pomiarze różnicy potencjału skóry, za pomocą elektrod zamocowanych wokół oczu. Położenie oka można wykryć z dużą precyzją dzięki rejestrowaniu bardzo niewielkich różnic w potencjale skóry. Technika ta jest jednak dość kłopotliwa i nie nadaje się do codziennego użytku, ponieważ wymaga bliskiego kontaktu elektrod ze skórą użytkownika, jak na rys. 2.4. Metoda ta była najczęściej stosowanym rozwiązaniem 40 lat temu, ale nadal znajduje zastosowanie[3].

Metoda opiera się na pomiarze różnicy w potencjałach bioelektrycznych mięśni położonych w okolicy ocznej. Na podstawie amplitudy sygnału oblicza się odległość pomiędzy szybkimi zmianami położenia oka (zwanymi również sakadami). Jest to możliwe dzięki temu, iż potencjał z przodu gałki ocznej różni się od potencjału jej tylnej części. Systemy bazujące na tej metodzie są bardzo podatne na szum wywołany aktywnością innych mięśni twarzy. Elektrookulografię wykorzystuje się najczęściej w medycynie.

Zmiany zachodzące w ładunku pól elektrycznych wywołane ruchem oka pozwalają monitorować jego położenie.



Rysunek 2.5: Nieinwazyjna metoda umieszczenia cewki w oku

2.2.2 Technik wykorzystujące soczewki kontaktowe

Dzięki zastosowaniu specjalnych soczewek kontaktowych możliwe jest dokładne ocenienie ruchu gałki ocznej. Soczewki takie zawierają małe cewki indukcyjne. Dokładne ustawienie soczewek jest możliwe poprzez rejestrowanie zmiany pola elektro-magnetycznego wywołanego ruchem oka. Jednym z głównych problemów jest kompensacja ruchów głowy. Konieczne jest stosowanie urządzeń mocowanych na głowie. Wadami są ograniczenia w ruchu i nieporęczne urządzenia, przez co zastosowanie tej metody ograniczone jest do eksperymentów laboratoryjnych.

2.2.3 Techniki optyczne bazujące na rejestracji obrazu video

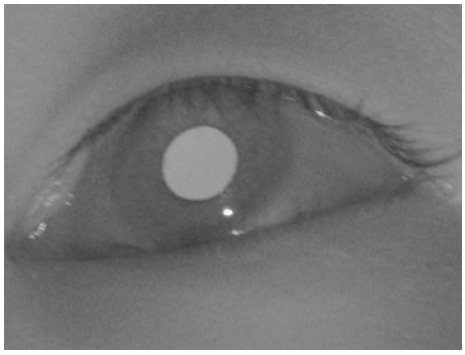
2.2.3.1 Z wykorzystaniem światła podczerwonego

Oświetlenie podczerwone osi optycznej oka zdecydowanie ułatwia lokalizacje tęczówki [4]. Źrenica odbija prawie całe światło, co sprawia, że kamera rejestruje ją jako biały okrąg rys. 2.6. Jest to zjawisko analogiczne to efektu czerwonych oczu. Źródło światła podczerwonego znajdującego się poza osią optyczną oka sprawia, że kamera rejestruje źrenicę jako ciemny obszar rys. 2.6

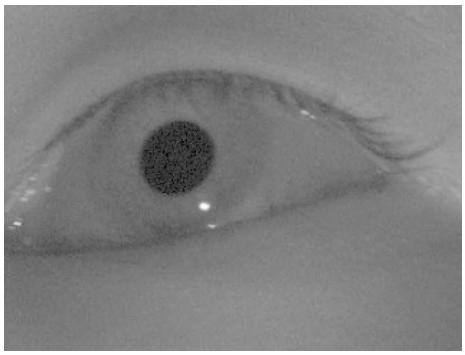
Metoda polega na określeniu położenia między środkiem źrenicy a odbiciem źródła światła podczerwonego od rogówki. Refleksy światła odbite od różnych struktur gałki ocznej nazywane są obrazami Purkinjego (ang. “Purkinje imaging“) rys. 2.7.

Eye trackery tego rodzaju wyznaczają pozycję oka rejestrując pozycję odbicia światła od powierzchni rogówki (tzw. pierwszego obrazu Purkinjego – P1 – zwanego również glintem) względem środka źrenicy. W celu zwiększenia dokładności pomiaru możliwe jest zwiększenie liczby rejestrowanych przez urządzenie punktów (glintów) do czterech.

Istnieją również urządzenia o większej precyzji pomiaru zwane podwójnymi trackerami Purkinjego (ang. “dual-Purkinje eye trackers“) wykorzystujące odbicia światła od powierzchni rogówki

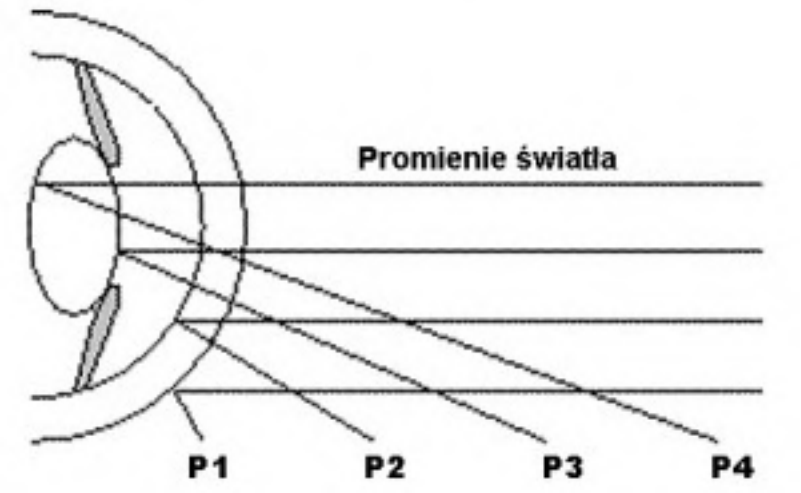


(a) Efekt jasnej źrenicy wywołany źródłem światła podczerwonego znajdującego się na osi oka



(b) Efekt ciemnej źrenicy wywołany źródłem światła podczerwonego znajdującego się poza osią oka

Rysunek 2.6: Oświetlenie oka światłem podczerwonym[5]



Rysunek 2.7: Cztery rodzaje obrazów Purkinjego

(czyli pierwszego obrazu Purkinjego, P1) w odniesieniu do tylnej powierzchni soczewki (zwanego czwartym obrazem Purkinjego, P4). Takie systemy cechują się znaczną dokładnością, jednakże ich największą wadą jest fakt, że czwarty obraz Purkinjego jest bardzo słaby, w wyniku czego badana osoba nie może poruszać głową podczas przeprowadzania eksperymentu. Efekt ten osiąga się zazwyczaj dzięki zastosowaniu podpórki pod brodę lub specjalnego gryzaka (ang. "bite-bar").

2.2.3.2 Bez wykorzystania światła podczerwonego

Obraz rejestrowany jest za pomocą standardowej kamery. Działanie tej grupy systemów polega na pomiarze kształtu oraz pozycji rogówki oka względem wybranej części twarzy (np. rogi oczu). Bardzo często zdarza się, że część rogówki zostaje przesłonięta przez powiekę, co może powodować zniekształcenie danych o wertykalnym położeniu oka. Rozwiązaniem jest śledzenie źrenicy zamiast rogówki, co sprawia, że obszar mogący zostać zakryty przez powiekę się zmniejsza. Jednak kontrast między źrenicą a pozostałą częścią oka jest znacznie mniejszy niż w przypadku rogówki.

Odmienne podejście polega na stosowaniu sieci neuronowej. Zbiorem uczącym jest sekwencja obrazów przedstawiających oko oraz punkt na monitorze, na jaki w danym momencie spogląda dana osoba. Proces kalibracji takich systemów jest bardzo czasochłonny. Wymagana jest duża ilość obrazów wejściowych do prawidłowego nauczenia sieci neuronowej [21] .

Systemy oparte na wykorzystaniu standardowej kamery wideo zakładają przeważnie, że użytkownik ma zupełnie nieruchomą głowę podczas badania.

Aby umożliwić swobodny ruch głowy konieczna jest implementacja metod precyzyjnie określających aktualną pozycje głowy (przesunięcia oraz skręty).

2.3 Przegląd istniejących rozwiązań

Na rynku istnieje cała gama komercyjnych systemów do eye-trackingu. Wiodącymi produktami są Tobii [25] oraz SMI [26]. Są to systemy zdalne, czyli nie jest wymagane montowanie żadnych urządzeń na głowie użytkownika. Swoje działanie opierają na zastosowaniu kamery rejestrującej obrazy w paśmie podczerwieni oraz źródeł światła podczerwonego. Aktualny punkt, na który patrzy się użytkownik, wyznaczany jest badając zmianę pozycji między środkiem źrenicy a refleksami odbitymi od tęczówki oka wywołanymi emiterem światła podczerwonego.

Nie istnieje jednak komercyjny system, którego działanie opierałoby się na zastosowaniu standardowej kamery. Istnieje jednak duża ilość ośrodków badawczych, w których są prowadzone intensywne badania w celu stworzenia prototypu takiego systemu. Przykładem może być organizacja COGAIN [27], która między innymi pracuje nad stworzeniem taniego systemu do eye-trackingu z wykorzystaniem standardowej kamery.

Obecnie jedynym dostępnym projektem zajmującym się badaniem wzroku za pomocą standardowej kamery jest opengazer [22]. Jest to program typu open source. Jego główna wadą jest założenie, że głowa użytkownika jest zupełnie nieruchoma podczas działania programu. Nawet niewielki ruch sprawia, że konieczna jest ponowna kalibracja systemu.

Rozdział 3

Podstawy teoretyczne

3.1 Układ wzrokowy

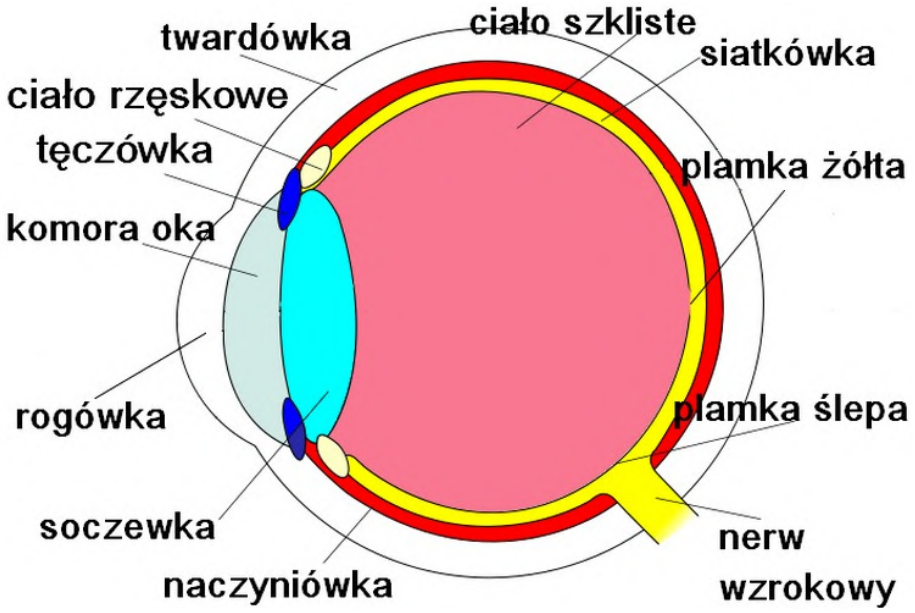
Do budowy systemu śledzącego wzrok konieczna jest znajomość podstawowych zagadnień związanych z systemem wzrokowym człowieka. Wiedza ta jest konieczna do zrozumienia sposobu wyznaczania aktualnego kierunku wzroku, jak również do identyfikacji istniejących ograniczeń, jakie muszą zostać uwzględnione podczas tworzenia systemu śledzenia wzroku.

3.1.1 Opis układu wzrokowego

Oko ludzkie zbudowane jest z soczewki ze zmienną i regulowaną ogniskową, tęczówki (przesłony) regulującej średnicę otworu (źrenicy), przez którą wpada światło, oraz światłoczułej siatkówki w głębi oka. Fotoreceptory znajdujące się na siatkówce zmieniają światło na potencjał elektryczny, który wywołuje impuls nerwowy transportowany do kory mózgowej przez nerw wzrokowy.

Siatkówka składała się z dwóch rodzajów fotoreceptorów: czopków oraz pręcików. Pręciki są bardzo wrażliwe na intensywność światła, sprawia to, że jest możliwe widzenie w warunkach otoczenia o bardzo małym natężeniu światła. Jednak te receptory nie są w stanie rejestrować barwy, co sprawia, że wszystko widziane nocą ma odcień szarości. W przeciwieństwie do pręcików czopki potrzebują znacznie jaśniejszego światła do produkcji impulsów nerwowych, ale za to są w stanie rozpoznawać kolory. Pręciki mają trzy rodzaje filtrów czułych na różne długości światła: czerwony, zielony oraz niebieski.

Dzięki istnieniu dużego kontrastu między twardówką a tęczówką możliwa jest ocena kierunku wzroku. Jednym z powodów takiej właśnie budowy oka jest fakt, iż człowiek jest istotą społeczną, a określenia kierunku wzroku jest bardzo pomocne podczas komunikacji.



Rysunek 3.1: Budowa oka

Za ostrość obrazu odpowiedzialna jest plamka żółta. Jest to największe skupisko czopków o średnicy około dwóch minimetrów znajdujące się w centrum siatkówki. Na pozostałej część siatkówki mieści się od 15 od 50 procent fotoreceptorów. Sprawia to, że szerokość widzianego obrazu ma około jednego stopnia. Nie jest możliwa koncentracja wzroku na obiekcie mniejszym niż plamka żółta, przez co nie jest możliwe wyznaczenie kierunku wzroku z większą dokładnością niż jeden stopień.

3.1.2 Ruch oka

Ruchy oczu służą dwóm podstawowym celom. Po pierwsze stabilizacji obrazu na siatkówce, w celu skompensowania ruchów głowy lub ruchów przedmiotów w polu widzenia. Po drugie ustawieniu oka względem otoczenia tak, by analizowany w danej chwili fragment obrazu był rzutowany na część środkową siatkówki o największej czułości i zdolności przetwarzania szczegółów. Ze względu na zróżnicowanie funkcji siatkówki, przetwarzanie szczegółów obrazu (np. kształtów liter podczas czytania) możliwe jest tylko na niewielkim obszarze około jednego stopnia kątowego w pobliżu środka siatkówki.

Podczas normalnej aktywności człowieka, przez większość czasu oczy pozostają w stanie fiksacji, tzn. w stanie względnego spoczynku. Podczas fiksacji zachodzi pobieranie informacji wzrokowej z otoczenia. Czas trwania fiksacji jest zależny od sposobu przetwarzania informacji. Na ogół waha

się w granicach od ok. 0,15 sek. do 1,5 sek. Średnio 4-6 razy na sekundę wykonywana jest sakada, czyli bardzo szybki, skokowy ruch zmiany położenia pomiędzy kolejnymi fiksacjami. Sakada trwa na ogół ok. 0,03 do 0,06 sek.

3.2 Przepływ optyczny

Przepływ optyczny (ang. opticalflow) to pole wektorowe, które umożliwia przekształcenie danego obrazu w sekwencji w kolejny obraz tej sekwencji, poprzez przemieszczenie obszarów z pierwszego obrazu (dla których zostało określone to pole), zgodnie z odpowiadającymi im wektorami tego pola na drugi obraz. Mówiąc w skrócie, przepływ optyczny jest zbiorem translacji (w postaci pola), które przekształcają dany obraz w sekwencji w następny obraz w sekwencji [3].

Istnieje wiele metod wyznaczania przepływu optycznego. Można je podzielić na trzy główne grupy:

- metody gradientowe bazujące na analizie pochodnych (przestrzennych i czasowych) intensywności obrazu,
- metody w dziedzinie częstotliwości oparte na filtrowaniu informacji obrazowej w dziedzinie częstotliwości,
- metody korelacyjne bazujące na odpowiedniości obszarów obrazów.

Algorytm Lucas-Kanade W pracy został wykorzystany algorytm Lucas-Kalman. Jest to metoda gradientowa.

Działanie algorytmu opiera się na trzech podstawowych założeniach:

1. Jasność obrazu nie ulega dużym zmianą między kolejnymi klatkami sekwencji.
2. Prędkość ruchu obiektów na obrazie jest niewielka.
3. Punkty znajdujące się w niewielkiej odległości od siebie poruszają się podobnie.

