AKADEMIA GÓRNICZO-HUTNICZA
im. St. Staszica w Krakowie
WEAIiE, Katedra Automatyki
Laboratorium Biocybernetyki

Przedmiot: Interfejsy multimodalne. PRMM113_MOUTHDET

Temat projektu: Detekcja ruchów ust.

Spis treści:

1. ABSTRAKT..... 2
2. WSTĘP..... 3
3. KONCEPCJA PROPONOWANEGO ROZWIĄZANIA..... 4
4. REZULTATY I WNIOSKI..... 6
5. PODSUMOWANIE..... 8
6. LITERATURA..... 9
7. DODATEK A: OPIS OPRACOWANYCH NARZĘDZI I METODY POSTĘPOWANIA..... 10
8. DODATEK B: REALIZACJA PROPONOWANEGO ROZWIĄZANIA..... 11
9. DODATEK C. OPIS INFORMATYCZNY PROCEDUR..... 12
10. DODATEK D. SPIS ZAWARTOŚCI DOŁĄCZONEJ PŁYTY CD..... 14

Wykonali: Wojciech Daniło, Mateusz Ziarko
III rok *Inf.Stos*
konsultant: *dr inż. Jaromir Przybyło*

Wersja 1.0

Kraków, maj 2011.

1. Abstrakt

Głównym celem realizowanego projektu było stworzenie za pomocą programu MatLab systemu detekcji ruchów ust. W tym celu został wykorzystany prosty algorytm detekcji składający się m.in. z wyznaczania MHI, filtracji, operacji morfologicznych oraz etykietowania. Działaniu algorytmu zostały sekwencje wideo przechwycone za pomocą kamer zintegrowanych z komputerami przenośnymi. Na podstawie zebranych danych zostało udowodnione, że dzięki zastosowaniu prostego algorytmu jest możliwa detekcja ruchu ust przy większości zarejestrowanych sekwencji.

2. Wstęp

2.1. Cele i założenia projektu

Celem realizowanego projektu było stworzenie systemu detekcji ruchu ust z wykorzystaniem prostego algorytmu do tego przeznaczonego oraz udowodnienie bądź obalenie tezy, iż zastosowanie podstawowych znanych metod wystarczy do dokładnego wychwytywania poruszających się ust na sekwencji wideo.

Głównym założeniem było dostarczenie odpowiednio przygotowanych plików z nagraniami twarzy, wykonanych w podobnym środowisku i układzie, w celu porównania efektywności stworzonego rozwiązania. W wyciągnięciu wniosków z otrzymanych wyników, posłużyło ich statystyczne opracowanie, które było również podstawą do potwierdzenia skuteczności detekcji ruchów ust większej jak 90%.

2.2. Ogólny zarys proponowanego rozwiązania

Proponowane rozwiązanie problemu wykrywania ruchu ust oparte zostało o wiedzę zdobytą na zajęciach z Interfejsów Multimodalnych oraz własne doświadczenia z przetwarzaniem obrazów. Implementowany algorytm składa się z kilku podstawowych kroków m.in. konwersji obrazu do innej przestrzeni barw, wyznaczania MHI, operacji morfologicznych czy etykietowania.

Dostarczona sekwencja wideo jest przepuszczana przez skonstruowany w tym celu system, który wyodrębnia najistotniejsze informacje oraz wizualizuje etapy pośrednie przetwarzania obrazu pod kątem poruszających się ust.

2.3. Dyskusja alternatywnych rozwiązań

Rozwiązaniem alternatywnym do prezentowanego, byłby wbudowany system detekcji ruchu obiektów coraz częściej implementowany we współczesnych kamerach. Alternatywę stanowiłoby również stworzenie bardzo rozbudowanego i elastycznego algorytmu działającego w sposób podobny do będącego tematem projektu. Jednakże dla celów i założeń przedstawionych w poprzednich punktach dostarczona implementacja jest w zupełności wystarczająca.

