STUDIA INFORMATICA 2013
Volume 34 Number 2A (111)

Grzegorz KOSZOWSKI
Politechnika Śląska, Instytut Informatyki

ROZPOZNAWANIE GESTÓW WYKONYWANYCH DŁOŃMI NA PODSTAWIE POJEDYNCZYCH OBRAZÓW[1]

Streszczenie. Artykuł zawiera wyniki 3 grup algorytmów stosowanych do rozpoznawania gestów wykonywanych dłońmi na podstawie pojedynczych obrazów. Algorytmy te opierają się na: dopasowaniu szablonów masek binarnych obrazów, metodach opartych na konturach dłoni w obrazie oraz detekcji reprezentantów punktów reprezentujących krzywizny konturów dłoni. Wyniki uzyskano na podstawie 3 baz danych, w których obrazy zostały podzielone na bazę testową i referencyjną. Jedna baza referencyjna została wygenerowana na podstawie modelu 3D.

Słowa kluczowe: rozpoznawanie gestów, dopasowywanie szablonów, metryka Hausdorffa, detekcja punktów krzywizn

HUMAN HAND GESTURE RECOGNITION BASED UPON SINGLE IMAGES

Summary. This Paper include results of 3 groups of algorithms used to hand pose gesture recognition based on single images. Those algorithms base on: Template Matching of binary masks from images, methods based on hand contours in image and the Points representing points from curvatures detection. Results were obtained with 3 databases, which images were split for test and reference database. One of reference databases was generated from the 3D model.

Keywords: gesture recognition, Template Matching, Hausdorff metric, curvature points detection

[1] This work was supported by the European Union from the European Social Fund (grant agreement number: UDA-POKL.04.01.01-00-106/09)

1. Wprowadzenie

Często spotykamy się z zastosowaniem ludzkiego głosu w komunikacji między człowiekiem a komputerem. Odpowiednie aplikacje umożliwiające taką komunikację istnieją od ponad 6 lat nawet w telefonach komórkowych, których aktywna liczba już w 2012 roku przekroczyła 6 mld. Innym zastosowaniem pospolitego telefonu jest połączenie modułu GPS z wirtualną rzeczywistością (z ang. Virtual Reality), bazującą na wizji komputerowej. W 2010 roku w sprzedaży pojawiły się 2 kontrolery gier: The Movie i Kinect, pierwszy bazujący na ruchu dłoni trzymającej kontroler, drugi – na ruchu szkieletu ciała. W 2011 roku Grupa Allegro udostępniła darmową aplikację dla smartfonów, umożliwiającą rozpoznawanie produktu na podstawie jego zdjęcia, a następnie wyszukanie rozpoznanego produktu w serwisie aukcyjnym.

Jak można zauważyć, coraz częściej do komunikacji pomiędzy urządzeniami a użytkownikami używa się naturalnych metod komunikacji, jak głos bądź też ruch ciała. Również przekazywanie obrazu jako metody komunikacji staje się coraz powszechniejszą metodą wraz z rosnącą liczbą aparatów i kamer montowanych w wielu urządzeniach elektronicznych. Coraz częściej spotykamy się też z przypadkiem montowania podwójnego aparatu lub kamery, z przodu i z tyłu urządzenia.

W wyżej wymienionych urządzeniach celowo pominęliśmy rozwiązania korzystające z gestów ludzkich dłoni. Pomimo zwiększających się możliwości urządzeń rzadko jesteśmy świadkami takich rozwiązań, które powstają głównie na potrzeby osób niepełnosprawnych, a mogą okazać się równie atrakcyjne dla współczesnego konsumenta.

Samo rozpoznawanie gestów jest procesem skomplikowanym, szczególnie ze względu na ilość stopni swobody dłoni [1]. Problem może być podzielony na: wyekstrahowanie wektora cech z obrazu oraz klasyfikację gestu. My skupimy się na ekstrakcji różnych cech w celu porównania skuteczności różnych metod na różnych bazach danych.

Nasze podejście polega na stworzeniu dwóch baz danych, testowej i referencyjnej, przy użyciu obrazów reprezentujących gesty, przy czym obrazy dłoni poszczególnych osób nie mogą znaleźć się w dwóch bazach równocześnie.