Jasność obrazu jest określona funkcją zależącą od czasu.

$$f(x, y, t) \equiv I(x(t), y(t), t)$$

Wymaganie, aby jasność obrazu nie zmieniała się znacząco w czasie przedstawia równanie:

$$I(x(t), y(y), t) = I(x(t + dt), y(t + dt), t + dt)$$

Oznacza to, że intensywność śledzonego piksela nie zmienia się w czasie:

$$\frac{\partial f(x,y)}{\partial t} = 0$$

Wykorzystując to założenie można zapisać warunek przepływu optycznego oznaczając przez u wektor prędkości w kierunku x, natomiast przez v wektor prędkości w kierunku y

$$-\frac{\partial I}{\partial t} = -\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Niestety jest to równanie z dwiema niewiadomymi. Aby je rozwiązać potrzebne są dodatkowe informacje. Właśnie w tym momencie wykorzystywane jest trzecie założenie: punkty znajdujące się w niewielkiej odległości od siebie poruszają się podobnie. Dzięki temu to rozwiązania równania dla jednego punktu wykorzystuje się dodatkowe piksele go otaczające. Do rozwiązania układu równań stosuje się metodę najmniejszych kwadratów.

Założenia tej metody sprawiają, że jest ona w stanie wykryć tylko bardzo nieduży ruch między kolejnymi klatkami. Aby poprawić jej działanie stosuje się piramidę obrazów. Polega to na wykonaniu algorytmu kolejno na obrazie o zmniejszanej rozdzielczości.

Rozdział 4

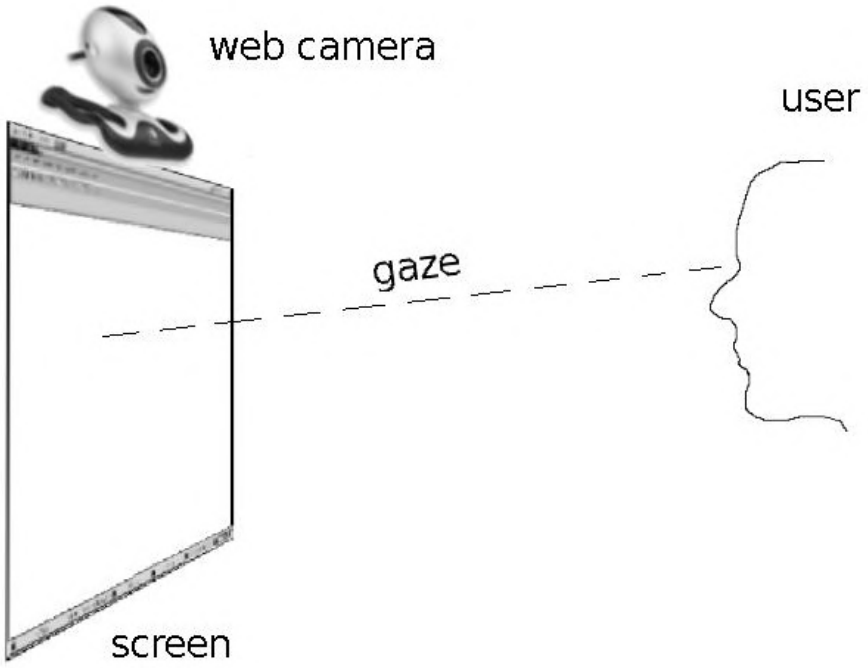
Opis własnego rozwiązania

Prezentowany systemem do śledzenia wzroku opiera działanie na rejestracji obrazu oka za pomocą standardowej kamery internetowej. Schemat systemu został przedstawiony na rys. 4.1. Dla uproszczenia dalszych rozważań zakłada się, że kamera jest umieszczona na monitorze na jego środku. Większość komercyjnych systemów wymaga, aby kamera była umieszczona poniżej monitora, daje to nieco lepszy obraz oka. Jednak celem tej pracy było stworzenie uniwersalnego rozwiązania. Standardowo kamera internetowa jest ulokowana nad monitorem, szczególnym przykładem mogą być laptopy z wbudowaną kamerą, gdzie nie istnieje możliwość zmiany jej lokacji.

Celem pracy jest wyznaczenie punktu na monitorze, na jaki w danym momencie patrzy użytkownik. Aby uzyskać takie dane konieczne jest określenie względnej pozycji oka, jak i głowy.

Pierwszym etapem jest inicjalizacja algorytmu, podczas której zostaje stworzony model głowy. Po inicjalizacji następuje faza śledzenia. Do określenia pozycji 3D głowy wykorzystywany jest przepływ optyczny między kolejnymi klatkami pozyskiwanymi z kamery oraz algorytm POSIT [13]. Zmiana kierunku wzroku jest wyznaczana badając różnice odległości między środkiem gałki ocznej a środkiem źrenicy oka. Dane te są otrzymywane przy użyciu całego szeregu metod do przetwarzania obrazów opisanych w dalszej części pracy. Do uzyskania prezentowanych rezultatów zostało przetestowana cała gama algorytmów, których część została zaprezentowana w dalszej części pracy.

Następnie po wyznaczeniu względnej pozycji oczu i głowy wykorzystując model geometryczny następuje mapowanie wzroku na płaszczyznę monitora. Aby wyznaczyć niezbędne współczynniki mapowania konieczny jest proces kalibracji na początku sesji działania programu. Kalibracja systemu polega na podążaniu wzrokiem za punktem wyświetlanym w określonych lokacjach na monitorze.



Rysunek 4.1: Schemat systemu

Rozdział 5

Śledzenie pozycji głowy

Duża ilość dostępnych systemów do śledzenia wzroku zakłada ograniczony ruch głowy. Nie jest to rozwiązanie wygodne ani praktyczne. Człowiek nie jest w stanie utrzymać głowy w zupełnym bezruchu przez długi okres czasu bez zastosowania podstawki, na której by można oprzeć brodę lub czoło. Takie urządzenia stosowane są w okulistyce podczas badania wzroku. Na mimowolny ruch głowy ma duży wpływ oddychanie. Aby możliwe było wyznaczenie kierunku, w którym jest skierowany wzrok osoby siedzącej przed komputerem bez nakładania restrykcyjnego ograniczenia ruchu główny, konieczna jest bardzo precyzyjne śledzenie pozycji 3D twarzy.

Wyznaczanie pozycji głowy jest bardzo ważną dziedziną badań nad interakcją człowiek-komputer (HCI). Istnieje wiele metod estymacji pozycji za pomocą pojedynczej kamery. Metody te można podzielić na dwie główne grupy: bazujące na modelu głowy oraz na cechach charakterystycznych twarzy. Metody używające modelu estymują pozycję wyznaczając zależność 2-3D między cechami. Za pomocą tych zależności wyznaczana jest pozycja.

Metody oparte na własnościach twarzy zakładają, że istnieje pewna relacja między pozycją 3D a pewnymi własnościami obrazu twarzy. Wyznaczają one te zależności przy użyciu dużej ilości obrazów trenujących ze znaną pozycją twarzy do wytrenowania sieci neuronowej. Wadą tego podejścia jest nieprawidłowe działanie dla nowych twarzy, których obraz nie został użyty podczas treningu.

W pracy zostaną zaprezentowane dwa odmienne podejścia. Pierwsze zainspirowane pracami [10] używa algorytmu AAM [28]. Jest to jedno z najpopularniejszych rozwiązań śledzenia pozycji głowy w ostatnich latach. Jako druga przedstawiona jest metoda używająca sinusoidalnego modelu głowy oraz algorytmu POSIT [13] do wyznaczania aktualnej pozycji. W pracy [30] została zaprezentowane połączenie obu tych metod.

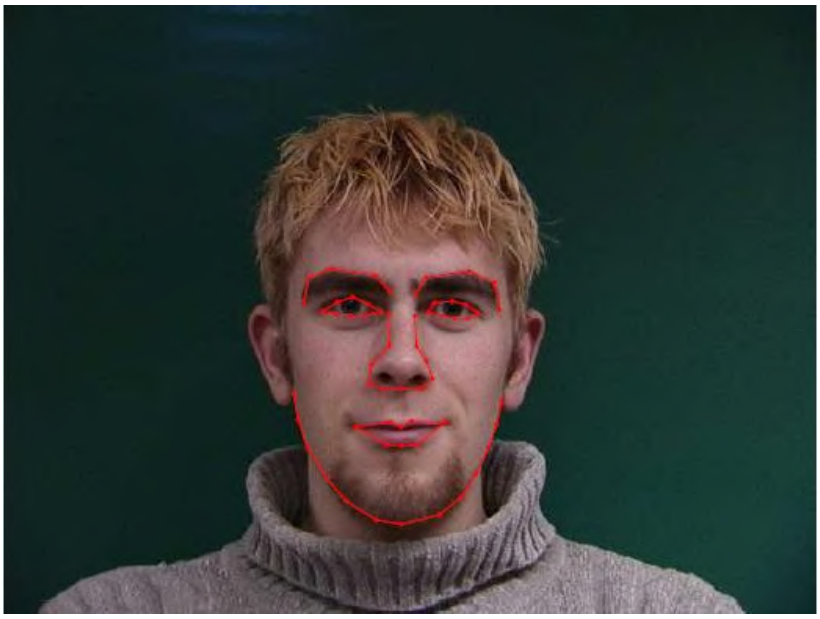
5.1 Active Appearance Model

Aktywny model wyglądu (AAM) [28] to metoda służąca do detekcji oraz śledzenia dowolnych obiektów, których model jest tworzony podczas fazy uczenia na podstawie odpowiednio przygotowanego zbioru obrazów wejściowych. AAM ma bardzo szeroki wachlarz zastosowań zaczynając od segmentacji obrazów medycznych, rozpoznawania twarzy, jak również śledzenia głowy. Za pomocą algorytmu można modelować dowolny obiekt, którego znany jest kształt oraz tekstura. AAM jest metodą typu "data-driven", co oznacza, że nie ma konieczności ręcznego doboru parametrów określających działanie. Algorytm automatycznie dostraja się do danych podczas procesu inicjalizacji. Jego główna wadą jest fakt, że powodzenie działania bardzo mocno zależy od odpowiedniego doboru zbioru obrazów z naniesionym kształtem obiektu używanych podczas fazy tworzenia modelu. Zbiór danych uczących może składać się z setek obrazów, a prawidłowe naniesienie kształtu obiektu może być zadaniem pracochłonnym. Oczywiście możliwa jest automatyzacja procesu nanoszenia kształtu na zbiorze danych uczących, ale to zagadnienie wykracza poza ramy tej pracy. Do budowy modelu jest wykorzystywana metoda składowych głównych (PCA), oraz zakłada się, że poszukiwane parametry modelu mają rozkład gusowski, co w skrajnych przypadkach może sprawić, że kształt modelu będzie podlegał nierealnym deformacją. Aktywny model wyglądu składa się ze statycznego modelu kształtu oraz modelu tekstury.

5.1.1 Statyczny model kształtu

Aktywny model kształtu jest strukturą zawierającą informację o średnim kształcie obiektu danego typu (np. twarzy) oraz dane opisujące najbardziej charakterystyczne modyfikacje tego kształtu, zaobserwowane w zbiorze uczącym. Postać modelu może być modyfikowana przez algorytmy, które starają się dopasować go do rzeczywistego kształtu, nie dopuszczając jednocześnie do nienaturalnych deformacji. Tworzenie modelu rozpoczyna się od zbudowania wzorca obiektu, tj. zbioru etykietowanych punktów reprezentujących dany kształt. Jest to tzw. model rozkładu punktów (Point Distribution Model – PDM), zilustrowany na rys. 5.1. Punkty charakterystyczne muszą być następnie naniesione w adekwatnych miejscach na wszystkich N obrazach uczących. W ten sposób otrzymujemy zbiór kształtów uczących, zapisanych w postaci wektorów zawierających współrzędne punktów charakterystycznych. Dane dotyczące rozmiaru i położenia obiektów są usuwane z wektorów przez specjalną procedurę normalizacyjną, tak, że pozostaje tylko informacja o kształcie. Automatyczne pozycjonowanie punktów charakterystycznych na obrazach stanowi bardzo złożony problem, tak więc najbezpieczniejszą metodą uzyskania zbioru przykładowych kształtów jest ich ręczne zaznaczanie, będące niewątpliwie żmudnym i czasochłonnym zajęciem.

Kształt jest zdefiniowany jako zbiór punktów 2D tworzący siatkę rozpiętą na śledzonym obiekcie. Punkty orientacyjne (landmarks) mogą być umieszczone na obrazie automatycznie lub ręcznie przez użytkownika. Matematyczny zapis kształtu s wyrażony jest przez wektor $2n$ wymiarowy.



Rysunek 5.1: Obraz twarzy z naniesionym kształtem

$$s = [x_1, x_{2,} \ldots, x_n, y_1, y_{2,} \ldots, y_n,]$$

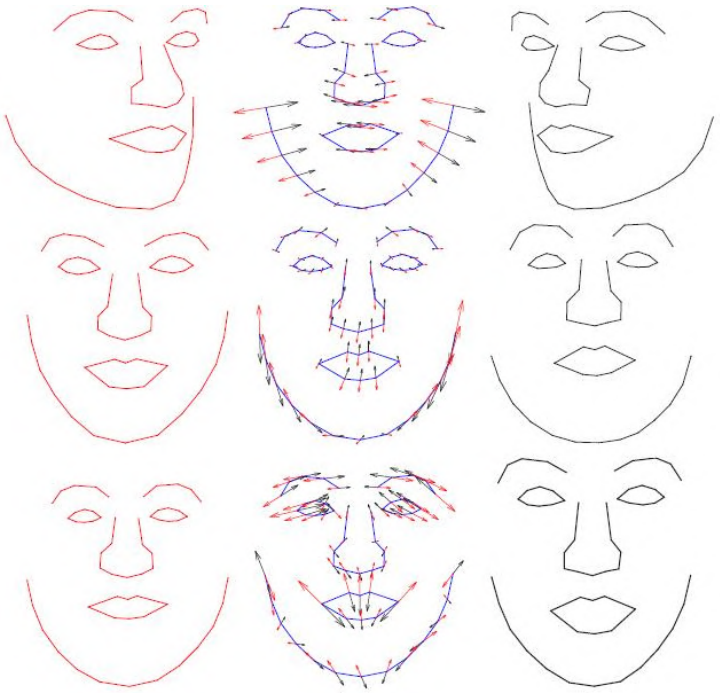
Zbiór danych uczących jest podawany normalizacji, a następnie wyznaczane są jego statyczne parametry przy użyciu analizy głównych składowych (PCA). Do analizy głównych składowych konieczne jest wyliczenie wartości oraz wektorów własnych macierzy kowariancji C, gdzie przez N oznacza się ilość obrazów w zbiorze trenującym.

$$C = \frac{1}{N-1} SS^T$$

$$S = [s_1 - s_0, s_1 - s_0, \ldots s_N - s_0]$$

PCA jest stosowane jako metoda do redukcji wymiarowości zbioru danych. Używane są tylko wektory własne odpowiadające największym wartością własnym. Ilość stosowanych wektorów t zależy od zróżnicowania zbioru danych wejściowych. Umożliwia to aproksymowanie instancji kształtu s jako kombinacje liniową wektorów własnych macierzy kowariancji.

$$s \approx s_0 + \Phi_s b_s$$



Rysunek 5.2: Deformacja kształtu twarzy

Gdzie b_s jest wektorem parametrów opisanym następująco:

$$b_s = \Phi_s^T(s - s_0)$$

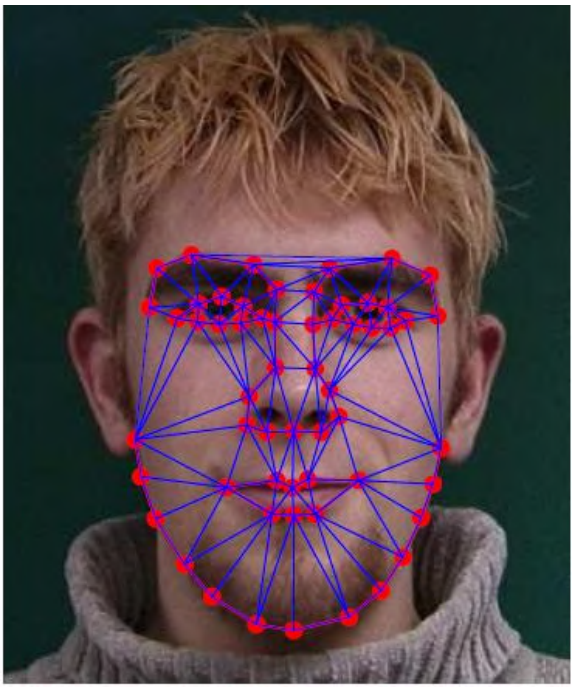
a Φ_s^T jest macierzą zawierającą t wektorów własnych.