2.4. Struktura pracy

W rozdziale 3 założono prezentację koncepcji tworzonego rozwiązania wraz z wizualizacją pośrednich etapów działania algorytmu. Rozdział 4 zawiera przedstawienie wyników, ich omówienie oraz wnioski z nich wynikające. Podsumowanie niniejszej pracy znajduje się w rozdziale 5. Rozdział 7 poświęcony jest opisowi opracowanych rozwiązań oraz metod postępowania. Następny, rozdział 8 zawiera informacje techniczne oraz wykonawcze projektu. Rozdziały 9 i 10 zawierają kolejno opis procedur informatycznych oraz spis zawartości płyty CD.

3. Koncepcja proponowanego rozwiązania

Proponowane rozwiązanie zostało podzielone na dwie główne fazy, rozpoczynając na konwersji i prostych operacjach na wejściowej sekwencji wideo, kończąc na dużo bardziej skomplikowanych i czasochłonnych analizie obrazu pod względem jasności oraz algorytmie detekcji ruchu ust.

3.1. Wstępne przetwarzanie obrazu

W pierwszej fazie, przetwarzana sekwencja wideo jest poddawana konwersji do odcieni szarości, umożliwia to dalsze postępowanie z obrazem i poddanie go działaniu MHI (Motion History Image) co umożliwia śledzenie ruchu na obrazie poprzez budowanie obrazu historii ruchu. Poruszające się usta pozostawiają w każdej próbkowanej chwili czasowej swój odcisk rzutowany na płaszczyznę obrazu.

Po wykonaniu MHI sekwencja przechodzi przez ciąg operacji filtrowania medianowego oraz zamykania, co umożliwia wyczyszczenie oraz dokładniejsze zaznaczenie kształtów na obrazie.

Ta część systemu została zbudowana w taki sposób, aby możliwy był podgląd obrazu po konwersji, wykonaniu MHI oraz pierwszym przebiegu przez filtrowanie medianowe i zamknięcie.

3.2. Analiza obrazu i detekcja ruchu ust

Etap analizy rozpoczyna się w bloku, który wyznacza na obrazie obszary odróżniające się jasnością od tła. Pozwala to wydzielić sylwetkę nagrywanej osoby od pozostałych obiektów występujących na klatce nagrania. Z teoretycznego punktu widzenia system mógłby mieć swoje zakończenie po wyjściu z tego bloku, ponieważ umożliwia on wykrywanie większych ruchów analizowanego obszaru. Natomiast podchodząc do problemu bardziej praktycznie, taki wynik nie jest wystarczająco satysfakcjonujący, dlatego też otrzymane dane przekazywane są do bloku zawierającego algorytm umożliwiający detekcję nawet niewielkich ruchów warg.

3.3. Algorytm detekcji niewielkich ruchów ust

Jest to dość prosty, aczkolwiek bardzo skuteczny algorytm wykorzystujący współczynnik kształtu zwany współczynnikiem Fereta charakteryzujący wydłużenie cząstki. Formalna definicja takiego współczynnika jest następująca:

$$R_F = \frac{L_k}{L_V}$$

gdzie:

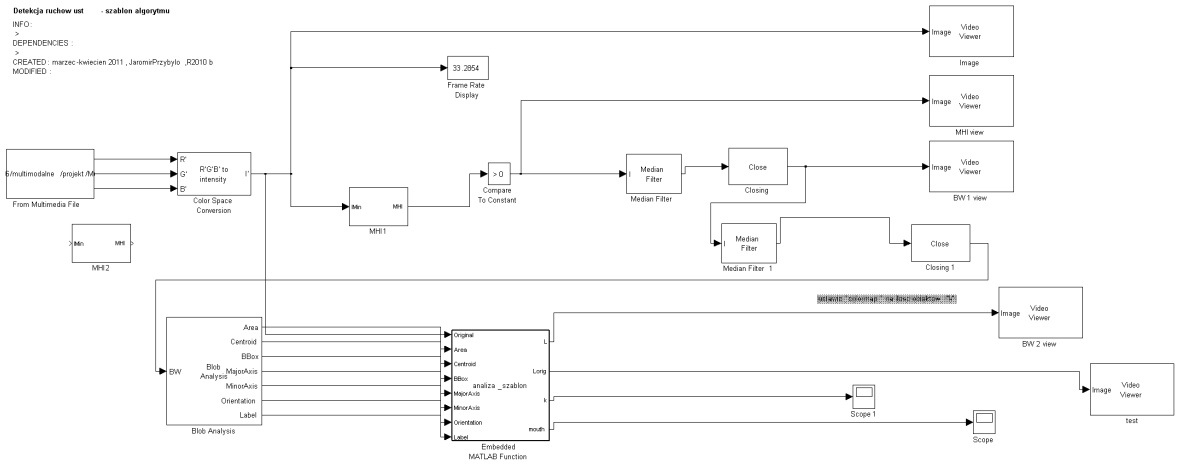
L_k - maksymalna średnica obiektu w poziomie