W części 2 zostaną omówione niektóre prace dotyczące rozpoznawania gestów, w części 3 – stworzone bazy danych, w części 4 – algorytmy oraz ich modyfikacje, w części 5 zaś zaprezentujemy otrzymane wyniki klasyfikacji. Część 6 będzie zawierać konkluzję.

2. Zagadnienie w literaturze

Do tej pory zetknęliśmy się z wieloma podejściami do tworzenia wektora cech oraz klasyfikacji gestów. Rosales zastosował 2 referencyjne bazy danych zawierające 9 000 i 750 000 obrazów referencyjnych, z czego pierwsza zawierała obrazy gestów wykonywanych dokładnie w stronę kamery. Punkty charakterystyczne: nadgarstek, opuszki palców, środek ciężkości dłoni, zostały zebrane przy użyciu specjalnej rękawicy nakładanej podczas wykonywania gestów. Rosales zastosował metodę dopasowywania szablonu. Osiągnął 70% skuteczności [2]. Stenger dla każdego gestu zastosował 40 obrazów wzorcowych. Następnie każdy z nich został obrócony 10 razy, aby uzyskać obrazy od -10 do 10 stopni odchylone od obrazu wzorcowego, i otrzymano dla każdego 400 obrazów wzorcowych. Przy tworzeniu bazy danych również zastosował wirtualną rękawicę [3].

W [4] zastosowano 2 metody oraz ich połączenie. Pierwsza polegała na porównywaniu wektorów odległości kolejnych punktów konturu od środka ciężkości maski binarnej obrazów dłoni. Druga polegała na obliczeniu szybkiej transformaty Fouriera uzyskanego wektora. Pracę przeprowadzono na 7 gestach uzyskanych przy pomocy 7 osób. Otrzymane wyniki to 77% i 64%, połączenie zaś algorytmów – 87%.

W [5] zastosowano inny wektor cech. Proces składa się z wykrycia punktów krzywizn przy użyciu algorytmu Chetrikova i Szabo na podstawie konturu, a następnie znalezieniu jednego reprezentanta każdego ze zbioru punktów krzywizn. Do procesu kwalifikacji użyto par: odległość pomiędzy reprezentantem danej krzywizny a punktem środka ciężkości masy maski binarnej dłoni oraz kąt pomiędzy ustaloną prostą w obrazie a prostą stworzoną z wcześniej wymienionych punktów. Do klasyfikacji użyto metody najbliższego sąsiada oraz sztucznej sieci neuronowej. Eksperymenty przeprowadzono na 8 gestach i 14 360 obrazach. Uzyskano wyniki od 96,4% do 98,1%.

W [6] wektor cech oparto na wykrytych opuszkach palców dłoni. W przypadku gestów z niewidocznymi bądź też częściowo ukrytymi opuszkami palców skuteczność wyniosła od 10% do 20% i metody te, jak podają sami autorzy, służą raczej jako metody pomocnicze w przypadku rozpoznawania gestów. Warto nadmienić, że stosowane obrazy samej dłoni miały wymiary 800 na 600 pikseli.

Oprócz wyżej wymienionych stosuje się również samoorganizujące i samorosnące sieci, klasyfikowane następnie przy użyciu sieci neuronowych [7], oraz operację na obrazie: zliczanie pikseli w określonych kierunkach od puntu ciężkości masy dłoni czy też histogramów orientacji gradientów z mniejszych obrazów wydzielanych z obrazu dłoni [5].

3. Bazy danych

3.1. Detekcja nadgarstka i wstępna obróbka obrazu

W tworzeniu baz danych wykorzystano zrobione przez autora zdjęcia oraz autorską aplikację wykorzystującą model 3D do generowania obrazów z dłonią wykonującą dany gest. Każde zdjęcie poddano procesowi wstępnego przetworzenia w celu uzyskania binarnej maski obrazu. Użyty algorytm to połączenie metody detekcji skóry na podstawie barwy skóry w przestrzeni HSV, progowania, detekcji największych spójnych obszarów i następnie algorytmu pożaru prerii aż do najbliższej krawędzi. Do wykrycia krawędzi użyto filtru Canniego udostępnionego w bibliotece OpenCV [8] i autorskiego algorytmu do łączenia krawędzi. W niektórych przypadkach obrazy zostały ręcznie opracowane. Biblioteka OpenCV [8] została również użyta do wczytywania, zapisywania, pakowania, rozpakowywania oraz przechowywania struktur obrazów.