Wykorzystując model kształtu utworzony stosując PCA możliwe jest generacja nowych obiektów o kształcie podobnym do zbioru uczącego.

Na rysunku rys. 5.7 przedstawiono generacje nowych kształtów poprzez deformacje kształtu głównego. W środkowym rzędzie znajduje się kształt główny z zaznaczonym kierunkiem deformacji, natomiast po prawej i lewej stronie mieszczą się deformacje tego kształtu ze współczynnikiem równym odchyleniu standardowemu podzielonemu przez faktor $b_{s_i} = \pm 2\sqrt{\lambda_i}$

5.1.2 Statyczny model tekstury

Model tekstury jest tworzony na podstawie zbioru obrazów uczących z naniesioną siatką 2D twarzy. Siatka jest wyznaczana stosując metodę triangulacji Delone [29] na zbiorze punktów charakterystycznych opisanych w poprzednim rozdziale. Twarz z naniesioną siatką przedstawia rys. 5.3.



Rysunek 5.3: Twarz z naniesioną siatką 2D

Tekstura g dla każdego obrazu wejściowego jest zdefiniowana jako intensywność pikseli wewnątrz siatki rozpiętej na punktach charakterystycznych .

$$g = [g_1, g_2, \ldots, g_m]^T$$

Model tekstury opisuje różnice wyglądu między obrazami wejściowymi. Do utworzenia takiego modelu konieczne jest, aby na wszystkich obrazach uczących odpowiednie części twarzy się pokrywały. Można to osiągnąć stosując deformacje kształtu poszczególnych twarzy na kształt główny s_0 wyznaczony w poprzednim etapie. Transformacja zbioru obrazów uczących do wspólnego kształtu jest wyznaczana stosując mapowanie afiniczne poszczególnych trójkątów siatki twarzy.

Transformacja tekstury obrazu uczącego do postaci referencyjnej jest wykonywana w następujący sposób:

1. Dla każdego piksela twarzy wyszukiwany jest trójkąt siatki 2D, w którym jest on zlokalizowany.
2. Wykonywana jest transformacja afiniczna wyznaczonego trójkąty tak, aby jak najlepiej pasował on do kształtu głównego.
3. Przekształcony trójkąt nanoszony jest na obraz wyjściowy.

Po wstępnej normalizacji następuje przetworzenie zbioru wejściowego metodą składowych głównych (PCA) w analogiczny sposób, jak był tworzony model kształtu. Kolumnami macierzy G są wektory znormalizowanych tekstur obrazów uczących g. Macierz kowariancji jest wyliczana następująco.

$$\sum_g = \frac{1}{N-1} G^T G$$

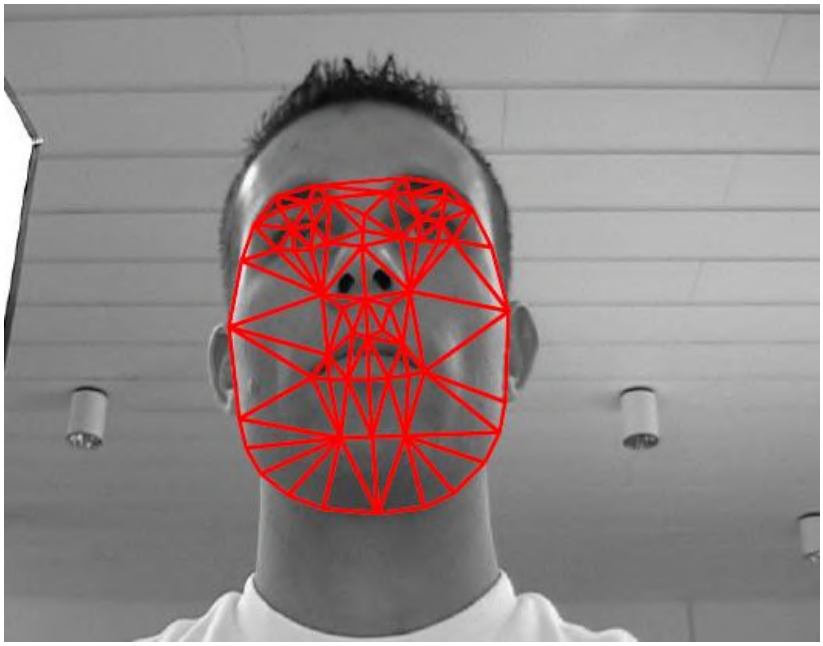
Nowa tekstura jest generowana jako kombinacja liniowa wektorów własnych macierzy kowariancji.

$$g = g_0 + \Phi_g b_g$$

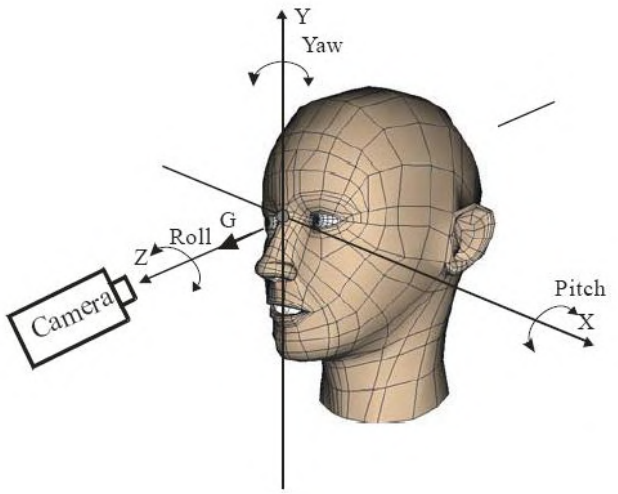
Gdzie b_g oznacza wektor parametrów. Rysunek 5.4 przestawia wynik działania metody. Na środku znajduje się tekstura główna twarzy, a po lewej i prawej stronie tektury utworzone przez jej deformację.

Częstym podejściem do określenia pozycji 3D głowy jest stosowanie algorytmu AAM (Active Appearance Model). Główna idea opiera się na stworzeniu modelu 2D wyglądu twarzy trenując go przygotowanymi obrazami z naniesioną siatką 2D obrysowującą kontur twarzy, oczu, nosa i ust (rys. 5.5). Następnie tak przygotowany model jest dopasowywany do nowych obrazów wyznaczając cechy określone podczas uczenia modelu. Badając zniekształcenie siatki 2D jest możliwe przybliżone określenie pozycji 3D. Przykładem może być praca [14].

Rysunek 5.4: Tekstura główna twarzy oraz jej deformacje



Rysunek 5.5: Active apperance model



Rysunek 5.6: Model głowy wraz osiami obrotu

Metoda została przetestowana przy użyciu gotowej implementacji algorytmu zawartej w bibliotece open source: The AAM-API [11].

Zaletami są duża skuteczność oraz dokładność, ale tylko w przypadku, kiedy model zostanie przygotowany dla danej osoby. Kiedy model zostanie stworzony dla dużej bazy osób dochodzi do nieprawidłowego dopasowania szukanych cech. Konieczność ręcznego nanoszenia siatki na twarz użytkownika sprawia, że taki program wymagałby skomplikowanej konfiguracji. Było to powodem, dla którego zrezygnowano z tego podejścia.

5.2 Wyznaczanie pozycji głowy przy użyciu modelu 3D

Pozycja głowy jest określona przez sześć stopni swobody: trzy kąty obrotu (rys.5.6) oraz trzy wartości przesunięcia $M(x, y, z)$. Obrót głowy można scharakteryzować przez trzy kąty Eurela: wokół osi z (roll, θ), następnie wokół osi y (yaw, β) a na końcu wokół osi x (pitch, α).

Macierz rotacji R jest wyznaczana na postawie znajomości trzech kątów Eulera.

$$R_z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ sin\beta & 0 & cos\beta \end{bmatrix}$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

Algorytm wyznaczania pozycji 3D obiektu opiera się na uproszczonym modelu kamery zwany "camera obscura". Idea działania polega na aproksymacji parametrów modelu na podstawie estymacji projekcji cech obiektu najlepiej dopasowanych do lokacji tych cech na obrazie. Używając uproszczonego modelu kamery projekcja punktu $\overrightarrow{a}$ modelu 3D na plan obrazu $\overrightarrow{b}$ przy założeniu braku zniekształceń wywołanych niedoskonałością soczewek obiektywu może być opisana w następujący sposób:

$$\overrightarrow{b} = T\overrightarrow{a}$$

$$u_x = f\frac{b_x}{b_z}$$

$$u_y = f\frac{b_y}{b_y}$$

gdzie T oznacza macierz transformacji w homogenicznym układzie współrzędnych. Macierz T jest złożeniem następujących operacji geometrycznych: obrót wokół osi współrzędnych o kąty θ,β i na końcu α oraz translacji o wektor M

$$T = M(x, y, z)R_z(\theta)R_y(\beta)R_x(\alpha)$$

Współczynnik f reprezentuje wartość ogniskowej obiektywu. Współrzędne obrazu przedstawione w pikselach $\overrightarrow{q}$ są obliczane następująco:

$$q_x = \frac{u_x}{s_x} + c_x$$

$$q_y = \frac{u_y}{s_y} + c_y$$

Współczynnik s określa fizyczną wartość odległości środków dwóch sąsiednich pikseli na matrycy kamery, a c oznacza przemieszczenie między osią optyczna a środkiem matrycy. Dla uproszczenia obliczeń można założyć, że wartość $\overrightarrow{c}$ jest zerowa. Odpowiada to sytuacji, kiedy matryca jest

umieszczona idealnie na osi optycznej obiektywu. Wartość f można wyznaczyć eksperymentalnie podczas procesu kalibracji kamery. Do dalszych obliczeń założono znajomość f. Ustalenie stałej wartości f nie ma dużego wpływu na dalsze działanie algorytmu, ponieważ do dalszego działania wystarczające jest wyznaczanie względnych zmian pozycji głowy. Po zastosowaniu tych uproszczeń aktualna pozycja głowy $\overrightarrow{p}$ opisana jest za pomocą sześciu zmiennych

$$\overrightarrow{p} = \{x, y, z, \alpha, \beta, \theta\}$$

W ogólnym przypadku projekcja punktów obiektu 3D na plan 2D jest operacja nieliniową, ale przy założeniu małych zmian pomiędzy znaną pozycją wyznaczoną w poprzedniej klatce a aktualną pozycją. Operacja ta może być dobrze przybliżona funkcją liniową. Z tym założeniem parametry opisujące pozycje mogą być wyznaczane iteracyjne.

$$\overrightarrow{p}_{i+1} = \overrightarrow{p}_i - \overrightarrow{d}$$

W każdym kroku iteracji wektor korelacji $\overrightarrow{d}$ jest obliczany przy pomocy minimalizacji wektora $\overrightarrow{e}$ błędu będącego sumą odległości między projekcją punktów modelu, a pozycją odpowiednich cech na obrazie. Stosując metodę Newtona, gdzie przez J oznacza się macierz jakobianu, $\overrightarrow{d}$ wyznacza poniższe równanie:

$$J\overrightarrow{d} = \overrightarrow{e}$$

Równanie to można rozwiązać stosując pseudo inwersje

$$\overrightarrow{d} = (J^t J)^{-1} J^t \overrightarrow{e}$$

Algorytm Levenberga-Marquardta

Przedstawiona metoda wyznaczania pozycji obiektu sprawia, że wszystkie parametry podczas optymalizacji są traktowane z jednakową wagą. Przykładowo rotacja o 0,5 radiana wokół osi z może mieć znacznie większy wpływ niż zmiana projekcji wywołana przesunięciem o 50 mm. Konieczne jest normalizacja współczynników uwzględniając odchylenie standardowe każdego wiersza macierzy J. Do tego celu wprowadza się macierz W, której elementy leżące na przekątnej są odwrotnie proporcjonalne do odchylenia standardowego σ.

$$W_{ii} = \frac{1}{\sigma_{p_i}}$$

Dla poprawy zbieżności metody dodawany jest parametr λ w celu kontroli wagi stabilizacji.

$$\overrightarrow{d} = (J^t J + \lambda W^t W)^{-1} J^t \overrightarrow{e}$$

Wyznaczanie jakobianu J jest operacją złożoną obliczeniowo. Sprawia to, że ta metoda nie jest najlepszym rozwiązaniem do zastosowań mających działać w czasie rzeczywistym. Było to powodem rezygnacji z tego podejścia i zastosowania algorytmu POSIT.

5.2.1 Algorytm POSIT

Algorytm POSIT (Pose from Orthography and Scaling with Iteration) [13] służy do estymacji pozycji w trzech wymiarach znanego obiektu. Został on zaprezentowany w 1992 jako metoda do wyznaczania pozycji (pozycja określona przez wektor translacji T oraz macierz orientacji R) obiektu 3D, o znanych wymiarach. Do wyznaczenia pozycji konieczne jest określenie co najmniej czterech punktów (nie leżących na jednej płaszczyźnie) na powierzchni obiektu. Algorytm składa się z dwóch części: wstępna estymacja pozycji oraz iteracyjna poprawa wyniku. Pierwsza część algorytmu (POS) zakłada, że wyznaczone punkty obiektu znajdują się w tej samej odległości od kamery oraz różnica rozmiaru obiektu związana ze zmianą dystansu do kamery jest pomijalne mała. Założenie, że punkty są w tej samej odległości oznacza, że obiekt znajduje się wystarczająco daleko od kamery i można pominąć różnice głębi (założenie słabej perspektywy). Dzięki takiemu podejściu znając parametry kamery można wstępnie wyznaczyć pozycje obiektu używając skalowania perspektywicznego. Wyliczenia takie są przeważnie nie wystarczająco dokładne, dlatego stosowana jest interakcyjna poprawa rezultatu. Przy użyciu wyliczonej pozycji w poprzedniej iteracji punkty są podawane projekcji na obiekt 3D. Wynik jest używany jako punkt startowy pierwszego etapu algorytmu.

Metoda ta umożliwia wyznaczenie pozycji 3D obiektu za pomocą widoku z pojedynczej kamery. Do działania konieczna jest znajomość aktualnej mapowanej pozycji 2D co najmniej 4 punktów nie leżącej na jednej płaszczyźnie oraz ich współrzędnych 3D w modelu obiektu. Algorytm nie uwzględnia perspektywy. Nie ma to jednak wpływu na działanie, jeśli obiekt jest dostatecznie oddalony od kamery, ponieważ wtedy wpływ perspektywy na wygląd obiektu jest znikomy.

Opisany algorytm jest bardzo dobrym wyborem w przypadku, gdy aplikacja powinna działać w czasie rzeczywistym, ponieważ jego złożoność obliczeniowa jest niewielka. Jego podstawowa wadą jest konieczność wyznaczenia punktu referencyjnego, którego współrzędne 3D muszą być zerowe. Nieprawidłowe wyznaczenie projekcji punktu referencyjnego ma bardzo duży wpływ na zakłócenie prawidłowej oceny aktualnej pozycji obiektu. Dodatkowym problemem może być sytuacja, kiedy projekcja punktu referencyjnego nie będzie widoczna. Taka sytuacja jest możliwa przy dużym obrocie głowy, kiedy przykładowo nos zasłoni cechy znajdujące się po jego jednej ze stron. Rozwiązaniem tego problemu jest wyznaczanie lokacji referencyjnej cechy wykorzystując zbiór punktów z jej sąsiedztwa.

Algorytm wyznaczający pozycje 3D głowy składa się z dwóch głównych etapów

1. Inicjalizacja algorytmu. Wykonywana jest detekcja twarzy i oczu. Aby można było przystąpić do dalszej analizy konieczne jest wyodrębnienie rejonu twarzy od otoczenia. Następnie wyznaczany jest zbiór cech, które będą śledzone w kolejnym etapie. Na podstawie wyznaczonych cech oraz znanej pozycji twarzy tworzony jest model 3D głowy.

2. Śledzenie pozycji 3D głowy. Etap powtarzany iteracyjne po zainicjowaniu algorytmu. Na podstawie wykrytej zmiany pozycji punktów określonych w poprzednim etapie wyznaczana jest aktualna pozycja głowy w przestrzeni 3D opisana przez wektor translacji T oraz macierz obrotu R.

Poszczególne fazy algorytmu zostały szczegółowo przedstawione poniżej.