L_V - maksymalna średnica obiektu w pionie

Dodatkowo analizowany jest jeszcze współczynnik orientacji obiektu, co w połączeniu ze współczynnikiem Fereta umożliwia wykrycie tylko kształtów, które w rzeczywistości odpowiadają poruszającym się ustom, czyli zakresowi kształtu od elipsy do koła bez większych zmian orientacji. Takie podejście eliminuje zakłócenia wywoływane poprzez np. drobne ruchy koszulki osoby której zachowanie zostało zarejestrowane.

3.4. Schemat blokowy systemu

Budowa systemu została przedstawiona na schemacie blokowym pochodzącym z programy MatLab Simulink.



4. Rezultaty i wnioski

Przeprowadzając serię testów na czterech różnych sekwencjach wideo zarejestrowanych w odmiennych warunkach oświetleniowych, ale za pomocą tego samego sprzętu, którym była kamera wbudowana w komputer przenośny *Apple MacBook Pro* osiągnięte zostały satysfakcjonujące wyniki detekcji ruchu ust.

Nazwa pliku sekwencji wideo	Movie_0			**Komentarze**	**Uwagi**
				Podstawowy film z dobrym oświetleniem od frontu, model mało ruchliwy. Zdarza się ze znacznie porusza się ubranie, ale i tak ruch ten nie jest kwalifikowany jako ruch ust.	
Klatki w których wystąpił ruch	*rzeczywisty*	*wykryty*	*opóźnienie*		
				Ani razu nie zdarzyło się zgubienie ruchu ust. To znaczy, nawet podczas długiej wypowiedzi, system nie tracił informacji o tym że	
	9	9	0	ruch występuje.	
	23	23	0		
	29	29	0		
	49	49	0		
	59	63	4		
	89	90	1		
	104	104	0		
	116	116	0		
	127	130	3		
	173	174	1		
	190	195	5		
			Średnio	**Wykrywalność [%]**	
			1,2727272727	100,0%	

Nazwa pliku sekwencji wideo	Movie_1			**Komentarze**	**Uwagi**
				Testowy film ze słabym oświetleniem z góry. Ustawienia programu takie same jak w pierwszym przypadku. Model zmęczony ciągłym nagrywaniem czasem mruga oczami. Mrugnięcia nie są interpretowane jako ruch ust.	
				Poza jednym razem podczas długich sekwencji (takich w których nie było zatrzymania ruchu ust na więcej niż 8 klatek) nie był gu-	
Klatki w których wystąpił ruch	*rzeczywisty*	*wykryty*	*opóźnienie*	biony przez system	
	13	15	2	jeden raz ruch nie został wykryty	
	21	23	2		
	42	45	3		
	88	98	10		
	110				
	134	139	5		
	146				
	210	220	10		20 klatkowa strata ruchu
	287	289	2		
			Średnio	**Wykrywalność [%]**	
			4,8571428571	77,0%	

Nazwa pliku sekwencji wideo	Movie_2			**Komentarze**	**Uwagi**
				Testowy film ze złą pozycją przed kamerą i modelem bardzo zaspanym.	
Klatki w których wystąpił ruch	*rzeczywisty*	*wykryty*	*opóźnienie*		
	28	30	2		
	50	51	1		
					10 klatkowa strata ruchu przy robieniu
	72	73	1		dziubka z ust …
	122	122	0		
		139			mocne mruganie
	145	145	0		
	165	165	0		
	189	191	2		
		205			mocne mruganie
	215	215	0		
	230	230	0		
		240			mocne mruganie
	241	247	6		
	265	265	0		
	289	289	0		KONIEC PO 300 klatkach
			Średnio	**Wykrywalność [%]**	
			1	100,0%	