Do detekcji nadgarstka wykorzystano wstępnie obrócone wraz z kierunkiem obrazy dłoni. Obrazy to maski binarne dłoni już po detekcji skóry w wymiarach 800 na 600 pikseli, gdzie 0 oznacza piksel tła, a 1 – piksel dłoni. W celu opisu przyjmuje się, że przekrój p oznacza długość przekroju prostopadłego do kierunku dłoni. Długość wyraża się w pikselach. Zakładamy również, że dłoń obrócona jest w prawo na obrazie, przedramię zaś znajduje się po lewej stronie. Algorytm detekcji składa się z 2 kroków. Pierwszy polega na oszacowaniu długości dłoni i pierwszego punktu podejrzewanego o bycie punktem reprezentującym nadgarstek. Od lewej strony obliczamy przekrój p dla kolejnych rosnących współrzędnych x, szukając takiego x, dla którego przekrój rośnie bądź jest równy dla kolejnych n wartości x. W pierwszej fazie przebiegu n jest równe 0,8% szerokości obrazu (minimum 5 pikseli). Szacowana długość dłoni d to odległość od wykrytego x do ostatniego x, dla którego przekrój jest większy od 0. W drugim przebiegu szukamy najwęższego przekroju dla x od wartości wykrytej w pierwszym przebiegu do $x_2 = x + 30\% * d$. Każda znaleziona wartość x musi spełniać również warunek z przebiegu pierwszego, z tą różnicą, iż n jest równe 2% szacowanej szerokości dłoni.

Ten prosty algorytm pozwala wykryć nadgarstek w obrazach wykonywanych przez większość osób w naszej bazie danych. W przypadku tylko jednej osoby algorytm ten wykrywa nadgarstek w innych miejscach, co pokazano na rys. 1. W przypadku uproszczonej wersji algorytmu, zawierającej tylko pierwszy przebieg algorytmu, błąd występował w 8 obrazach, w przypadku drugiej – w 6. Przy próbie manipulacji n bądź też szukania ostatniego przewężenia przy dłoniach w gestach ze złożonymi palcami i schowanym kciukiem wykrywany nadgarstek był przesunięty w kierunku palców lub znajdował się przed kostką nadgarstka.

W przedstawionych w tym artykule bazach danych pozycja nadgarstka w przypadku wizualnego stwierdzenia złego wykrycia została ręcznie skorygowana.

Wszystkie obrazy zostały poddane procesowi normalizacji. Normalizacja oraz tworzenie bazy danych na podstawie modelu zostały dokładnie opisane w [9].

a) b)

Rys. 1. Przykłady niepoprawnego wykrycia nadgarstka. Przekroje reprezentowane są przy użyciu linii wertykalnych. Przekrój pierwszy od lewej reprezentuje wykryty nadgarstek, drugi od lewej – poprawny. Przykładowa długość przekroju w oryginalnym obrazie to odpowiednio 124 piksele i 126 pikseli dla obrazu a)

