

Paweł Zięba^{*}, Mateusz Czapliński^{*}, Mateusz Biliński^{*}

PRECYZYJNA ANALIZA RUCHU GAŁEK OCZNYCH

1. WSTĘP

Gwałtowny wzrost mocy obliczeniowej komputerów ma duży wpływ na technologię w dzisiejszych czasach, dając możliwości stosowania komputerów w wielu nowych dziedzinach. Niniejsza praca prezentuje wykorzystanie wspomnianych możliwości w zastosowaniu bardzo praktycznym – sterowaniu komputerem za pomocą wzroku. Rozwiązanie takie jest bardzo potrzebne i można je wykorzystać w wielu dziedzinach: pomocy osobom niepełnosprawnym, marketingu, komputerowej rozrywce.

Prezentowany system jest rozwinięciem wcześniej opublikowanych pomysłów [1]. Zasada jego działania opiera się na analizie obrazów oka ludzkiego, wykonanych przez kamerę podłączoną do komputera. Zestaw 4 diod LED, umieszczonych w narożnikach ekranu komputera, stwarza na oku refleksy, które są wykrywane na zarejestrowanym obrazie i pozwalają wyznaczyć układ odniesienia. Naprzemiennie z tymi diodami, system zapala piątą diodę, umieszczoną w osi obiektywu kamery (zob. rys. 1). Tak umieszczone światło wywołuje na obrazie z kamery tzw. „efekt czerwonych oczu”, który silnie wzmacnia widoczność źrenicy oka. Dzięki temu możliwe jest znalezienie położenia środka źrenicy, którego nałożenie na znaleziony w poprzednim etapie układ odniesienia pozwala wyznaczyć punkt na ekranie, w który wpatruje się użytkownik komputera.

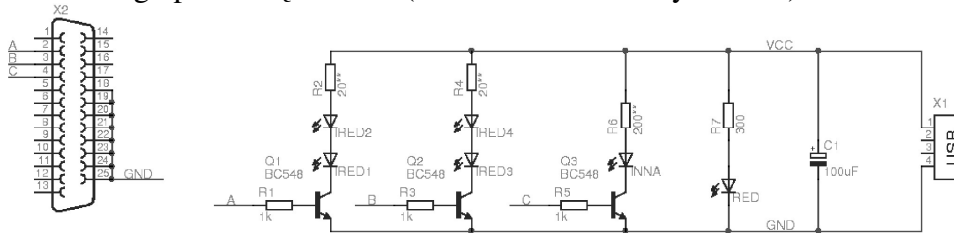
2. SPRZĘT I POBIERANIE OBRAZU

^{*} student, Koło Naukowe Informatyków „Kernel”, Wydział Fizyki i Informatyki Stosowanej AGH



Rys. 1. Sposób instalacji diod w narożnikach ekranu oraz w osi obiektywu kamery.

Część sprzętowa omawianego systemu składa się z kamery oraz 5 czerwonych diod LCD (cztery z nich zamocowane w rogach ekranu, jedna w osi obiektywu kamery) połączonych z komputerem za pomocą kontrolera sterowanego przez złącze LPT (schemat układu na rysunku 2).



Rys. 2. Schemat układu pośredniczącego w sterowaniu zapalaniem diod. Linie A oraz B sterują diodami w narożnikach ekranu (IRED1-4). Linia C steruje diodą umieszczoną w osi obiektywu kamery (INNA). Dioda RED informuje czy układ jest poprawnie zasilany. Kondensator C1 służy do filtrowania napięcia.

3. ANALIZA OBRAZU, ROZPOZNAWANIE KSZTAŁTÓW

Aby móc na podstawie uzyskanych z kamery obrazów wyznaczyć kierunek patrzenia użytkownika, konieczne jest przeanalizowanie obrazu. Celem tego etapu jest znalezienie na obrazie punktów, w których odbija się światło diod LED z narożników ekranu oraz wyznaczenie położenia środka obrazu źrenicy oka użytkownika. Analizę prowadzi się na podstawie dwóch obrazów z kamery – jeden z nich wykonuje się przy włączonych diodach umieszczonych w narożnikach ekranu, drugi przy włączonej diodzie leżącej w osi obiektywu kamery.

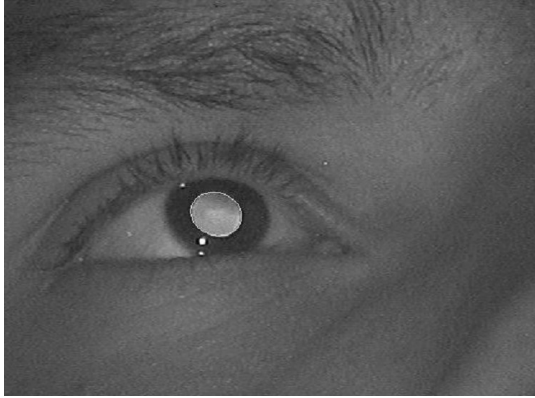
Kluczowym problemem jest tu wyznaczenie środka źrenicy oka. W tym celu niezbędne jest znalezienie elipsy, będącej jej brzegiem. Do wykonania tej procedury w opisywanym systemie stosowany jest algorytm genetyczny przedstawiony przez P. Ziębę [2].