5.2.2 Detekcja twarzy i oczu.

Do wykrycia pozycji twarzy na obrazie została zastosowana metoda zaproponowana przez Poula Viole oraz Michaeala Jones'a [12]. Jest to jedno z najbardziej popularnych rozwiązań stosowanych do wykrywania twarzy. Charakteryzuje się dużą szybkością działania oraz bardzo dobrą skutecznością. Metoda służy do lokalizacji na obrazie obiektów za pomocą wcześniej stworzonego klasyfikatora. Klasyfikator jest trenowany przy użyciu obrazów zawierających szukany obiekt oraz obrazów, na których dany obiekt nie występuje. Przy użyciu tak przygotowanego zbioru obrazu tworzony jest klasyfikator kaskadowy, który następnie jest używany do lokalizacji obiektu na nieznanych obrazach.

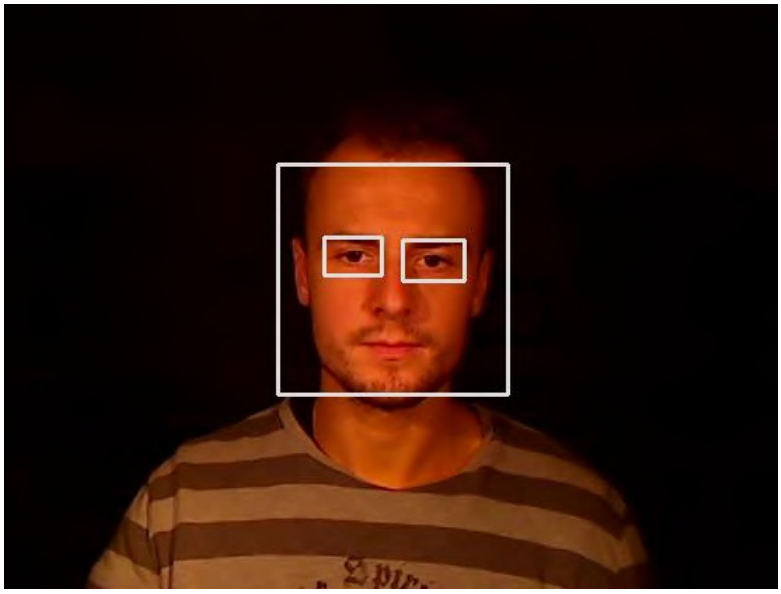
W pracy użyto implementację algorytmu zawartą w bibliotece Open Computer Vision Library (OpenCV) [20]. Jest to projekt typu open source tworzony przez firmę Intel. Projekt ten zawiera gotowe klasyfikatory służące do wykrywania twarzy oraz oczu.

Przebieg algorytmu detekcji twarzy i oczu

1. Przeszukanie obrazu przy użyciu klasyfikatora do detekcji twarzy. Wynikiem jest lokalizacja wszystkich odnalezionych twarzy. Do dalszej analizy jest uwzględniana twarz o największym rozmiarze, czyli osoby znajdującej się najbliżej kamery.

2. Wykorzystując wcześniej określoną lokalizacje twarzy zostaje zastosowany kolejny klasyfikator służący do detekcji oczu. W celu mniejszego obciążenia procesora przeszukiwana jest tylko górna połowa twarzy. Zmniejsza to także ryzyko nieprawidłowego wykrycia oczu, np. w miejscu ust lub nosa.

3. Sprawdzana jest wielkość oraz położenie wzajemne twarzy oraz oczu. Wymagane jest, aby rozmiar obu oczu oraz ich położenie poziome były do siebie zbliżone. Jeśli nie są spełnione te warunki cały proces detekcji jest wykonywany od początku na nowej klatce.



Rysunek 5.7: Detekcja twarzy i oczu

Wynik działania algorytmu przedstawia rys. 5.7.

Połączenie działania klasyfikatora do detekcji twarzy oraz oczu daje znacznie lepsze rezultaty niż stosowanie pojedynczego klasyfikatora twarzy. Na podstawie przeprowadzonych doświadczeń można stwierdzić, że stosując tylko klasyfikator do detekcji twarzy można otrzymać błędne rezultaty, np. obszar obrazu, na którym nie znajduje się twarz, może zostać niepoprawnie sklasyfikowany. Taka sytuacja ma przeważnie miejsce w przypadku warunków oświetlenia odmiennych od tych, które były na obrazach użytych do wytrenowania klasyfikatora. Szczególnie często nieprawidłowy rezultat osiągano przy oświetleniu bocznym twarzy.

Połączenie działania obu klasyfikatorów oraz sprawdzenie wzajemnej lokalizacji oczu oraz ich rozmiaru daje znacznie większą pewność działania algorytmu. Cena, jaką za to trzeba zapłacić jest fakt, że czasami, mimo iż na obrazie wejściowym znajduję się twarz, nie zostanie ona odnaleziona. Nie jest to jednak dużym problemem, ponieważ w programie wykorzystuje się sekwencje wideo i można pominąć kilka początkowych klatek. Algorytm detekcji twarzy jest powtarzany interakcyjne do momentu wykrycia twarzy, oczu oraz spełnienia warunków ich wzajemnego położenia.

5.2.3 Wyznaczanie cech do śledzenia

Komercyjne systemy do detekcji mimiki i ruchu głowy używają markerów z materiału odbijającego światło podczerwone naklejonych na twarz. Rozwiązanie to jest bardzo dokładne i odporne na

zakłócenia. Wymaga jednak drogiego sprzętu. Założeniem pracy było stworzenie systemu niewymagającego skomplikowanej konfiguracji sprzętowej. Przedstawione rozwiązanie bazuje wyłącznie na standardowej kamerze internetowej bez stosowania żadnych dodatkowych markerów poprawiających działanie algorytmów.

Do śledzenia zmian pozycji głowy został wykorzystany dyskretny przepływ optyczny Lucasa-Kalmana opisany w części teoretycznej. Do działania wymaga on wyznaczenia zbioru cech, których zmiana pozycji będzie śledzona w kolejnych klatkach sekwencji wideo. Cechy muszą zostać tak dobrane, aby możliwe było jednoznaczne wyznaczenie zmiany ich położenia w kolejnych klatkach. Algorytm Lucasa-Kalmana dobrze działa, jeśli śledzone punkty są zlokalizowane w miejscach istnienia ostrych krawędzi. Dobór odpowiedniego zbioru cech jest bardzo istotną kwestią. Ma on zasadniczy wpływ na dalszy przebieg śledzenia zmiany pozycji twarzy. W literaturze można znaleźć wiele różnych podejść do wyznaczania punktów dobrych do śledzenia. Jednym z podstawowych kryterium wyboru odpowiedniej metody była złożoność obliczeniowa. Z założenia program powinien pracować w czasie rzeczywistym, więc nie można było sobie pozwolić na wybór skomplikowanej metody, która by zbyt obciążała procesor.

Jedną z najczęściej stosowanych definicji krawędzi została wprowadzona przez Harrisa [12]. Definicja ta opiera się na macierzy pochodnych drugiego stopnia (hesjan) intensywności obrazu. Hesjan w punkcie $p(x, y)$ określony jest w następujący sposób:

$$H(p) = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix}$$

Macierz autokorelacji M Hesjanu jest wyznaczana przez sumowanie wartości drugich pogodnych w sąsiedztwie danego punktu:

$$M(x, y) = \begin{bmatrix} \sum_{-K \le i,j \le K}^{n} I_x^2(x+i, y+j) & \sum_{-K \le i,j \le K}^{n} I_x(x+i, y+j) I_y(x+i, y+j) \\ \sum_{-K \le i,j \le K}^{n} I_x(x+i, y+j) I_y(x+i, y+j) & \sum_{-K \le i,j \le K}^{n} I_y^2(x+i, y+j) \end{bmatrix}$$

Krawędzie znajdują się w miejscach, w których macierz autokorelacji hesjanu ma dwie duże wartości własne. Oznacza to, że tekstura w otoczeniu danego punktu znacząco zmienia się w dwóch niezależnych kierunkach. Dzięki wykorzystaniu tylko wartości własnych, wyznaczone krawędzie nie zmieniają się przy obrocie obrazu. Wyszukując maksima lokalne autokorelacji obrazu wejściowego można w prosty sposób uzyskać punkty, których śledzenie będzie możliwe przy użyciu przepływu optycznego.

Bardzo istotne jest, aby wyznaczone cechy były rozmieszczone równomiernie na powierzchni śledzonego obiektu. Osiągnąć to można ograniczając minimalną odległość między sąsiednimi punk-

tami. Takie założenie jest konieczne, ponieważ w przypadku, gdy autokorelacja hesjanu tekstury obiektu będzie się charakteryzować dużym maksimum w jednym miejscu, rezultatem będzie lokalizacja większości cech wokół tego maksimum. W takim przypadku algorytm wyznaczający pozycję 3D obiektu może dawać nieprawidłowe wyniki.

Przedstawione podejście opiera się na wyznaczeniu naturalnych cech twarzy, które są łatwe to śledzenia za pomocą przepływu optycznego. Algorytm wyszukuje lokalne maksima, co sprawia, że adaptacyjne w zależności od wyglądu twarzy danej osoby wyszukuje optymalne punkty do śledzenia. Takie rozwiązanie jest znacznie bardziej uniwersalne i daje lepsze rezultaty niż bazowanie na ściśle określonych cechach twarzy, takich jak kąciki ust, rogi oczu itp. Wyznaczanie ustalonych cech nie zawsze jest możliwe. W sytuacji, gdy założymy, że korzystamy np. z rogów oczu, a algorytm wyszukiwania tych cech nie przewidzi faktu, że osoba może nosić okulary, może to prowadzić do pogorszenia rezultatu, a nawet do nieprawidłowych wyników.

Algorytm wyznaczenie cech do śledzenia

1. Określenie obszaru poszukiwania. Badanie występowania cech jest ograniczone do obszaru twarzy wyznaczonego podczas poprzedniego etapu. Nie jest brany pod uwagę rejon, w którym zlokalizowane zostały oczy, ponieważ ruchy gałki ocznej będą zakłócać prawidłowe wyznaczenie pozycji głowy.

2. Wyznaczanie cech. Do wyznaczenia cech został zastosowany algorytm Harrisa [12].

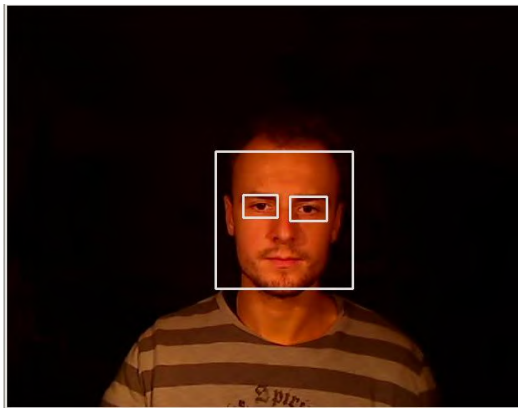
3. Pominięcie punktów zbyt blisko leżących obok siebie. W celu zwiększenia wydajności eliminowane są cechy zlokalizowane zbyt blisko siebie.

5.2.4 Inicjalizacja modelu 3d głowy

Głowa jest modelowana za pomocą siatki sinusoidalnej przedstawionej na rys. 5.9. Jest to estymacja przybliżona nieuwzględniająca szczegółowego kształtu twarzy, ale jej działanie jest w zupełności zadowalające i nie wymaga skomplikowanego podejścia.

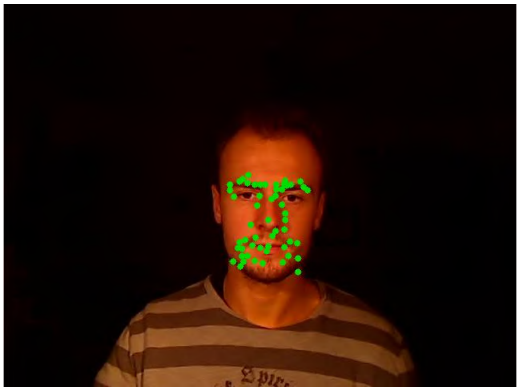
Takie uproszczenie modelu ma szereg zalet:

- Łatwość wyznaczenia modelu,
- Automatyczna inicjalizacja. Nie ma konieczności wcześniejszego przygotowania siatki modelu dla danego użytkownika,
- Uniwersalność związana z brakiem konieczności uwzględnienia indywidualnego profilu twarzy,
- Szybkość działania wynikająca z prosty,



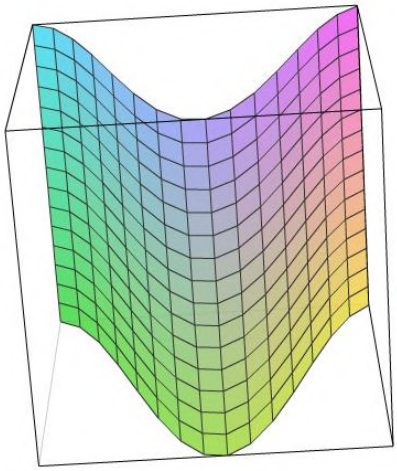
(a) obraz wejściowy z zaznaczonym rejonem twarzy i oczu

(b) intensywność krawędzi (autokorelacja Hesjanu intensywności obrazu)



(c) wyznaczone cechy

Rysunek 5.8: Wynik działania algorytmu wyszukiwania cech służących do wyznaczania zmiany pozycji głowy

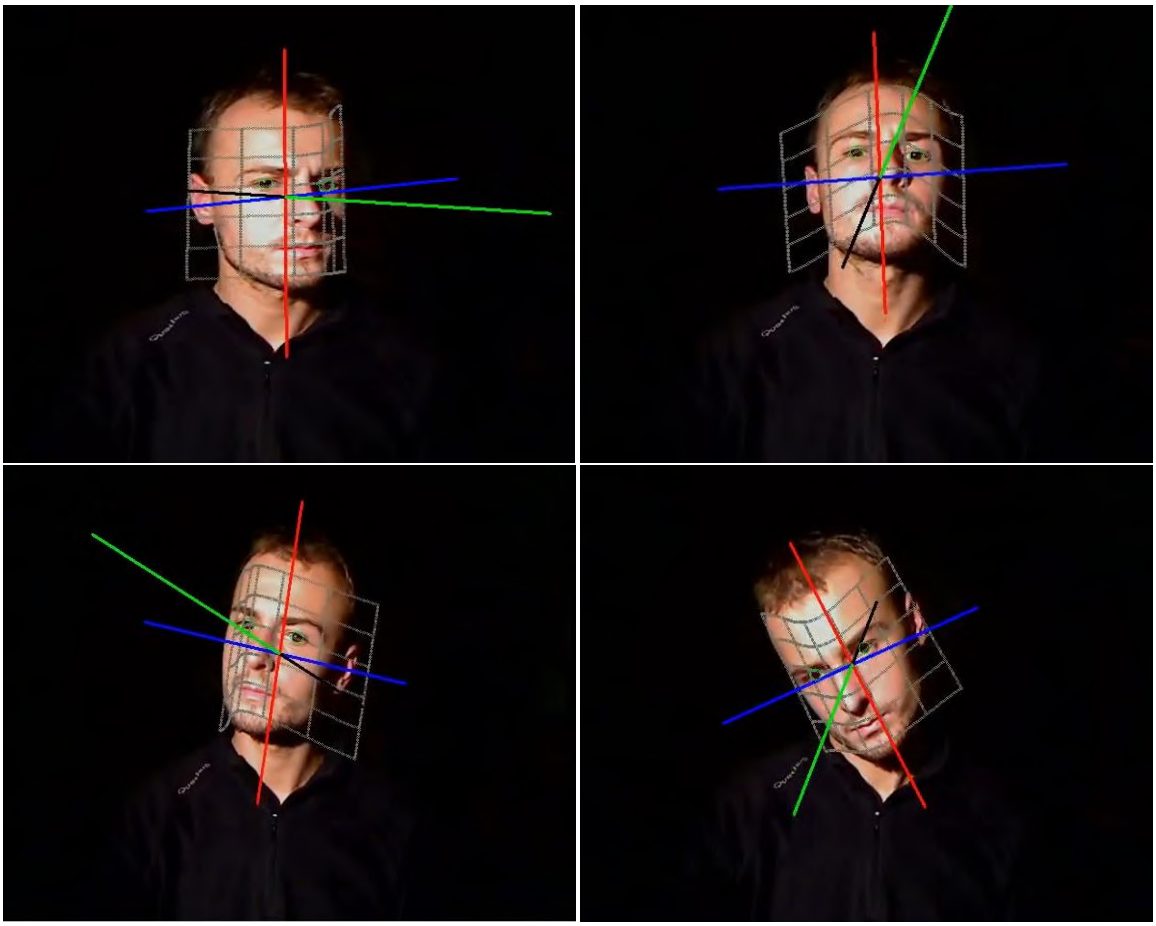


Rysunek 5.9: Sinusoidalny model głowy

- Odporność na zakłócenia.