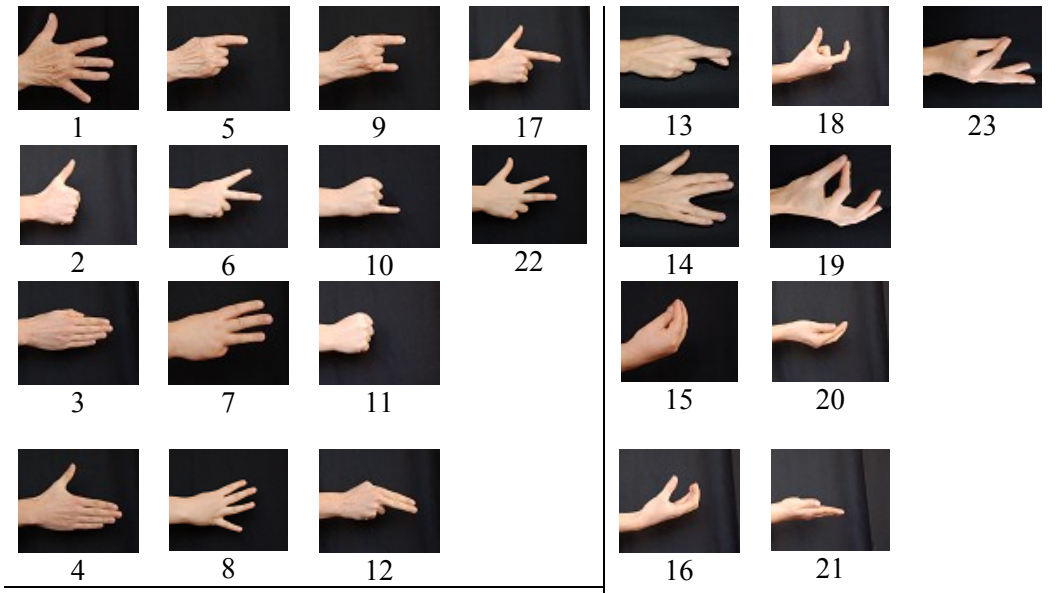
Fig. 1. Examples of not properly detected wrist. Sections are presented with an usage of vertical lines. First section from left represent detected wrist and second proper wrist. Example length of section in original image is adequately 124 and 126 pixels for image a)

3.2. Bazy danych

Do celów badawczych utworzono 3 bazy danych. Bazy nr 1 i 3 zawierają obrazy dłoni rzeczywistych osób, natomiast baza nr 2 zawiera obrazy dłoni rzeczywistych osób w części testowej oraz obrazy wygenerowane przy użyciu aplikacji na podstawie modelu 3D w części referencyjnej. Wszystkie bazy danych powstały na podstawie 180 zdjęć i 129 obrazów wygenerowanych przy użyciu modelu 3D. Bazy nr 1 i 2 zawierają obrazy 14 gestów, a baza 3 – obrazy 23 gestów. Warto nadmienić, że bazy nr 1 i 2 zawierają gesty bardziej zróżnicowane pod względem wizualnym. Zdjęcia dłoni zrobiono osobom w wieku od 15 do 79 lat. Zarówno podział, jak i obrazy przykładowych gestów zamieszono na rys. 2. Liczba obrazów referencyjnych w bazie nr 1 to 32, a testowych to 80, w bazie nr 2 odpowiednio 129 i 122, w bazie nr 3 zaś – 63 i 117.

W bazie testowej znajduje się od 52% do 72% obrazów, natomiast w bazie referencyjnej od 28% do 48% reprezentacji danego gestu. W przypadku bazy referencyjnej generowanej z modelu 3D procentowo zawiera ona więcej obrazów niż w przypadku baz nr 1 lub 3. Liczba obrazów reprezentujących dany gest nie jest jednakowa, ponieważ nie wszystkie osoby były w stanie wykonać prezentowane im gesty. Szczególnie dotyczyło to osób starszych, które skarżyły się na ból w stawach i nie potrafiły np. skrzyżować palców.

W bazach danych zostały zapisane zarówno znormalizowane binarne maski obrazów dłoni, jak i wektory cech.



Rys. 2. Reprezentacja graficzna gestów w bazach danych
Fig. 2. Graphic representation of gestures in databases

4. Algorytmy

4.1. Dopasowywanie szablonu

W tym algorytmie porównujemy binarne maski z bazy testowej i referencyjnej. Każdy obraz z testowej bazy danych porównywany jest z wszystkimi obrazami w bazie referencyjnej z zastosowaniem metody dopasowywania szablonu (z ang. *Template Matching*). W tym celu użyto metody dostępnej w bibliotece OpenCV [8], której wynikiem jest skalar:

$$R=\sum_{x,y}(T(x\ ,y\ )-I(x\ ,y\ ))^2, \tag{1}$$

gdzie: R to wynik, T i I to para porównywanych obrazów, a $x\,,y$ to współrzędne obrazu. Miara podobieństwa danych dwóch obrazów wynosi 0 przy obrazach identycznych.