Nazwa pliku sekwencji wideo	Movie_3			Komentarze	Uwagi
				Średnie oświetlenie i modelowi wpadła rzęsa do oka	
Klatki w których wystąpił ruch	*rzeczywisty*	*wykryty*	*opóźnienie*		
	57	57	0		
	80	84	4		
	100	100	0		
	140	142	2		
	168	168	0		
	200	210	10		
		247			mocne mruganie (screenshot 31)
					KONIEC PO 300 klatkach
			Średnio	**Wykrywalność [%]**	
			2,6666666667	100,0%	

Podsumowanie

Średnie opóźnienie wykrycia 2,4491341991 klatki

Średnia wykrywalność 94,25%

Ze względu na chęć lepszego odzwierciedlenia oraz porównania przetwarzanych sekwencji wideo, dłuższe zostały ograniczone do przeanalizowania pierwszych 300 klatek zarejestrowanego obrazu.

Na podstawie otrzymanych wyników można stwierdzić, że dla zdecydowanej większości przypadków pojawiającego się ruchu ust algorytm zadziałał prawidłowo. Pewne etapy jego działania zostały udokumentowane w postaci zrzutów ekranów, które znajdują się na dołączonej do raportu płycie CD. Każde odstępstwo od ideału zostało zarejestrowane i opatrzone odpowiednim komentarzem.

Jak dowodzą przeprowadzone testy, za pomocą dość prostego algorytmu jest możliwe osiągnięcie wykrywalności ruchu ust na poziomie 90%, a nawet większym tak jak to wystąpiło w analizowanym przypadku. Poziom wykrywalności jest w znacznej mierze uzależniony od warunków w jakich została zarejestrowana sekwencja wideo, dla bardzo złych warunków oświetleniowych oraz pozycji modela wykrywanie jest ciężkie bądź nawet nie możliwe, natomiast przy średnio dobrym i dobrym oświetleniu oraz wykonującym tylko niewielkie ruchy modelu wykrywana jest zdecydowana większość ruchów.

5. Podsumowanie

Głównym celem i założeniem projektu było stworzenie prostego algorytmu umożliwiającego detekcję ruchu ust na poziomie co najmniej 90% przypadków. Wyniki oraz wnioski przedstawione w poprzednim rozdziale dają pewność, że realizacja tak sformułowanego celu jest możliwa możliwa. Realizowany problem został rozwiązany w pełnym założonym zakresie.

Oczywiście przygotowany algorytm nie jest idealny i działa tylko w specyficznych warunkach rejestracji wideo, gdzie model umieszczony jest w odpowiedniej odległości od urządzenia rejestrującego, odpowiednio doświetlony i nie wykonuje żadnych zbędnych ruchów. Kolejne modyfikacje rozwiązania mogłyby służyć zwiększeniu wykrywalności ruchu ust na wszystkich analizowanych sekwencjach wideo gdzie występuje złe doświetlenie, poruszający się model ustawiony pod kątem i wiele innych czynników, które obniżają efektywność prostego algorytmu który został stworzony w ramach projektu. Również zastosowanie algorytmu mogłoby być zdecydowanie szersze. Dla przykładu można byłoby określać nastrój i poziom napięcia osoby mówiącej, lecz tak postawiony cel byłby nie do zrealizowania za pomocą prostych metod i w tak krótkim czasie.

6. Literatura

[1] Tadeusiewicz R., Korohoda P., „Komputerowa analiza i przetwarzanie obrazów.” Wyd. Fundacji Postępu Telekomunikacji, Kraków, 1997.