Mapowanie cech twarzy (2D) wyznaczonych w poprzednim etapie na siatkę 3D modelu jest możliwe dzięki założeniu, że podczas inicjalizacji głowa użytkownika jest skierowana na wprost w kierunku monitora. Założenia takie można poczynić, ponieważ model będzie służył do względnej oceny zmiany pozycji głowy w stosunku do pozycji ustalonej podczas inicjalizacji. Przykład dopasowania siatki do twarzy jest przedstawiony na rys. 5.10

Stosowanie uproszczonego modelu twarzy umożliwia łatwą inicjalizacja systemu. Proste modele takie jak cylinder, elipsoida czy sinusoida są powszechnie stosowane do śledzenia ruchu głowy, ponieważ inicjalizacja takich modeli wymaga doboru znacznie mniejszej ilości parametrów niż w przypadku rzeczywistego modelu twarzy dla indywidualnej osoby. Na podstawie eksperymentów można stwierdzić, że inicjalizacja uproszczonego modelu jest relatywnie odporna na nieprecyzyjny dobór parametrów początkowych, takich jak dokładny rozmiar głowy, czy aktualna jej pozycja. W przypadku szczegółowego modelu bardzo istotna jest precyzyjna inicjalizacja, ponieważ niewielka zmiana parametru startowego ma duży wpływ na dalszy przebieg śledzenia obiektu. Dysponując nawet precyzyjną siatką twarzy danej osoby uzyskaną, np. za pomocą skanera laserowego, jeśli podczas inicjalizacji algorytmu śledzącego zmianę położenia zostanie nieprecyzyjnie dobrane startowa pozycja, dokładny model twarzy staje się bezużyteczny. Podczas inicjalizacji konieczne jest ustalenie aktualnej pozycji modelu tak, aby siatka twarzy użytkownika pokrywała się z obrazem uzyskanym z kamery. Proces ten może zostać wykonany automatycznie przy założeniu, że znana jest pozycja kamery oraz osoba patrzy w kierunku kamery podczas inicjalizacji. Pozycje głowy można wtedy wyznaczyć poprzez detekcje twarzy za pomocą metod opisanych powyżej. Siatka sinusoidalna



Rysunek 5.10: Głowa z nałożoną siatką 3D modelu

nie jest precyzyjną estymacją rzeczywistego modelu głowy, ale korzystając z uproszczonego modelu można przyjąć, że dokładność doboru parametrów startowych będzie wystarczająca do prawidłowego badania zmian pozycji głowy. W przypadku zastosowania dokładnego modelu twarzy, proces inicjalizacji musiał by być wykonany manualnie. Było by to znaczącym utrudnieniem i sprawiło by, że system stał by się mało praktyczny. Inicjalizacja systemu wymagała by dodatkowej wiedzy od użytkownika.

W pracy [24] zostało przedstawione porównanie stosowania modelu cylindrycznego oraz dokładnego modelu uzyskanego za pomocą skanera laserowego Cyberware, w algorytmie śledzenia zmiany pozycji głowy. Z rezultatów tej pracy jasno wynika, że uproszczony model jest w zupełności wystarczający, a jego główna zaletą jest możliwość automatycznej inicjalizacji.

W prezentowanym rozwiązaniu został zastosowany model sinusoidalny. Jest on nieco bardziej skomplikowany od modelu cylindrycznego. Jego główną zaletą jest lepsze odzwierciedlenie kształtu nosa, co sprawia, że pozycja głowy w przypadku dużych skręceń jest wyznaczane z większą dokładnością.

Inicjalizacja algorytmu śledzącego zmianę pozycji oparte na wykorzystaniu modelu 3D obiektu, polega na wyznaczeniu relacji między punktami 3D leżącymi na powierzchni siatki obiektu, a projekcją tych punktów na startowej klatce sekwencji wideo. Wyznaczanie punktów do śledzenia przy użyciu algorytmu Harrisa zostało opisane powyżej. Mając zbiór takich punktów konieczne jest wyznaczenie ich lokacji na powierzchni modelu. Nie jest to w ogólnym przypadku zadaniem prostym, ponieważ w przypadku, gdy model jest skomplikowany nie można w analityczny sposób jednoznacznie określić mapowania z przestrzeni 2D do 3D. Jednym z prostszych rozwiązań jest wykorzystanie biblioteki OpenGl. Każdemu trójkątowi siatki obiektu przypisuje się odmienny kolor. Następnie taki obiekt jest renderowany z wykorzystaniem startowej pozycji obiektu. Odczytując kolor pokrywający dany punkt wyznacza się zależność między punktem na obrazie a jego odpowiednikiem w modelu. Jest to rozwiązanie ogólne i nadaje się do szerokiej klasy modeli. W przypadku stosowania uproszczonego modelu sinusoidalnego, który można opisać za pomocą równań algebraicznych nie ma konieczności stosowania tak skomplikowanej inicjalizacji.

Projekcja punktu modelu 3D na płaszczyznę projekcji $\overrightarrow{q}$, przy założeniu uproszczonego modelu kamery opisanego w sekcji 5 przedstawiają równania:

$$u_x = f\frac{b_x}{b_z}$$

$$u_y = f\frac{b_y}{b_y}$$

$$q_x = \frac{u_x}{s_x} + c_x$$

$$q_y = \frac{u_y}{s_y} + c_y$$

W powyższych równaniach występują parametry określające własności kamery, jednak bez straty ogólności można przyjąć, że te parametry są stałe i ustalone z góry. Takie założenia są możliwe, ponieważ dla działania systemu nie jest istotne wyznaczenie bezwzględnej pozycji głowy. Konieczne jest wyłącznie określenie zmiany pozycji między kolejnymi klatkami sekwencji wideo. Wynika to z faktu, że podczas kalibracji wzroku zostaną uwzględnione odpowiednie współczynniki. Na etapie inicjalizacji nie jest konieczne ustalanie dokładnego modelu kamery.

Istnieje w wiele metod [23] umożliwiających wyznaczenie parametrów kamery, ale wymaga to dodatkowego procesu kalibracji z wykorzystaniem specjalnie przygotowanych obrazów. Jedną z najbardziej popularnych metod jest użycie szachownicy o znanym rozmiarze pól. Obraz tej szachownicy jest rejestrowany w kilku różnych pozycjach. Na podstawie tych danych obliczany jest model kamery. Dzięki takiemu podejściu, można wyznaczać bezwzględną pozycję 3D śledzonego obiektu. Jednak proces kalibracji musi być przeprowadzony indywidualnie dla każdej kamery.

Zakładana jest znajomość startowej pozycji głowy. Przy założeniu, że użytkownik podczas inicjalizacji spogląda prosto w kamerę można ustalić, że macierz obrotu R jest macierzą jednostkową. Natomiast wektor translacji T jest określany na postawie znanej pozycji głowy wyznaczonej automatycznie podczas procesu detekcji twarzy i oczu. Współrzędne x oraz y wektora T są otrzymywane bezpośrednio z aktualnego środka twarzy natomiast współrzędna z jest określana badając skale twarzy. Większy rozmiar twarzy oznacza, że użytkownik jest bliżej kamery, na tej podstawie określa się wartość z wektora T.

5.2.5 Wykorzystanie klatek referencyjnych

Przedstawiona powyżej metoda wykorzystuje informacje o zmianie położenia wybranych cech twarzy tylko między kolejnymi klatkami sekwencji wideo. Takie podejście daje zadowalające wyniki w przypadku, kiedy ilość przetwarzanych klatek nie będzie zbyt duża. W przypadku dłuższych sekwencji zauważalna staje się akumulacja błędu oceny pozycji cech między kolejnymi klatkami. Rozwiązaniem tego problemu było zastosowanie metody zaprezentowanej w pracy [19]. Podejście to polega na stosowaniu zbioru obrazów referencyjnych. Przez obraz referencyjny określana jest klatka obrazu ze znaną aktualną pozycją głowy oraz zbiorem cech twarzy. Kiedy pozycja głowy zbliża się do pozycji zarejestrowanej w klatce referencyjnej następuje niwelacja błędu akumulacji. Sprawia to, że działanie algorytmu śledzenia pozycji głowy nie pogarsza się z upływem czasu, co miało miejsce w przypadku wykorzystywania wyłącznie informacji odnośnie zmiany pozycji cech twarzy między kolejnymi klatkami.

Podczas inicjalizacji systemu tworzona jest startowa klatka referencyjna, a następnie, kiedy pojawi się taka konieczność, dodawane są automatycznie nowe klatki. Podczas śledzenia pozycji

głowy, gdy aktualna pozycja znacząco odbiega od zbioru zarejestrowanych klatek referencyjnych tworzona jest nowa klatka z pozycją wyznaczoną w poprzedniej iteracji algorytmu. Podejście takie sprawia, że algorytm działa w sposób automatyczny. Nie ma konieczności rejestracji zbioru klatek referencyjnych podczas procesu inicjalizacji. Zastosowanie jednej startowej klatki oraz algorytmu automatycznego dodawania następnych klatek dają zadowalające rezultaty i jest rozwiązaniem uniwersalnym.

5.2.6 Eliminacja zakłóconych danych

Do wyznaczania pozycji 3D głowy wykorzystywana jest znajomość aktualnej pozycji cech charakterystycznych twarzy wyznaczonych podczas procesu inicjalizacji. Algorytm wyznaczający pozycję jest oparty na metodzie najmniejszych kwadratów. Pociąga to za sobą konieczność wykrywania cech o nieprawidłowo wyznaczonej pozycji, ponieważ duże odchylenie pozycji nawet pojedynczej cechy może mieć bardzo duży wpływ na działanie całej metody. Zmiana pozycji i punktów jest wyznaczana przez przepływ optyczny. Jest to rozwiązanie precyzyjne, jednak zakłócenia spowodowane nagłą zmianą oświetlenia czy częściowym przysłonięciem twarzy mogą spowodować, że nowo wyliczone położenie cech twarzy nie będzie prawidłowe. Przepływ optyczny określa zmianę położenia między kolejnymi klatkami. Pojedyncze zakłócenie ma wpływ na dalszy przebieg lokalizacji cech. Wynikiem tego była konieczność zaimplementowania metody wykrywania zakłóceń i ich eliminacji. Wynik działania algorytmu detekcji cech o nieprawidłowo wyznaczonej pozycji przedstawiony jest na rys. 5.11. Zielone punkty oznaczają cechy, których lokacja została wyznaczona prawidłowo, natomiast na czerwono oznaczone punkty, które zostały sklasyfikowane poprzez algorytm eliminacji zakłóceń jako nieprawidłowe dane.

Przebieg działania algorytmu wykrywającego nieprawidłowy wynik przepływu optycznego

1. Wyliczenie średniej wartości wektora przesunięcia wszystkich rozważanych cech między kolejnymi klatkami.
2. Wyznaczenie odchylenia standardowego od średniej wartości przesunięcia.
3. Eliminacja cech, których wartość przekracza trzykrotnie średnią wartość odchylenia przesunięcia. Cechy wyeliminowane nie biorą udziału w aktualnej iteracji przy wyznaczaniu pozycji głowy.
4. Poprawa pozycji wyeliminowanych cech za pomocą projekcji punktów odpowiadających danym cechom modelu 3D głowy. Dzięki temu w kolejnej iteracji punkty te mogą ponownie zostać użyte.

Zastosowanie takiego podejścia sprawia, że algorytm wyznaczania pozycji głowy jest odporny na zakłócenia, takie jak zmiana mimiki twarzy, przesłonięcie twarzy dowolnym obiektem, czy nagła zmiana oświetlenia sceny.

Rysunek 5.11: Eliminacja nieprawidłowego działania przepływu optycznego

Rozdział 6

Wyznaczanie kierunku wzroku

Detekcja oka ludzkiego jest trudnym zadaniem z powodu niskiego kontrastu pomiędzy okiem a otaczającą je skórą. W konsekwencji wiele istniejących systemów używa specjalnej kamery montowanej w niewielkiej odległości od oka, w celu uzyskanie dużej rozdzielczości obrazu.

Najpopularniejszą metodą określania kierunku wzroku jest badanie różnicy odległości między środkiem źrenicy, a refleksem świetlnym wywołanym przez źródło światła podczerwonego. Aktualne położenie tych dwóch punktów określa wektor wzroku. Mapowanie między wektorem wzroku, a aktualnym punktem fiksacji wzroku ustalane jest podczas procesu kalibracji. Jest to sposób najprostszy i powszechnie stosowany w systemach komercyjnych. Nie wymaga on skomplikowanych algorytmów przeważania obrazów. Użycie zewnętrznego źródła światła sprawia, że działanie nie jest zależne od warunków oświetlenia.

Z założenia system przedstawiony w tej pracy nie powinien wymagać stosowania żadnego dodatkowego sprzętu poza standardową kamerą internetową. Dlatego do wyznaczenia kierunku wzroku konieczne było zastosowanie metody bazującej wyłącznie na widzialnym spektrum światła. Aktualny kierunek, w jakim patrzy użytkownik, może zostać określony poprzez badanie zmiany konta gałki ocznej w stosunku do osi kamery. Kąt ten jest proporcjonalny do zmiany położenia między środkiem źrenicy, a środkiem gałki ocznej. Zakłada się, że podczas inicjalizacji systemu użytkownik patrzy na środek ekranu. Rejestrowana jest wtedy pozycja środka źrenicy, która następnie jest traktowana jako wartość referencyjna określająca oś oka. Zmiana pozycji osi związana z ruchem głowy może być wyznaczona dzięki znanej pozycji 3D głowy wyznaczonej w poprzednim etapie. Obraz głowy z naniesionymi wektorami wzorku (wektory wzroku są reprezentowane przez żółte linie) przedstawia rys. 7.9.

Rysunek 6.1: Twarz z naniesionymi wektorami wzroku

6.1 Wyznaczanie wektora wzroku

Kierunek, w jakim spogląda dana osoba, można jednoznacznie wyznaczyć badając przesunięcie między środkiem źrenicy a rogami oka. Należy również uwzględnić aktualną pozycję głowy (przesunięcie oraz obrót). Metody wyznaczanie tych wartości zostały zaprezentowane w powyższych rozdziałach.

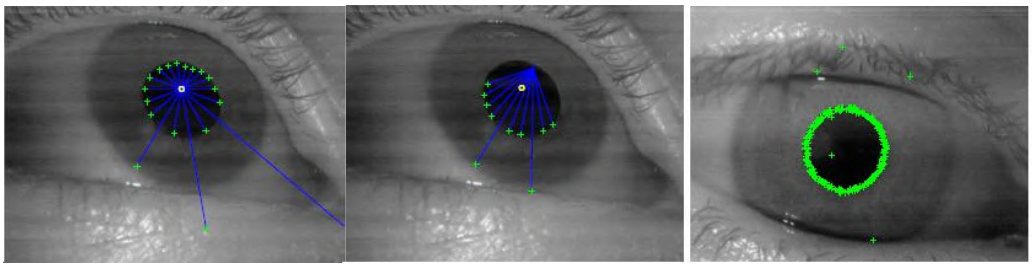
Najprostszym podejściem do wyznaczenia zmiany kąta gałki ocznej jest badanie przesunięcia między źrenicą a rogami oka. Algorytmy wykorzystywane do śledzenia zmiany pozycji oka można podzielić na dwie główne grupy: bazujące na cechach oraz na modelu oka. Podejście oparte na cechach polega na detekcji cech obrazu zależnych od aktualnej pozycji oka. Wymaga to określenia odpowiedniego kryterium zależnego od stosowanej metody określającego wystąpienie poszukiwanej cechy (np. w przypadku binaryzacji konieczne jest określenie wartości progu odcienia). Dobór wartości kryterium jest przeważnie parametrem systemu, który należy ustawić ręcznie. Rodzaj stosowanych cech jest zróżnicowany i zależy od konkretnej metody, ale najczęściej oparte są one na poziomie intensywności albo gradiencie obrazu.

Na wystarczająco oświetlonym obrazie źrenica jest obszarem znacznie ciemniejszym od otaczającej ją rogówki. Centrum źrenicy może zostać wyznaczone jako środek geometryczny obszaru uzyskanego po binaryzacji z odpowiednio dobranym progiem.