4.2. Odległość pomiędzy punktami konturów

Kontur dłoni jest reprezentowany w postaci wektora punktów. Początek wektora zawsze znajduje się w lewej górnej połowie obrazu, na którym sama dłoń skierowana jest w prawo. W ten sposób uzyskujemy uporządkowany wektor punktów.

W tych algorytmach celem jest uzyskanie jednej liczby reprezentującej odległość pomiędzy konturami. Zaproponowano tutaj dwa podejścia: odległość Hausdorffa oraz maksimum

z pary liczb reprezentujących sumę kwadratów najmniejszych odległości pomiędzy punktami konturu obrazu z bazy referencyjnej oraz bazy testowej, a także odwrotnie, obrazu bazy testowej i referencyjnej. Druga metoda będzie przez nas nazywana Sumą Kwadratów Odstępów (SKO). Operację tę wykonuje się pomiędzy danym obrazem z bazy testowej a kolejnymi obrazami z bazy referencyjnej. Sama klasyfikacja polega na metodzie najbliższego sąsiada. Odległość Hausdorffa opisuje wzór (2), SKO zaś – wzór (3):

$$H(X,Y)=\max\{\sup_{x\in X}\inf_{y\in Y} d(x,y),\sup_{y\in Y}\inf_{x\in X} d(x,y)\}, \tag{2}$$

$$SKO(X,Y)=\max\{\sum_{x\in X}\left(\inf_{y\in Y} d(x,y)^2\right),\sum_{y\in Y}\left(\inf_{x\in X} d(x,y)^2\right)\}, \tag{3}$$

gdzie H oznacza odległość Hausdorffa, SKO to wynik drugiej metody, nazywanej przez nas Sumą Kwadratów Odstępów, X i Y oznaczają wektory reprezentujące kontury poszczególnych obrazów, a x i y to punkty zapisane w wektorach. Obie te metody jako wynik dają skalarną wartość liczbową, która staje się miarą podobieństwa pomiędzy parą obrazów. W przypadku identycznych obrazów wynikiem powyższych operacji jest 0.

4.3. Punkty reprezentujące odcinki największych krzywizn konturu

W tych algorytmach wykorzystuje się w praktyce od kilku do kilkunastu wykrytych punktów, będących reprezentantami poszczególnych krzywych o określonej minimalnej krzywiźnie, uzyskanych z konturu dłoni. Algorytm wykrycia tych punktów składa się z 3 faz. Na początku musimy uzyskać kontur opisany w podpunkcie 4.2. W fazie 2 poruszamy się punkt po punkcie wektora reprezentującego ciągły kontur dłoni. Dla każdego punktu obliczamy kąt trójkąta pomiędzy tym punktem a punktami w przyjętym zakresie odległości. W naszym wypadku odległość ta to zakres od 4 do 8 punktów. Jeżeli kąt wcześniej wspomnianego trójkąta znajduje się w założonym zakresie, punkt ten należy do krzywych o określonej krzywiźnie, w innym wypadku – nie należy. W rezultacie uzyskujemy zbiory punktów reprezentujących odcinki krzywizn konturu dłoni. Jest to algorytm Chetrikova i Szabo [10]. W celu uzyskania lokalnych reprezentantów poszczególnych krzywizn wybieramy punkt znajdujący się w środku wektora reprezentującego dany odcinek krzywej. Proces przedstawiono na rys. 3a. Zależnie od metody będą brane pod uwagę: położenie punktów w 2-wymiarowym obrazie, odległość pomiędzy punktami a punktem będącym środkiem ciężkości dłoni, kąt pomiędzy założona prostą a prostą przechodzącą przez dany punkt reprezentujący krzywiznę oraz punkt będący środkiem ciężkości dłoni. Poszczególne cechy pokazano na rys. 3b. Według autorów [5] wykryte punkty powinny reprezentować czubki palców oraz zagłębienia pomiędzy palcami.

a) b) c)