7. DODATEK A: Opis opracowanych narzędzi i metody postępowania

Przygotowany system przeznaczony jest do użytkowania w środowisku oprogramowania MatLab co najmniej w wersji 2008b. Warunkiem koniecznym do uruchomienia jest posiadanie zainstalowanej wersji wcześniej wspomnianego oprogramowania pod systemem Windows 7. W celu przeprowadzenia detekcji ruchu ust za pomocą dostarczonego systemu jest posiadanie pliku .avi, na którego podstawie będzie przeprowadzana detekcję oraz wykonanie następujących kroków:

1. Uruchomienie oprogramowanie MatLab
2. Załadowanie pliku **mouthDet_template_R2010b.mdl**
3. Otworzy się okienko z blokowym schematem algorytmu, należy dwukrotnie kliknąć na pierwszy z bloków o nazwie „Video sequence”
4. W polu Input wybrać przygotowany wcześniej plik z sekwencją wideo
5. W celu uruchomienia systemu kliknąć klawisz F5 lub przycisk „Play”
6. Otworzą się okienka wizualizujące pośrednie etapy przebiegu algorytmu oraz okienko „Detection”, w którym będzie można zaobserwować poruszające się usta

Do testów systemu zostały wykorzystane nagrania zarejestrowane w specyficznych warunkach oświetleniowych. Na etapie wstępnych testów, możliwe było wyciągnięcie wniosków, że przygotowane algorytmy dobrze radzą sobie z wideo, na którym postać jest oświetlana w taki sposób, że usta chociaż delikatnie można odróżnić od reszty twarzy. Nie zaleca się rejestrowania nagrań kiedy postać jest zacieniona, bądź też nadmiernie doświetlona.

System przyjmuje pliki wideo w formacie .avi. Zalecane jest przygotowanie takich plików bez kompresji, nie zaburzy ona wtedy osiąganych wyników. Na wyjściu systemu otrzymywany jest obraz jedynie poruszających się ust oraz wykres pokazujący kiedy na obrazie występuje ruch.

8. DODATEK B: Realizacja proponowanego rozwiązania

Przygotowane rozwiązanie nie wymaga wysoce wyspecjalizowanego sprzętu. Zostało ono zoptymalizowane do pracy z kamerami internetowymi, które obecnie bardzo często wbudowane są w komputery przenośne. Jakość detekcji ruchu zależy w pewnym stopniu od rozdzielczości zarejestrowanego obrazu, im większa rozdzielczość tym lepsze wyniki zwraca aplikacja.

Korzystanie z dostarczonego systemu jest możliwe tylko i wyłącznie w ramach środowiska MatLab, zalecana jest jego najnowsza wersja pracująca na systemach z rodziny Linux bądź Windows.

Opis działania poszczególnych części algorytmu został już przedstawiony w rozdziale 3 pracy. W gwoli uzupełnienia przedstawione zostaną wykorzystywane przezeń współczynniki i wzory nie uwzględnione poprzednio.

MHI (Motion History Image):

- prób (treshhold): 9
- tau: 8

Filtr medianowy (I):

- sąsiedztwo: [7 7]

Zamknięcie (I):

- sąsiedztwo: strel('disk', 3)

Filtr medianowy (II):

- sąsiedztwo: [5 5]

Zamknięcie (II):

- sąsiedztwo: strel('line', 20,0)

9. DODATEK C. Opis informatyczny procedur

Implementację algorytmu detekcji ruchów ust przeprowadzono w środowisku MatLab 2010b. Kompilacja i uruchomienie systemu jest możliwe tylko za jego pośrednictwem.