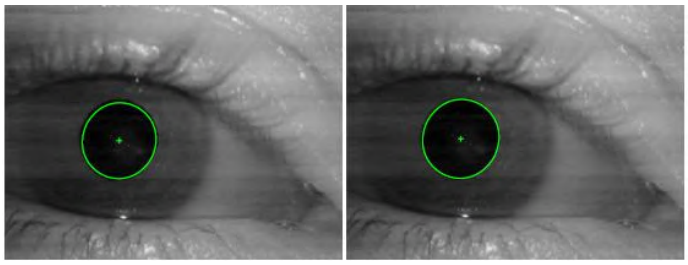
6.2 Algorytm "Starbust"

Algorytm "Starbust" [15] służy do detekcji oraz pomiaru pozycji źrenicy oraz refleksu światła podczerwonego odbitego od rogówki oka. Jest on częścią projektu open-source "openEyes". Celem tego projektu jest stworzenie łatwo dostępnego dla szerokiego kręgu użytkowników, taniego systemu do śledzenia wzroku. Starbust jest algorytmem hybrydowym wykorzystującym podejścia oparte na cechach oraz modelu. Kontur obrysowujący źrenice jest wyliczany przy pomocy estymacji elipsą zbioru punktów leżących na granicy między źrenicą a rogówką. Pozycja wyznaczonej elipsy jest poprawiana używając techniki opartej na modelu, który maksymalizuje stosunek między jasnością pikseli na zewnątrz i wewnątrz elipsy. Do detekcji kontury konieczna jest wstępna estymacja centrum elipsy. Może ona zostać wyznaczona jako lokacja z poprzedniej klatki, a dla pierwszej klatki można użyć inicjalizacji ręcznej lub po prostu można założyć, że środek oka pokrywa się ze środkiem źrenicy podczas inicjalizacji systemu. Punkty leżące na konturze są wyznaczane poprzez poszukiwane maksymalnej wartości zmiany intensywności obrazu wzdłuż promieni poprowadzonych ze środka oka (rys. 6.2).

Następnie dopasowywana jest elipsa do wyznaczonych punktów. Dla poprawy dokładności stosowany jest algorytm RANSAC [23] (Random Sample Consensus). Główna jego idea plaga na iteracyjnym losowym wyborze pięciu punktów ze zbioru wyznaczonego w poprzednim etapie. Ilość



Rysunek 6.2: Wyznaczanie punktów leżących na krawędzi źrenicy w algorytmie Startbust



Rysunek 6.3: Wynik poprawy lokacji elipsy przy użyciu modelu maksymalizującego ostrość krawędzi.

stosowanych punktów w każdej iteracji wynika z faktu, że aby wyznaczyć elipsę jest potrzebna znajomość pięciu punktów. Następnie dopasowywana jest elipsa do tego zbioru. Mając daną elipsę sprawdzane jest ile punktów z całego zbioru leży na jej krawędzi. Zakłada się, że punkt leży na krawędzi, jeśli jego odległość od niej nie przekracza pewnego eksperymentalnie dobranego progu. Do dalszego przetwarzania wybiera się największy zbiór punktów i do niego za pomocą metody najmniejszych kwadratów dopasowywana jest elipsa. Zastosowanie algorytmu RANSAC sprawia ze metoda jest znacznie bardziej niezawodna i odporna na zakłócenia niż w przypadku, kiedy do obliczania elipsy używa się całego zbioru krawędzi. Metoda najmniejszych kwadratów jest bardzo wrażliwa na występowanie zakłóconych danych, dlatego konieczne jest wyeliminowanie niepoprawnie odnalezionych punktów.

Ostatecznym etapem algorytmu startbust jest poprawa pozycji elipsy wyznaczonej przez algorytm RANSAC przy użyciu modelu maksymalizującego ostrość krawędzi konturu. Wynik działania tego etapu przedstawia rys. 6.3.

W pracy przetestowano część algorytmu odpowiedzialną za wyznaczanie środka źrenicy. Skorzystano z gotowej implementacji algorytmu zamieszczonej na stronie projektu OpenEyes [16]. Metoda ta jednak nie daje dobrych rezultatów, ponieważ w prezentowanym systemie rozdzielczość obrazu przedstawiającego obraz oka jest zbyt mała. Do prawidłowego działania algorytm startbust wymaga

obrazu oka w dużej rozdzielczości. Metoda ta jest dedykowana do zastosowań z wykorzystaniem kamery montowanej w niewielkiej odległości od oka.

6.3 Adaptacyjny dobór wartości progu binaryzacji

Aby można było ocenić kierunek, w którym są skierowane oczy, konieczne jest dokładne określenie środka źrenicy. Istnieje wiele skomplikowanych metod wyszukujących źrenic, ale większość z nich wymaga, aby obraz oka był w dużej rozdzielczości. Zakładając, że obraz z kamery jest uzyskiwany z niską rozdzielczością (640x480), większość z istniejących metod staje się bezużyteczna. Przykładem jest opisany powyżej algorytm startbust.

Kształt i wzajemne położenie tęczówki i źrenicy można w przybliżeniu opisać za pomocą dwóch współśrodkowych okręgów. Obliczając parametry okręgu opisującego kształt tęczówki można określić współrzędne środka źrenicy. Ze względu na swe właściwości tęczówka stanowi łatwiejszy w lokalizacji element ludzkiego oka w stosunku do źrenicy. Wynika to z faktu, iż granica tworzona przez tęczówkę z twardówką jest dużo bardziej kontrastowa niż tworzona ze źrenicą, w konsekwencji jest ona łatwiejsza do zlokalizowania.

W pracy zastosowano metodę podwójnej binaryzacji obrazu, która nie wymaga dużej rozdzielczości. Opiera się ona na fakcie, że źrenica wraz z tęczówką są znacznie ciemniejsza niż białko oka. Dobierając odpowiedni próg binaryzacji dokonuje się segmentacji obrazu. Binaryzacja jest bardzo wrażliwa na zmiany oświetlenia, co sprawia, że nie jest możliwe dobranie uniwersalnego progu binaryzacji tak, aby środek oka był zawsze wyznaczany prawidłowo, niezależnie od warunków oświetlenia oraz koloru tęczówki. Spowodowało to konieczność zaimplementowania procedury automatycznego doboru progu binaryzacji.

Aby zwiększyć niezawodność stosowane są dwa różne progi. Przy użyciu pierwszego progu wyznaczany jest w przybliżeniu środek źrenicy. Następnie podaje się obraz binaryzacji za pomocą drugiego progu, przy ograniczeniu, że obszar wyznaczony musi zawierać w sobie poprzednią estymacje. Dzięki takiemu podejściu wykorzystana jest niezawodność mniejszego progu oraz lepsza dokładność większego progu binaryzacji.

Po dokonaniu binaryzacji z wcześniej wyznaczonymi progami, następuje segmentacja obrazu. Często stosowanym sposobem określania punktów centralnych obszarów wydzielonych podczas segmentacji jest wyliczenie środka masy. Jednak światło odbite od oka może powodować, że obszary źrenicy będzie zawierał jasne refleksy, które zakłócają precyzyjne wyznaczenie centrum obszaru. Dlatego do estymacji zastosowano elipsę wyznaczoną za pomocą momentów centralnych konturu zbinearyzowanego obrazu oka.

Zaimplementowana metoda jest modyfikacją algorytmu opisanego w pracy [18].

Określanie środka obszaru za pomocą elipsy Po segmentacji obrazu wyznaczane są kontury wyodrębnionych obszarów. Następnie wyliczane są momenty centralne konturu. Momenty centralne uzyskuje się sumując wartości wszystkich pikseli zawartych na konturze. Ogólny wzór definiujący moment rzędu (p, q).

$$m_{p,q} = \sum_{i=1}^{n} I(x, y) x^p y^q$$

Dopasowanie elipsy (o środku w pkt (x, y), wysokości h i szerokości w) do obszaru dla którego zostały wyliczone momenty centralne określone jest następująco:

- Wprowadza się wartości pomocnicze

$$u_{0,0} = m_{0,0}$$

$$u_{1,0} = \frac{m_{1,0}}{m_{0,0}}$$

$$u_{0,1} = \frac{m_{0,1}}{m_{0,0}}$$

$$u_{1,1} = -\frac{m_{1,1} - m_{1,0}\frac{m_{0,1}}{m_{0,0}}}{m_{0,0}}$$

$$u_{2,0} = \frac{m_{2,0} - m_{1,0}\frac{m_{1,0}}{m_{0,0}}}{m_{0,0}}$$

$$u_{0,2} = \frac{m_{2,0} - m_{1,0}\frac{m_{1,0}}{m_{0,0}}}{m_{0,0}}$$

$$\Delta = \sqrt{4(u_{1,1})^2 + (u_{1,1} - u_{1,1})(u_{2,0} - u_{0,2})}$$

- Środek elipsy (x, y) wyznacza się następująco:

$$\begin{cases} x = & u_{1,0} \\ y = & u_{0,1} \end{cases}$$

- Rozmiar (wysokości h i szerokości w) określone są za pomocą równań:

$$\begin{cases} h = & \sqrt{2(u_{2,1} + u_{0,2} + \Delta)} \\ y = & \sqrt{2(u_{2,1} + u_{0,2} - \Delta)} \end{cases}$$

Opis działania algorytmu wyznaczania środka źrenicy

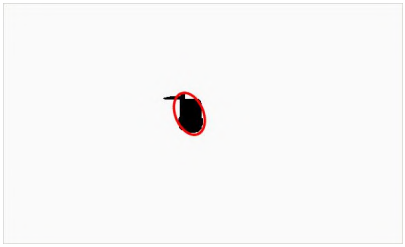
1. Wstępne wyznaczenie promienia tęczówki R. Stosunek odległości między oczami, a promieniem tęczówki jest w przybliżeniu stały u wszystkich ludzi i można go empirycznie wyznaczyć.
2. Obraz oka uzyskiwany podczas detekcji twarzy i oczu zostaje powiększony metodą gusowską do rozmiaru 100x100 pikseli. Dzięki temu uzyskuje się lepsze rezultaty dla bardzo małych rozdzielczości. Dokładność wyznaczenia środka źrenicy będzie większa niż jeden piksel.
3. Zastosowanie erozji w celu likwidacji refleksów.
4. Normalizacja obrazu, dzięki czemu można operować na bezwzględnych wartościach progu odcinania. Normalizacja sprawia, że najciemniejszy piksel ma zawsze wartość 0, a najjaśniejszy 255, bez względu na oświetlenie.
5. Dobór pierwszego progu binaryzacji. Iteracyjne sprawdzane są kolejne wartości od 0 do 40. Kryterium wyboru optymalnej wartości jest odległość od środka oka oraz kształt zbliżony do okręgu. Iteracja jest przerywana, gdy rozmiar osiągnie połowę wcześniej wyznaczonego R.
6. Dobór drugiego progu binaryzacji. Iteracyjne sprawdzane są kolejne wartości od 40 do 80. Kryterium wyboru optymalnej wartości jest odległość środka oka oraz kształt zbliżony do okręgu. Iteracja jest przerywana, gdy rozmiar osiągnie wcześniej wyznaczonego R.
7. Wyznaczenie środka źrenicy przy użyciu dobranych progów binaryzacji

Zaprezentowana metoda mimo swej prostoty jest bardzo skuteczna. Dobrze radzi sobie z małą rozdzielczością obrazu, jak i zmiennymi warunkami oświetlenia. Wynik działania algorytmu wyszukującego środek źrenicy został przedstawiony na rys. 6.4.

6.4 Kalibracja

Do wyznaczenia kierunku wzroku użytkownika wykorzystywane jest liniowe mapowanie homograficzne [4] wektora wzrokowego powstałego jako różnica odległości między aktualnym środkiem źrenicy a projekcją pozycji środka gałki ocznej na płaszczyznę kamery. Macierz mapowania określona jest jako H. Jest ona wyznaczana na podstawie zbioru relacji między wektorem wzroku, a punktem wyświetlanym na monitorze. Macierz H ma osiem stopni swobody, z czego wynika, że konieczna jest znajomość co najmniej czterech takich par. Dla zwiększenia dokładności, podczas procesu kalibracji jest rejestrowane dziesięć punktów. Wyznaczenie macierzy H oparte jest na metodzie najmniejszych kwadratów. Minimalizowana jest wyliczona różnica między wektorem wzroku a punktem na monitorze.

(a) Powiększony obraz po zastosowaniu erozji



(b) Binaryzacja przy użyciu pierwszego progu

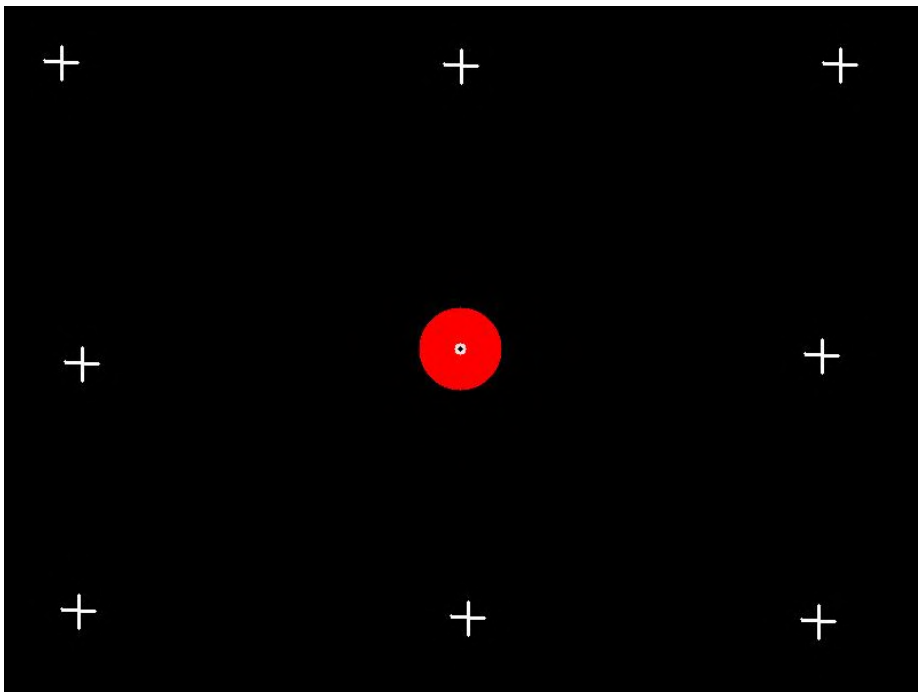
(c) Binaryzacja przy użyciu drugiego progu

(d) Elipsy wyznaczone na podstawie zbinearyzowanego obrazu

Rysunek 6.4: Wynik działania algorytmu wyszukującego środek źrenicy

Kalibracja aplikacji polega na wyznaczeniu zależności między wektorem wzroku a punktem fiksacji wyświetlanym na monitorze. Podczas procesu kalibracji użytkownik podąża wzrokiem za przemieszczający się punktem. Większa ilość wyświetlanych punktów sprawia, że kalibracja może zostać wyliczona w bardziej dokładny i niezawodny sposób. Jednak zbyt duża ilość punktów sprawia, że proces kalibracji może stać się nużący i użytkownik nie będzie staranie śledził punktów. Doświadczania z doborem różnej ilości punktów kalibracjach pokazały, że 10 punktów fiksacji wyświetlanych po 3 sekundy każdy rozmieszczonych równomiernie na ekranie daje zadowalające rezultaty, a proces kalibracji nie jest uciążliwy, gdyż trwa zaledwie 30 sekund. Oko ludzkie podczas fiksacji wykonuje nieustanny ruch w okuł śledzonego punktu. Aby zmniejszyć rozmiar tych mimowolnych ruchów gałki ocznej punkt, na którym się koncentruje użytkownik, zmienia rozmiar lub położenie. Sprawia to, że mimowolne odchyłki wzroku są znacznie mniejsze niż w przypadku statycznych punktów. Został przetestowany również schemat, w którym punkt kalibracyjny nieustannie jest w ruchu i przemieszcza się po całym ekranie. To podejście jednak nie daje najlepszych rezultatów. Znacznie lepszy efekt uzyskuje się, gdy punkt kalibracyjny jest wyświetlany w określonych lokacjach przez pewien czas. Dzięki temu można uśrednić wartość wektora wzroku dla danych lokacji i pominąć dane znacznie odstające od pozostałych. Przeważnie podczas pierwszej sekundy po zmianie położenia punktu rejestrowane jest duże odchylenie wektora wzroku od wartości średniej. Związane jest to z faktem, że wzrok podąża z pewnym opóźnieniem za wyświetlanym punktem. Pominięcie danych uzyskiwanych dla pierwszej sekundy poprawia rezultaty kalibracji.