Rys. 3. Od lewej: kontur (a), wyniki operacji wykrywania punktów krzywizn (b), punkty reprezentujące krzywizny i uzyskane cechy dla jednego z punktów: d – odległość od środka ciężkości masy dłoni, α – kąt trójkąta opisanego w podpunkcie 4.3 (c)

Fig. 3. From left: contour (a), Curvature detection operation results (b), points representing curvatures and obtained features for one of points: d – distance from the hand mass center, α – angle described in 4.3 triangle (c)

W tym artykule przedstawiono wyniki następujących metod opierających się na lokalnych punktach krzywizn:

- Odległość Hausdorffa pomiędzy lokalnymi reprezentantami punktów krzywizn pary obrazów.
- SKO (podpunkt 4.2) pomiędzy lokalnymi reprezentantami punktów krzywizn pary obrazów.
- Różnica pomiędzy odległościami w dwóch obrazach pomiędzy poszczególnymi lokalnymi reprezentantami punktów krzywizn a środkiem ciężkości dłoni.
- Wyżej wymieniona różnica i różnica kątów pokazanych na rys. 3b przy przypisanej do tych dwóch pozycji wadze.

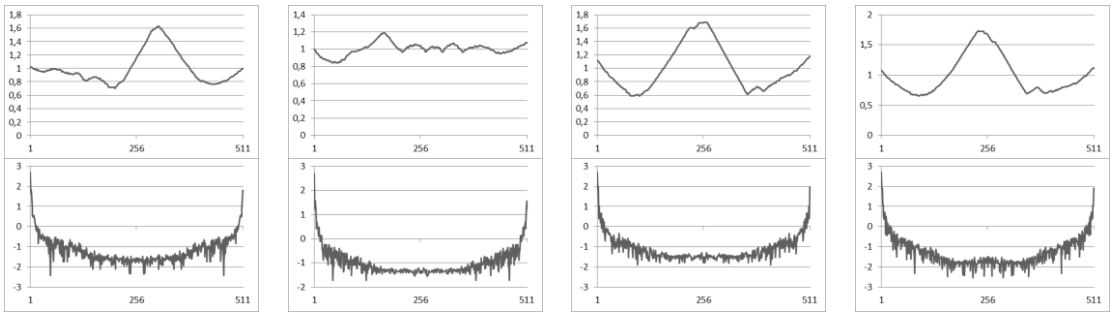
We wszystkich tych metodach porównanie dwóch obrazów następuje wówczas, gdy liczba wykrytych punktów się zgadza, w przeciwnym wypadku dana para obrazów nie jest brana pod uwagę w momencie klasyfikacji.

4.4. Wektor odległości punktów konturu od centrum masy dłoni

Innym podejściem, również opartym na konturze, jest analiza wektora odległości euklidesowej punktów konturu od punktu będącego środkiem ciężkości dłoni. W naszym przypadku długość początkowego wektora konturu wynosi od 211 do 389 punktów. Po utworzeniu wektor odległości jest rozciągnięty do 512 punktów z zastosowaniem aproksymacji liniowej. Następnie wartości wektora odległości zostają znormalizowane do maksymalnej wartości ze wszystkich elementów wektora lub też podzielone przez średnią ze wszystkich wartości elementów wektora odległości. Ostatnim etapem przygotowania wektora cech jest wyliczenie szybkiej dyskretnej transformaty Fouriera (SDTF) na podstawie wcześniej wyliczonego wektora odległości. W naszym przypadku po zastosowaniu SDTF skupimy się na częstotliwości i uzyskamy wektor liczb rzeczywistych:

$$STF(x) = \sqrt{I(x).re^2 + I(x).im^2} \tag{4}$$

Zarówno przykładowe wektory odległości, jak i SDTF zostały zamieszczone na rys. 4.



10 11 12 13

Rys. 4. Przykładowe wektory odległości (pierwszy wiersz) i SDTF w skali logarytmicznej dla gestów (drugi wiersz): 10,11,12,13

Fig. 4. Samples of distances vectors (first row) and SDTF in logarithm scale (second row) for gestures: 10,11,12,13

Wynikiem porównania dwóch obrazów będzie w przypadku porównywania wektorów odległości suma kwadratów odległości:

$$R_{odl} = \sum_{n} (x_n - y_n)^2, \tag{5}$$

gdzie x_n i y_n to wartości wektorów odległości pary obrazów, w przypadku SDTF zaś:

$$R_{SDTF} = \sum_{n} |x_n - y_n|, \tag{6}$$

gdzie x_n i y_n to wartości wektorów wynikowych SDTF.