Algorytm ten wymaga przeprowadzenia wstępnej obróbki danych wejściowych (ang. *feature detection*), której celem jest wyodrębnienie na obrazie szukanych obiektów. Wykonywane są następujące operacje: oba wejściowe obrazy poddawane są rozmyciu (redukcja szumów), następnie odejmowane od siebie (wyodrębnienie różnic wynikających z włączenia innych diod); dalej podlegają odcięciu (ang. *threshold*) w celu ponownej redukcji szumów; wreszcie aplikowany jest filtr Canny'ego [3], służący do wykrywania krawędzi.

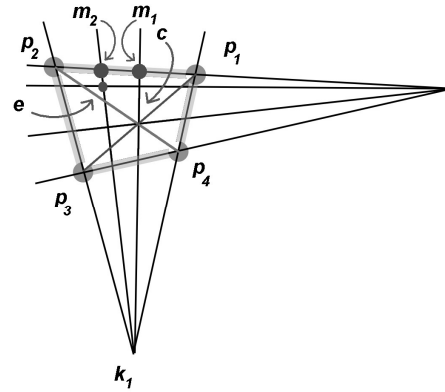
Algorytm wykorzystywany do wyszukania obróconej elipsy, będącej krawędzią zarejestrowanego obrazu źrenicy, należy do kategorii tzw. *algorytmów genetycznych*. Są to algorytmy heurystyczne, których celem jest optymalizacja zadanej funkcji (*funkcja dostosowania*). Opierają się one na modelowaniu rozwoju populacji osobników podobnym jak w ewolucji gatunków w świecie rzeczywistym.

W wykorzystywanym algorytmie genetycznym *osobnikiem* jest pojedyncza obrócona elipsa, której *genotyp* stanowi jej 5 parametrów (współrzędne środka, długości obu osi, kąt nachylenia osi wielkiej elipsy względem osi x). *Funkcja dostosowania* dla osobnika wyliczana jest na podstawie ilości białych pikseli obrazu (białym kolorem oznaczono piksele należące do krawędzi wykrytych przez filtr Canny'ego) leżących w pobliżu elipsy, opisywanej w/w parametrami, oraz bliskości każdego z tych pikseli do niej.

W opisywanym przypadku populacji elips, „dobór naturalny” (tzw. *selekcja*) jest niedeterministyczny, przy czym prawdopodobieństwo przetrwania osobnika jest proporcjonalne do jego wartości funkcji dostosowania. *Krzyżowanie* dwóch osobników prowadzi do powstania nowego osobnika, którego każdy współczynnik ma wartość losowaną z rozkładem jednorodnym z przedziału $[a, b]$, gdzie a, b są wartościami tego współczynnika u osobników krzyżowanych. Wreszcie, aby zapobiec degeneracji populacji wprowadzane są do genotypu losowych osobników niedeterministyczne zaburzenia. Po odpowiednim dostrojeniu, algorytm okazuje się bardzo dobrze sobie radzić z odnajdywaniem elipsy na testowanych obrazach.



Rys. 3. Obraz oka, z zaznaczoną elipsą źrenicy znalezioną przez algorytm genetyczny.



Rys. 4. Charakterystyczne punkty na oku przed dokonaniem mapowania.

4. MAPOWANIE NA EKRAN

W ostatnim etapie, na podstawie współrzędnych czterech punktów $p_i = (x_i, y_i)$ (gdzie $i = 1,2,3,4$) obrazu z kamery, w których odbijają się diody LED z narożników ekranu oraz współrzędnych punktu $e = (e_x, e_y)$, w którym znajduje się obraz środka źrenicy oka, wyznaczane są współrzędne ekranowe (w pikselach) punktu $s = (s_x, s_y)$, w który wpatrywał się użytkownik.

Bazując na wcześniej wspomnianej metodzie [1], algorytm wykorzystywany w tym module opiera się na wielkości zwanej *cross ratio*. Dla współliniowych punktów A, B, C i D współczynnikiem *cross ratio*, oznaczaną jako $(ABCD)$, nazywamy stosunek ich odległości wyrażony wzorem

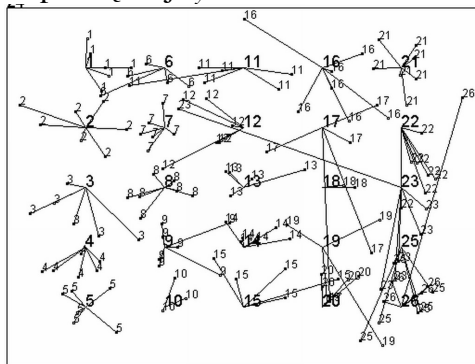
$$(ABCD) = \frac{DB \cdot CA}{DA \cdot CB}. \quad (1)$$

Współczynnik *cross ratio* ma kluczową cechę, która pozwala na jego zastosowanie w opisywanym problemie, a mianowicie jest *niezmiennikiem przekształceń perspektywicznych*. W przedstawianym algorytmie założono, że oko ludzkie ma płaską powierzchnię (dokładniej: że znalezione przez nas punkty p_i oraz e są współpłaszczyznowe). W takiej sytuacji, jeśli znajdzie się w płaszczyźnie oka cztery współliniowe punkty, można wyznaczyć ich *cross ratio*; następnie, znając punkty na monitorze komputera odpowiadające trzem spośród punktów z oka, można wyznaczyć punkt na ekranie odpowiadający czwartemu.