W niektórych komercyjnych systemach do śledzenia wzroku o dużej dokładności, podczas kalibracji punkt o zmieniającym się rozmiarze jest zastępowany małymi obrazami lub napisami. Jest to czynnik dodatkowo stymulujący oko do koncentracji w konkretnym punkcie. Jednak zauważalną różnicę między podejściem z punktem o zmieniającym się rozmiarze można zauważyć dopiero przy bardzo dużej dokładności wyznaczania wektora wzroku. Rozwiązanie przedstawione w tej pracy opiera się na pozyskiwaniu obrazu z kamery internetowej. Taki obraz jest przeważnie na tyle zaszumiony, że uzyskiwana dokładność nie pozwala na zarejestrowanie jakiejkolwiek poprawy kalibracji przy wykorzystaniu obrazów czy małych napisów. Było to powodem pozostania przy prostym schemacie wyświetlania pojedynczego punktu o zmieniającym się rozmiarze (rys. 6.5) .



Rysunek 6.5: Okno kalibracji

Rozdział 7

Testy opracowanego rozwiązania

Na dokładność otrzymywanych rezultatów ma wpływ wiele czynników, takich jak poziom naświetlenia, odległość od kamery, jakoś obrazu z kamery, kąt widzenia obiektywu, jak również dokładny proces kalibracji. System opiera działanie na wykorzystaniu obrazu z kamery internetowej, który często jest bardzo zaszumiony. Występowanie szumu na obrazie są wywołane stosowaniem dużej czułości matrycy w przypadku słabego oświetlenia sceny. Duży wpływ na wielkość uzyskiwanego obrazu oka ma odległość od kamery oraz kąt widzenia obiektywu. W przypadku stosowania obiektywu o szerokim kącie widzenia rozdzielczość obrazu twarzy oraz oczu użytkownika jest mniejsza, co pogarsza wyniki działania.

Dokładność określania wzroku komercyjnych systemów podawana jest w stopniach, dzięki czemu nie jest ona zależna od wielkości monitora. Błąd wyrażony w stopniach określa średnie odchylenie między wyznaczoną wartością wektora wzroku a aktualnym punktem fiksacji wzroku.

Najdokładniejszy system do śledzenia wzroku oferuje firma SMI. W specyfikacji systemu maksymalna dokładność określona jest jako 0,3 stopnia. Taka dokładność uzyskiwana jest tylko w przypadku, kiedy użytkownik ma nieruchomą głowę oraz występują laboratoryjne warunki oświetlenia. Systemy korzystające ze światła podczerwonego są bardzo wrażliwe na występowanie światła dziennego, przez co do ich prawidłowego działania wymagane jest sztucznie oświetlone pomieszczenie.

Najlepszą dokładnością wśród systemów opartych wyłącznie na wykorzystaniu standardowej kamery charakteryzuje się praca [14]. Autorzy podczas testów otrzymali dokładność 3 stopi. Prezentowane w pracy podejście wymaga skomplikowanego procesu inicjalizacji modelu głowy dla indywidualnego użytkownika.

7.1 Opis przeprowadzanego eksperymentu

Testy zostały wykonane na grupie 8 osób. Celem testu finalnego programu było zbadanie precyzji określania punktu, na który spogląda użytkownik. Test dokładności wyznaczania aktualnego kierunku wzroku polega na określeniu różnicy między wyświetlanym na monitorze punktem a wyznaczaną przez program punktem fiksacji, na którym jest skierowany wzrok badanej osoby. Do tego celu badana osoba była proszona o podążanie wzrokiem za przemieszczającym się punktem między określonymi lokacjami. Wydruk monitora wraz z lokacjami wyświetlanych punktów podczas badania dokładności algorytmu przedstawia rys 6.5. Błąd wyznaczania przez program wektora wzroku jest w przybliżeniu równy różnicy między wyznaczonym punktem przez algorytmy a wyświetlanym na monitorze.

Każdy z punktów był wyświetlany przez cztery sekundy. Do dalszej analizy pomijane były dane rejestrowane podczas wyświetlania przez pierwszą sekundę dla każdego punktu. Wzrok ludzki charakteryzuje się pewną inercją, przez co rejestrowane dane podczas szybkiego ruchu punktu są nieco opóźnione i przesunięte na osi czasu. Dzięki pominięciu początkowej fazy wyświetlania, zjawisko to nie ma wpływu na wynik przeprowadzanej analizy dokładności. Testy przeprowadzane dla ciągle poruszającego się punktu są znacznie mniej wymierne niż w przypadku punktów statycznych. Szybkość reakcji wzroku na zmianę położenia punktu jest zależna od refleksu badanej osoby i wymaga stosowania skomplikowanej analizy. Stosowanie statycznych punków upraszcza oraz poprawia dokładność rezultatów.

Testy przeprowadzono na monitorze o rozmiarze 17 cali. Do rejestracji obrazu użyto standardowej kamery internetowej (Logitech pro9000) zwracającej obraz w rozdzielczości 640x480 pikseli z częstotliwością 15 klatek na sekundę. Badana osoba znajdowała się w przybliżeniu w odległości 50 cm od centrum monitora.

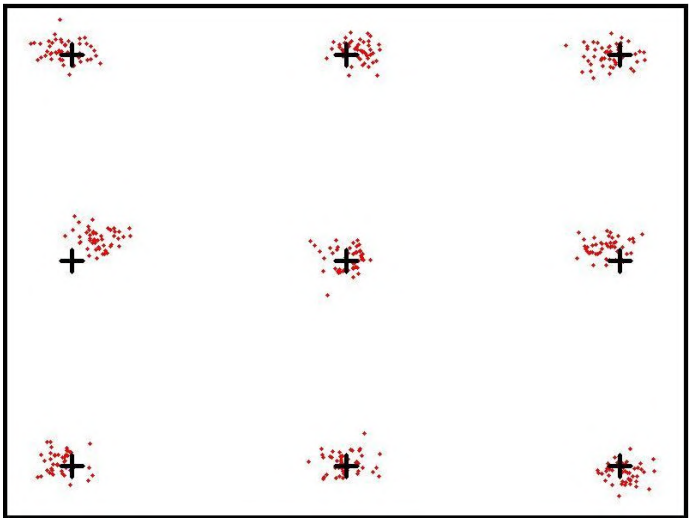
7.2 Wyniki

Dokładność opracowanego rozwiązania została zbadana wyliczając uśredniony błąd między aktualnie wyświetlanym punktem na monitorze a wyznaczonym przez program. Zbiorcze rezultaty zostały zaprezentowane w tabeli 7.1. Aby możliwe było wyznaczenie błędu w stopniach, konieczna była znajomość aktualnej odległości użytkownika od kamery. Dla uproszczenia zostało założone, że ta odległość wynosi 50cm dla wszystkich badanych osób. Jest to standardowa odległość, w jakiej znajduje się głowa użytkownika podczas pracy przy komputerze wyposażonym w monitor 17 calowy.

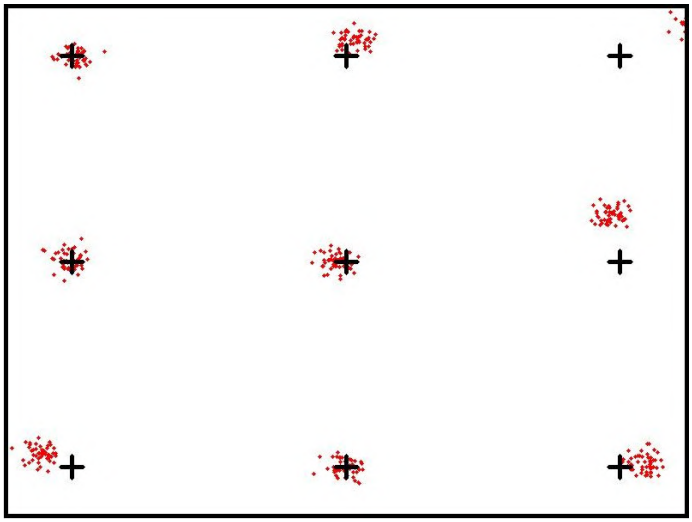
Średni błąd oceny wzroku podczas przeprowadzania testów wyniósł 1,5 stopnia. Jest to wystarczająca dokładność do szerokiej gamy zastosowań. Wyniki testów zostały przedstawiona na rys. 7.1 - 7.8.

Tester	średnie odchylenie w stosunku do szerokości monitora (%)	błąd w stopniach
tester 1	3.3	1.1
tester 2	4.6	1.6
tester 3	3.9	1.3
tester 4	3.5	1.2
tester 5	4.2	1.5
tester 6	5.5	1.9
tester 7	6.0	2.0
tester 8	2.8	0.9