W jednej z metod przy klasyfikacji zostanie wykorzystana również kompozycja uzyskanych wartości ze wzorów (6) i (5) z wagą w:

$$R_{kompozycji} = R_{odl} * w + R_{SDTF} * (1 - w), \tag{7}$$

gdzie: R_{odl} to wynik porównania znormalizowanych wektorów odległości, a R_{SDTF} to wynik z porównania SDTF.

5. Wyniki

5.1. Klasyfikacja

We wszystkich zastosowanych algorytmach klasyfikacja przebiega tak samo. Dla każdego z obrazów w bazie testowej obliczamy współczynnik podobieństwa pomiędzy obrazem z bazy testowej a każdym obrazem z bazy referencyjnej. Następnie wyszukujemy parę, dla której współczynnik był najmniejszy:

$$e_p = \min_{n}(R_n), \tag{8}$$

gdzie: n to liczba obrazów bazy referencyjnej, a R_n to wynik porównania obrazu z danym obrazem z bazy referencyjnej. Poprawna klasyfikacja zachodzi wówczas, kiedy dla znalezionej pary nazwy gestów się zgadzają. Oznacza to, że na obu obrazach pokazywany był ten sam gest.

5.2. Wyniki eksperymentów

Wszystkie metody zostały nazwane numerami:

1. Dopasowywanie Szablonów.
2. Metryka Hausdorffa pomiędzy konturami.
3. SKO (podpunkt 4.2) pomiędzy konturami.
4. Metryka Hausdorffa pomiędzy reprezentantami krzywizn.
5. SKO pomiędzy reprezentantami krzywizn.
6. Różnica pomiędzy odległościami między reprezentantami krzywizn a punktami ciężkości masy dłoni.
7. Różnica pomiędzy odległościami między reprezentantami krzywizn a punktami ciężkości masy dłoni i różnica pomiędzy kątami z podpunktu 4.3 a wagą w.
8. Różnica pomiędzy znormalizowanymi do średniej wartości wektorami odległości punktów konturu od punktu ciężkości masy dłoni.
9. Różnica pomiędzy znormalizowanymi do maksymalnej wartości wektorami odległości punktów konturu od punktu ciężkości masy dłoni.
10. Różnica pomiędzy SDTF (szybka dyskretna transformata Fouriera), pominięcie fazy.
11. Kompozycja metody 10 i 8.

Eksperymenty zostały przeprowadzone na komputerze przenośnym z procesorem Intel Core 2 Duo P 8700 2,53GHz. Wyniki eksperymentów na bazach nr 1, 2 i 3 zostały przedstawione w tab. 1.

Tabela 1

Wyniki klasyfikacji dla 3 baz danych, wyrażone w procentach

Baza danych	Metoda										
	1	2	3	4	5	6	7	8	9	10	11
1	82,5	77,5	86,25	73,75	75,0	47,5	43,75	82,5	83,75	75,0	83,75
2	77,0	60,6	68,8	43,4	45,0	31,1	25,4	72,1	69,6	58,1	72,1
3	70,9	71,7	74,3	59,8	68,3	41,8	32,4	68,3	68,3	61,5	68,3

Jak widać, najskuteczniejszą metodą w przypadku naszych baz danych okazała się metoda SKO, jednakże średni czas potrzebny na dokonanie obliczeń przekroczył 2 s (tab. 2) dla jednego obrazu z bazy testowej nr 2. W przytoczonych wcześniej pracach [1, 2, 5] często operowano na bazach zawierających 10 000 i więcej obrazów, co dałoby przy tej metodzie szacunkowo 200 s potrzebnych na analizę jednego zdjęcia, dlatego wydaje się, że nawet po

możliwych optymalizacjach metoda ta nie będzie przydatna w codziennych aplikacjach czasu rzeczywistego.