W omawianym zagadnieniu powyższą technikę należy zastosować dwukrotnie, osobno dla każdej współrzędnej ekranowej (poziomej oraz pionowej). Aby mieć komplet czterech punktów w obu przypadkach, konieczne jest znalezienie:

- punktu pomocniczego c , w którym przecinają się odcinki $\overline{p_1p_3}$ i $\overline{p_2p_4}$ (punkt ten odpowiada miejscu, gdzie odbijałaby się dioda umieszczona na środku ekranu komputera),
- przy poszukiwaniu współrzędnej poziomej (dla współrzędnej pionowej analogicznie):
 - punktu pomocniczego k_1 przecięcia przedłużeń $\overline{p_1p_4}$ i $\overline{p_1p_3}$ (zob. rys. 4),
 - punktów m_1 i m_2 wyznaczonych przez przecięcie odpowiednio $\overline{ck_1}$ i $\overline{ek_1}$ z odcinkiem $\overline{p_1p_2}$.

Współczynnik *cross ratio* wyznaczony przez p_1 , p_2 , m_1 i m_2 należy przyjąć za równy współczynnikowi wyznaczonemu przez górne narożniki ekranu, środek górnej krawędzi ekranu (odpowiada m_1) oraz rzut szukanego punktu s na górną krawędź ekranu – co bezpośrednio przekłada się na równanie, z którego można wyznaczyć s_x . Podobnie należy postąpić dla współrzędnej s_y .



Rys. 5. Przykładowe wyniki działania programu. Prostokąt oznacza obszar ekranu. Duże cyfry oznaczają położenie punktów znalezionych na ekranie, małe cyfry – położenie punktów wyznaczone przez system.

W rzeczywistości otrzymane w ten sposób wyniki z różnych przyczyn (np. przyjętych uproszczeń) nie nadają się bezpośrednio do zastosowania. Dlatego koniecznym okazuje się przeprowadzenie dwóch dodatkowych kroków. Pierwszy zabieg, kalibracja, polega na wyznaczeniu czterech punktów w pobliżu narożników ekranu, o znanych współrzędnych ekranowych. Następnie należy zebrać próbkę kilku-kilkunastu wyników uzyskiwanych w momencie wpatrywania się użytkownika w każdy z tych punktów. Po uśrednieniu uzyskuje się cztery punkty odniesienia, które pozwalają zbudować drugie mapowanie, analogiczne do opisanego

powyżej, które znacznie poprawia skuteczność systemu. Kalibrację wykonuje się raz na sesję. Drugim krokiem jest oczyszczenie danych z błędnych wyników generowanych w przypadku mrugnięcia oka użytkownika. Skuteczną okazuje się tu bardzo prosta metoda, polegająca na odfiltrowaniu wyników, które po obu mapowaniach trafiają poza obszar ekranu komputera.

5. PODSUMOWANIE

Opisany system, po skonstruowaniu przez autorów, okazał się działać skutecznie i spełniać postawione założenia (przykładowe wyniki zaprezentowane są na rysunku 5). Wykorzystanie stworzonego systemu poza warunkami laboratoryjnymi wymaga jednak jeszcze pewnych udoskonaleń. Sporym problemem w praktyce okazało się użycie diod LED ze spektrum światła widzialnego – znacznym udogodnieniem byłoby skorzystanie z diod podczerwonych, wymagają one jednak czulszego układu rejestrującego kamery. Warto przeanalizować również możliwości dalszej poprawy trafności, na przykład poprzez uśrednienie kilku wyników.

Referat pt. „Precyzyjna analiza ruchu gałek ocznych” – autorstwa Pawła Zięby, Mateusza Czaplńskiego oraz Mateusza Bilińskiego, został wygłoszony na XLIV Sesji Studenckich Kół Naukowych Pionu Hutniczego AGH w Sekcji Informatyki Stosowanej oraz wyróżniony I nagrodą. Opiekun naukowy referatu: dr hab. inż. Witold Dzwiniel.

LITERATURA

- [1] Dong Hyun Yoo: Non-contact Eye Gaze Estimation System using Robust Feature Extraction and Mapping of Corneal Reflections, Korea Advanced Institute of Science and Technology, 2002 (niepublikowana).
- [2] Zięba P.: Wyszukiwanie elips za pomocą algorytmów genetycznych, Wydział Fizyki i Informatyki Stosowanej AGH, Kraków, 2007 (praca inżynierska, niepublikowana).
- [3] Canny J.: A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, zesz. 8/1986.