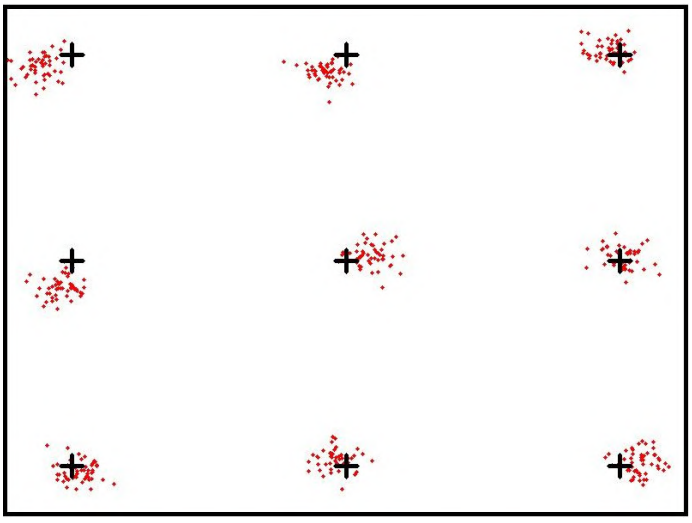
Tablica 7.1: Wyniki testów



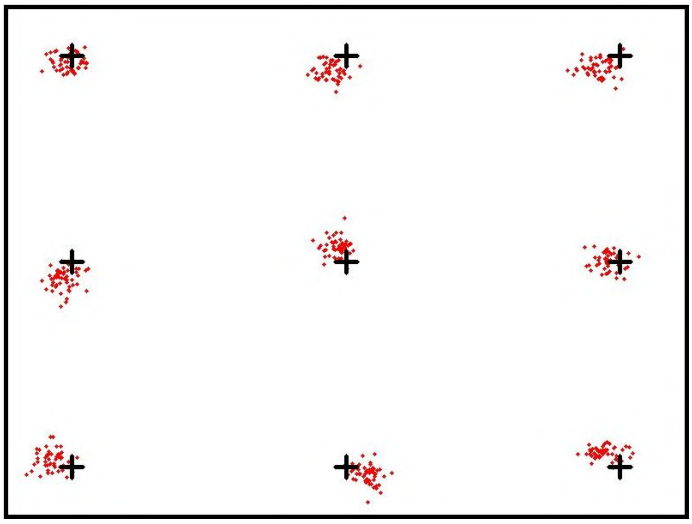
Rysunek 7.1: Tester 1



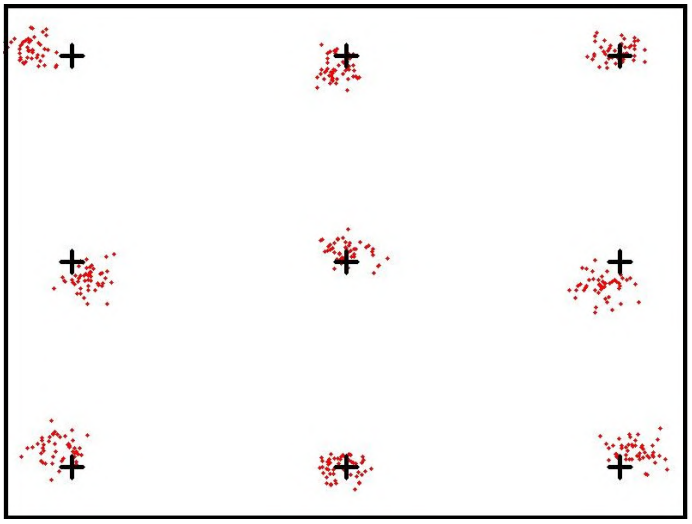
Rysunek 7.2: Tester 2



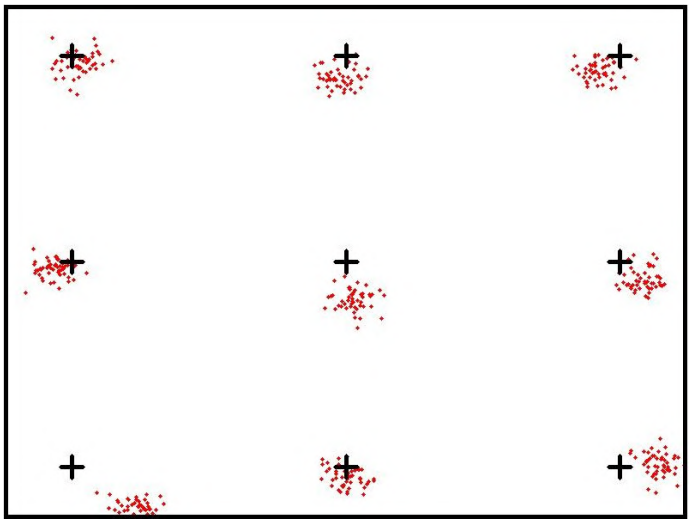
Rysunek 7.3: Tester 3



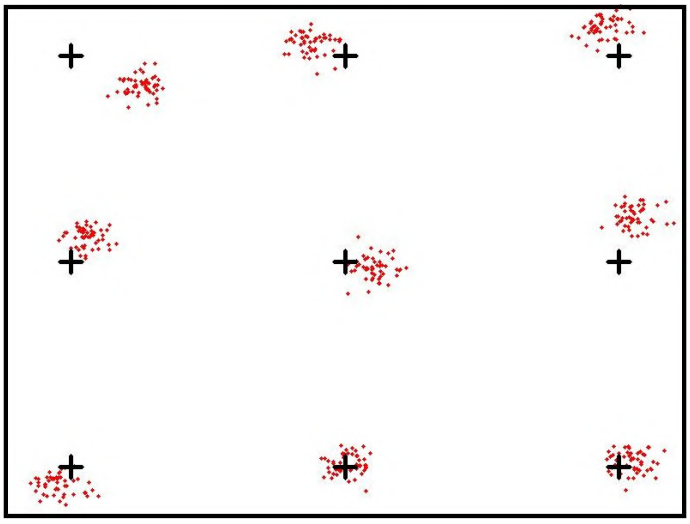
Rysunek 7.4: Tester 4



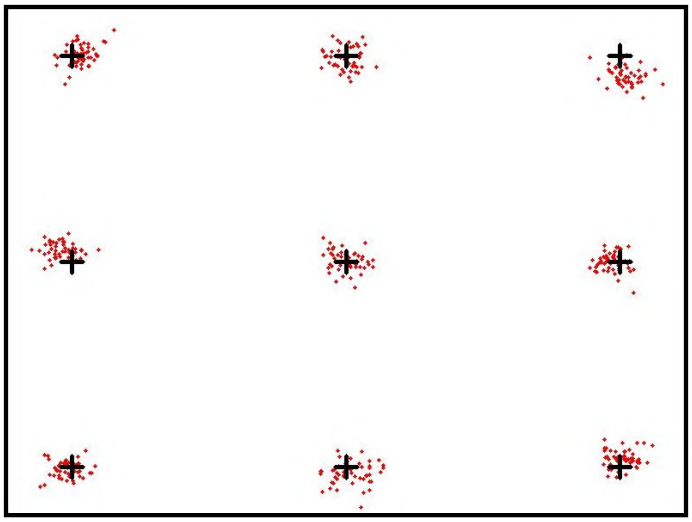
Rysunek 7.5: Tester 5



Rysunek 7.6: Tester 6



Rysunek 7.7: Tester 7



Rysunek 7.8: Tester 8

7.3 Przykłady zastosowania opracowanego rozwiązania

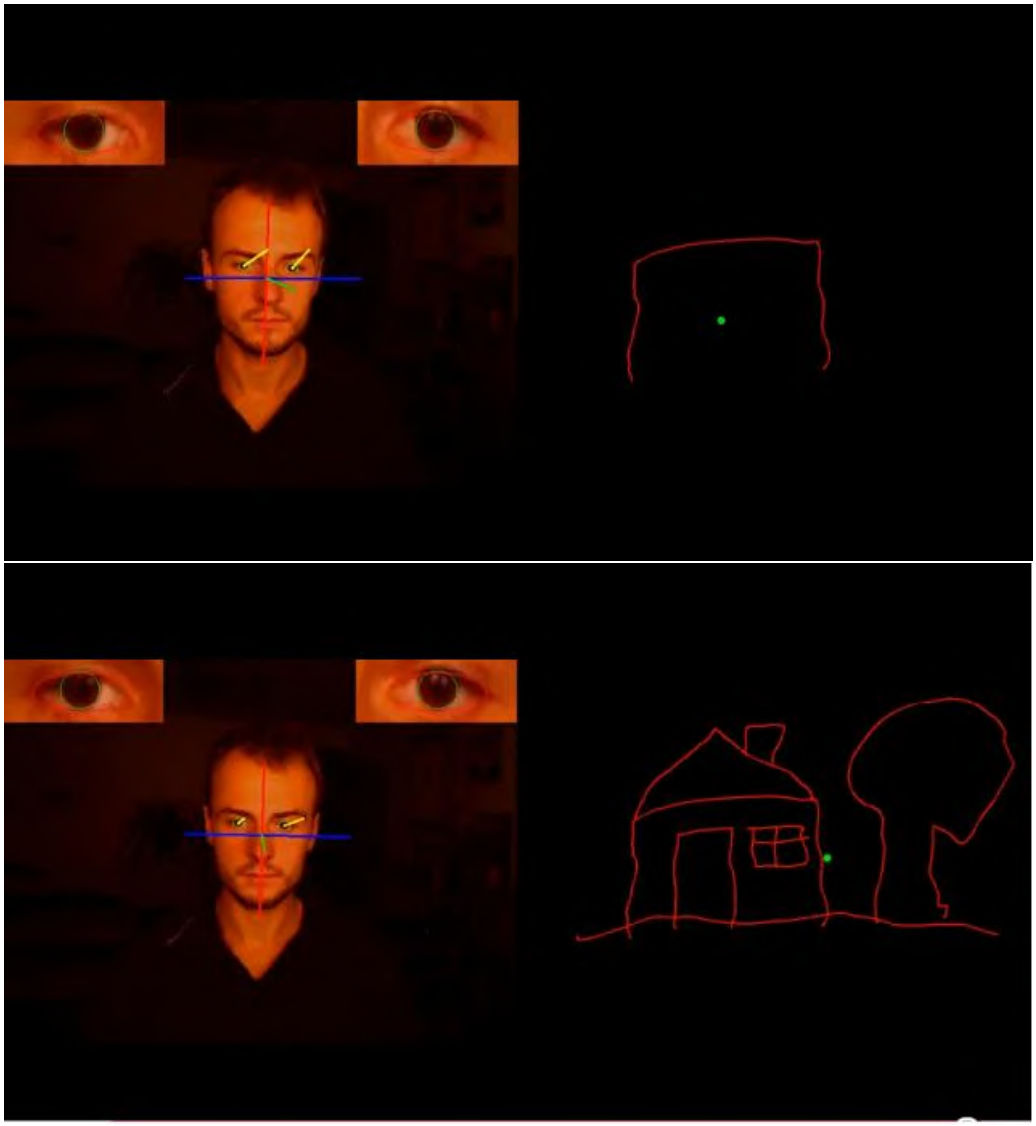
7.3.1 Sterowanie kursorem za pomocą ruchu głowy

Opracowana metoda do śledzenia aktualnej pozycji 3D głowy ma cały szereg zastosowań. Przykładem może być wykorzystanie aktualnego skrętu głowy wokół osi X oraz Y do kontroli położenia kursora myszy. Do demonstracji oraz testów dokładności stworzona została aplikacja umożliwiająca rysowania za pomocą ruchu głowy. Wynik działania przedstawia rys. 7.9.

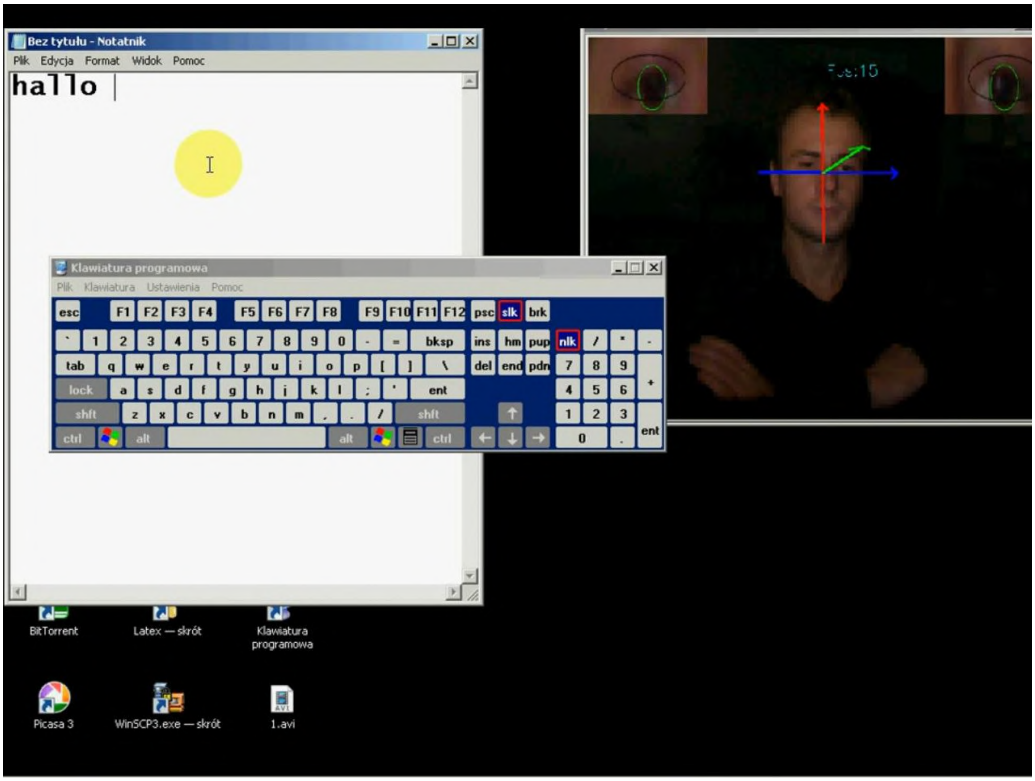
Wykorzystanie znajomości aktualnej pozycji głowy do kontroli pozycji kursora umożliwia kontrole komputera za pomocą ruchu głowy. Przykład zastosowania takiego rozwiązania do pisania przy użyciu klawiatury systemowej Windows przedstawiony jest na rys. 7.10.

7.3.2 Badanie punktu koncentracji

Zwracany przez program aktualny punkt fiksacji wzroku może zostać wykorzystany do badań marketingowych. Przykład zastosowania programu do badania aktualnego punktu koncentracji użytkownika podczas oglądania reklamy telewizyjnej został przedstawiony na rys. 7.11. Czerwony okrąg reprezentuje aktualny punkt koncentracji, a poprzedzająca go linia wyznacza ścieżkę wzrokową z kilku poprzednich klatek filmu.



Rysunek 7.9: Rysowania za pomocą ruchu głowy



Rysunek 7.10: Pisanie za pomocą ruchu głowy

Rysunek 7.11: Badanie reklamy telewizyjnej

Podsumowanie

Śledzenie wzroku jest skomplikowanym zadaniem. Istnieje wiele metod umożliwiających badanie aktualnego punktu fiksacji użytkownika. W dziedzinie śledzenia ludzkiego wzroku zostało wykonanych wiele badań, dzięki czemu istnieją liczne rozwiązania znacząco różniące się od siebie. Wybór optymalnej metody zależy od celu, w jakim są wykonywane badania. Przeznaczenie danego systemu określa wymaganą dokładność, rozdzielczość, częstotliwość pomiarów, łatwość i wygodę używania, a także cenę.

Celem pracy magisterskiej było opracowanie i implementacja systemu śledzącego oczy użytkownika komputera w czasie rzeczywistym za pomocą kamery internetowej. Najistotniejszym założeniem pracy postawionym przez autora było opracowanie uniwersalnej aplikacji niewymagającej żadnej specjalnej konfiguracji sprzętowej do działania. Idealne rozwiązanie powinno składać się z komputera osobistego oraz pojedynczej uniwersalnej kamery internetowej.

Na rynku istnieje wiele komercyjnych systemów umożliwiających śledzenie wzroku. Największą wadą istniejących rozwiązań jest konieczność posiadania specjalnego sprzętu, zaczynając od dedykowanych urządzeń montowanych na głowie do kamer działających na podczerwień. Łatwo dostrzec zalety systemu, który by nie wymagał specjalistycznego sprzętu. Nie istnieje jednak komercyjny system, którego działanie opierałoby się na zastosowaniu standardowej kamery. Istnieje jednak duża ilość ośrodków badawczych, w których są prowadzone intensywne badania w celu stworzenia prototypu takiego rozwiązania.

Obecnie jedynym dostępnym projektem zajmującym się badaniem wzroku za pomocą standardowej kamery jest projekt opensource opengazer [22]. Jego główną wadą jest założenie, że głowa użytkownika jest zupełnie nieruchoma podczas działania programu. Nawet niewielki ruch sprawia, że konieczna jest ponowna kalibracja systemu. Dokładność tego systemu pozostawia także wiele do życzenia i znacząco odbiega od komercyjnych rozwiązań. OpenGazer może być wykorzystywany, np. przez ludzi upośledzonych ruchowo, do komunikacji z komputerem. Możliwe jest wykorzystanie dedykowanego oprogramowania do pisania za pomocą ruchu gałki ocznej. Przykładem takiego programu może być Dasher [31].

W pracy zostało zaprezentowane rozwiązanie opierające swoje działanie na rejestracji obrazu

twarzy przy użyciu standardowej kamery internetowej. Obraz jest pozyskiwany w widzialnym spektrum światła. Nie jest wymagane stosowanie dedykowanej kamery ani źródeł światła podczerwonego. Dzięki zastosowaniu zaawansowanych metod przetwarzania obrazów opracowany program jest w stanie określić aktualny punkt koncentracji użytkownika komputera. Zastosowane algorytmy umożliwiają ocenę aktualnego punktu fiksacji wzroku ze stosunkowo dużą dokładnością. Dokładność wyznaczania aktualnej pozycji oka nie jest ograniczona rozdzielczością kamery, ponieważ zastosowano metodę określającą pozycję z sub pikselową precyzją.

Rozwiązanie ma szeroką gamę zastosowań. Przykładem wykorzystania systemu mogą być aplikacje służące do interakcji człowiek-komputer (HCI), jak również aplikacje badające użyteczność. Dokładność opracowanego rozwiązania umożliwia generacje map ciepła na podstawie zwróconych wyników.

Opracowany program umożliwia śledzenie wzroku w czasie rzeczywistym. Możliwy jest swobodny ruch głowy użytkownika, dzięki zastosowaniu algorytmów wyznaczających aktualną pozycję oraz skręty głowy. Jest to bardzo istotne, gdyż systemy zakładające, że głowa pozostaje zupełnie nieruchoma, wymagają ponownej kalibracji w przypadku nawet niewielkiego jej ruchu.

Duży nacisk położono na niezawodność rozwiązania. Zastosowano szereg algorytmów eliminujących wpływ zakłóconych danych. Algorytm śledzenia pozycji głowy jest odporny na przesłanianie twarzy nieznanymi obiektami. Zastosowanie klatek referencyjnych, w śledzeniu głowy sprawia, że nie występuje zjawisko akumulacji błędu. Jest to bardzo istotne, ponieważ bez takiego mechanizmu, w przypadku dłuższego działania programu, wyznaczana pozycja głowy może znacząco odbiegać od prawidłowej.

Zastosowane metody są w pełni adaptacyjne i dostosowują się do warunków otoczenia. Procedura automatycznego doboru progu binaryzacji sprawia, że cześć programu wykorzystująca binaryzację obrazu nie jest wrażliwa na zmiany oświetlenia.

Niewątpliwą zaletą prezentowanego rozwiązania jest w pełni automatyczna inicjalizacja. Program automatycznie dobiera wszystkie parametry konieczne do działania, nie wymagając od użytkownika żadnej specjalistycznej wiedzy do skonfigurowania systemu. Po uruchomieniu programu, wykrywana jest twarz użytkownika. Następnie tworzony jest model głowy. Wyznaczany jest zbiór cech charakterystycznych dla danego użytkownika, które następnie są śledzone przy użyciu przepływu optycznego. Po zainicjalizowaniu systemu następuje proces kalibracji. Jedyna wymagana ingerencja końcowego użytkownika, to konieczność prawidłowego przeprowadzenia kalibracji. Polega to na konieczności podążania wzrokiem za przemieszczającym się punktem po monitorze. Proces ten nie jest uciążliwy i trwa zaledwie 30 sekund. Przebieg procesu kalibracji jest w pełni bazowany na rozwiązaniach stosowanych w systemach komercyjnych, jak Tobii czy SMI.

Podstawową zaletą stworzonego programu jest prostota konfiguracji sprzętowej. System bazuje wyłącznie na uniwersalnej kamerze internetowej. Nie jest wymagany nawet konkretny model kamery. Parametry takie jak rozdzielczość oraz szerokość konta widzenia kamery mają wpływ na

dokładność zwracanych rezultatów, ale nie są stawiane żadne restrykcyjne wymagania. System z powodzeniem może być stosowany nawet przy użyciu kamer wbudowanych w laptopach, których rozdzielczość oraz jakość obrazu przeważnie pozostawia wiele do życzenia.

Bibliografia

[1] http://www.diku.dk/hjemmesider/ansatte/panic/eyegaze/article.html

[2] http://www.metrovision.fr

[3] Dinesh Kumar and Eric Poole. Classification of EOG for human computer interface. In Proceedings of the Second Joint EMBS/BMES Conference, Houston, TX, USA, October 2002.

[4] Duchowski, A.T., Eye Tracking Methodology: Theory and Practice, Springer2007.

[5] Carlos Hitoshi Morimoto, Dave Koons, Arnon Amir, and Myron Flickner. Pupil detection and tracking using multiple light sources. Image Vision Comput., 2000.

[6] http://www.eti.pg.gda.pl/katedry/kiw/dydaktyka/Widzenie_Komputerowe //Automatyczna_lokalizacja_i_sledzenie_obiektow2.pdf

[7] Paul Viola, Michael Jones "Robust Real-time Object Detection", Vancouver, Canada, July 13, 2001

[8] http://sourceforge.net/projects/opencvlibrary/

[9] Takahiro Ishikawa, Simon Baker, Iain Matthews, and Takeo Kanade. Passive driver gaze tracking with active appearance models. In Proceedings of the 11th World Congress on Intelligent Transportation Systems, October 2004.

[10] Denis Leimberg and Martin Vester-Christensen, "Eye Tracking", Master's Thesis, Technical University of Denmark, 2005.

[11] http://www2.imm.dtu.dk/~aam/aamapi/

[12] C. Harris and M. Stephens, "A combined corner and edge detector," Proceedings of the 4th Alvey Vision Conference (pp. 147–151), 1988

[13] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," Proceedings of the European Conference on Computer Vision (pp. 335–343), 1992.

[14] Takahiro Ishikawa, Simon Baker, Iain Matthews, and Takeo Kanade. Passive driver gaze tracking with active appearance models. In Proceedings of the 11th World Congress on Intelligent Transportation Systems, October 2004.

[15] Li, D., Winfield, D., & Parkhurst, D.J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. Proceedings of the 2nd IEEE CVPR Workshop on Vision for Human-Computer Interaction, San Diego, USA, 2005.

[16] http://thirtysixthspan.com/openEyes/

[17] http://mozyrko.pl/category/eye-tracking/

[18] Denis Leimberg and Martin Vester-Christensen, "Eye Tracking", Master's Thesis, Technical University of Denmark, 2005.

[19] Luca Vacchetti, Vincent Lepetit, Pascal Fua "Fusing Online and Offline Information for Stable 3D Tracking in Real-Time" Computer Vision Laboratory Swiss Federal Institute of Technology (EPFL) 1015 Lausanne, Switzerland

[20] "D5.2 Report on New Approaches to Eye Tracking" COGAIN Communication by Gaze Interaction 2006

[21] David B¨ack "Neural Network Gaze Tracking using Web Camera"

[22] http://www.inference.phy.cam.ac.uk/opengazer/

[23] Richard Hartley, Andrew Zisserman "Multiple View Geometry in Computer Vision" Second Edition Australian National University 2003, Australia

[24] Marco La Cascia, Stan Sclaroff"Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 4, APRIL 2000

[25] http://www.tobii.com

[26] http://www.smivision.com/

[27] http://www.cogain.org/

[28] Giorgio Panin, Alois Knoll "Fully Automatic Real-Time 3D Object Tracking using Active Contour and Appearance Models" Technical University of Munich

[29] Gary Bradski, Adrian Kaehler "Learning OpenCV" O'Reilly 2008

[30] Jaewon Sung, Takeo Kanade, Daijin Kim "Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models" 2008

[31] http://www.inference.phy.cam.ac.uk/dasher/