W budowie aplikacji wykorzystane zostały standardowe bloki dostępne w MatLab Simulink oraz jeden blok zawierający algorytm umożliwiający wykrywanie drobnych zmian kształtu ust na analizowanej sekwencji wideo.

```
/*********************************************************************/
/*                                                                   */
/*  analiza_szablonu                                                 */
/*                                                                   */
/*  Przeznaczenie:                                                   */
/*      Analiza zmian wspó czynnika kszta tu ust                      */
/*                                                                   */
/*  Argumenty funkcji:                                               */
/*                                                                   */
/*      (I) Area – obszar                                            */
/*      (I) Centroid                                                 */
/*      (I) Bbox                                                     */
/*      (I) MajorAxis                                                */
/*      (I) MinorAxis                                                */
/*      (I) Orientation – orientacja kszta tu                        */
/*      (I) Label – etykieta                                         */
/*      (O) L - obraz indeksowany po analizie                        */
/*      (O) k - ilo     obiektów po analizie                         */
/*      (O) mouth - sygna   [0,1] informuj cy o fakcie               */
/*                  detekcji ruchu ust                               */
/*                                                                   */
/*  Funkcja zwraca:                                                  */
/*      Tak jak opisano powy ej                                      */
/*                                                                   */
/*  U ywane funkcje:                                                 */
/*      abs, min, size, length, zeros, uint8                         */
/*                                                                   */
/*  U ywane zmienne:                                                 */
/*      brak zmiennych globalnych                                    */
/*                                                                   */
/*  Uwagi:                                                           */
/*      Algorytm wykrywania ruchu ust dzia a przy spe nionych        */
/*      poni szych za o eniach:                                      */
/*       - u ytkownik nie powinien wykonywa   znacz cych ruchów      */
/*         przed kamer   i powinien porusza   tylko ustami, oczami   */
/*         i mruga   powiekami                                       */
/*       - lekkie poruszanie cz   ci ubra , np. r kawów nie powinno  */
/*         wp yn     negatywnie na wynik dzia ania algorytmu         */
/*       - bardzo istotn   spraw   jest odpowiednie o wietlenie      */
/*         osoby                                                     */
/*                                                                   */
/*  Autor:                                                           */
/*      Wojciech Dani o,  Mateusz Ziarko, 2011                       */
/*                                                                   */
/*  Ostatnia modyfikacja:                                            */
/*      16.05.2011                                                   */
/*                                                                   */
/*********************************************************************/

function   [L,Lorig,k,mouth]     =   analiza_szablon(Original,   Area,
Centroid, BBox, MajorAxis, MinorAxis, Orientation, Label)

[m,n]=size(Label);
ilobj=length(Area);
```

```
rotFeret = zeros(1,ilobj);
Lorig = Original
L=zeros(m,n,'uint8');

mouth=0;

% Analiza
i=1;k=1;
for i=1:ilobj
    % Sprawdzenie wielkosci obiektow
    if Area(i) > 100 && Area(i) < (m*n*0.5)
        % sprawdzanie odpowiedniego kata wykrytego obiektu
        % jezeli wspolczynnik Orientation jest odchylony od 0 o wi cej
          niz 0.5
        % istnieje b. duze prawdopodobienstwo, ze nie wykryto ust
        %(lecz np. powiewajacy rekaw od bluzki - klatka ~ 100)
        if abs(Orientation(i)) < 0.5
            % rotFeret jest specyficznym wspolczynnikiem Fereta.
            % Jest to wspolczynnik Fereta ale analizowany w obr#conej
              przestrzeni
            % i jest on rowny stosunkowi promieni elipsy opisanej na
              znalezionym ksztalcie
            % im blizszy 1 tym ksztalt jest blizszy kole.
            % Podczas otwierania ust, ksztalt zmiany otworu
              gebowego jest blizszy kolu
            % niz ksztalt zakreslony przez zmine polozenia brody.

            rotFeret(i) = MajorAxis/MinorAxis;
%             L=L + uint8(Label==i)*k;
%             k=k+1;
        end
    end
end

% sprawdzamy ktory z ksztaltow ma wspolczynnik rotFeret najblizszy 1
rotFeret = abs(rotFeret-1);
[min mini] = min(rotFeret)

% sprawdzamy czy jakikolwiek ksztalt zostal wykryty i jezeli tak to
dajemy output
if min ~= 1
    mouth = 1
    L=L + uint8(Label==mini)*k;
    Lorig = Lorig + uint8(Label==mini)*255;
end
```

10. DODATEK D. Spis zawartości dołączonej płyty CD

- doc/ - tekst raportu w postaci elektronicznej oraz
- src/ - pliki źródłowe projektu
- media/ - sekwencje wideo wykorzystane podczas testów oraz zrzuty ekranów prezentujące działanie algorytmu