Tabela 2

Uśredniony czas klasyfikacji na jeden obraz [ms]

Baza danych	metoda										
	1	2	3	4	5	6	7	8	9	10	11
1	307	657	651	11	3	2	2	16	16	19	25
2	803	2312	2291	25	6	3	3	55	45	58	73
3	423	934	927	8	3	2	2	24	24	28	36

Interesujące okazały się metody operujące na wektorach odległości punktów konturu od centrum masy dłoni. W przypadku testowanych baz danych czas potrzebny na klasyfikację nie przekroczył 0,1 s, same zaś metody można poddać dalszej optymalizacji. W przyszłości to właśnie na tych metodach skupią się badania.

Zdecydowaną różnicę widać również pomiędzy bazami nr 1 i 2. Baza nr 2 zawiera w bazie referencyjnej 129 wygenerowanych obrazów gestów wykonywanych w różny sposób, lecz na podstawie jednego modelu 3D. Baza nr 1 zawiera tylko 42 obrazy gestów, jednakże pokazywane przez różne osoby, a więc są to różne modele dłoni. Według wstępnych badań posiadanie w bazie referencyjnej większej liczby modeli bardziej wpływa na skuteczność klasyfikacji niż liczba reprezentacji danego gestu jednego modelu. Hipoteza ta zostanie w przyszłości przebadana na większej bazie danych.

Rozpoznawalność poszczególnych gestów przedstawiono w tab. 3. Jak widać, bardzo dużo zależy od konkretnego gestu. Można wydzielić kilka gestów rozpoznawanych przez większość metod; są to: gesty 1, 9, 10 oraz te, które poprawnie rozpoznano tylko przy użyciu niektórych metod, jak gesty 19, 18.

Tabela 3

Tabela wystąpień poprawnej klasyfikacji dla wybranych gestów

Gest	Max.	Metoda										
		1	2	3	4	5	6	7	8	9	10	11
1	5	5	2	4	4	4	4	4	4	4	4	4
2	6	4	6	6	6	5	4	2	6	6	6	6
7	6	5	4	3	2	3	2	3	2	2	2	2
8	5	3	5	5	4	5	1	5	1	1	3	1
9	7	4	6	6	6	6	7	5	6	6	4	6
10	6	5	6	6	6	5	3	2	5	6	5	5
11	6	5	4	4	3	4	2	0	5	5	6	5
17	5	1	3	3	1	2	2	2	4	4	2	4
18	3	2	1	0	0	0	0	0	0	0	0	0
19	4	1	0	0	1	2	0	0	0	0	1	0

6. Podsumowanie

W niniejszym artykule zaprezentowano reprezentantów z 3 grup metod rozpoznawania gestów. Każdą metodę przetestowano na 3 bazach danych zawierających odpowiednio 14 gestów lub 23 gesty. Poprawność klasyfikacji gestów, zależnie od bazy danych oraz algorytmu, wahała się w granicach od 30% do 86%. Równie ważnym elementem było porównanie czasu detekcji w różnych metodach, szczególnie że celem jest utworzenie aplikacji działającej w czasie rzeczywistym.

W przyszłych badaniach planuje się porównanie baz referencyjnych tworzonych przy użyciu jednego modelu 3D z bazami tworzonymi przez jedną osobę, a także powiększenie istniejącej bazy danych o kolejne gesty i osoby. Algorytm rozpoznawania gestów zostanie oparty na wektorach odległości konturu od punktu ciężkości masy dłoni, które to podejście w przedstawionych badaniach okazało się kompromisem pomiędzy poprawnością rozpoznania a szybkością działania.

BIBLIOGRAFIA

1. Pickering C. A.: The search for a safer driver interface: a review of gesture recognition Human Machine Interface. IEE Computing and Control Engineering, 2005, s. 34÷40.
2. Rosales R., Sclaroff S.: Combining Generative and Discriminative Models in a Framework for Articulated Pose Estimation. International Journal of Computer Vision, Vol. 67, 2006.
3. Stenger B.: Template-Based Hand Pose Recognition Using Multiple Cues. Narayanan P. J. et al. (eds.): ACCV 2006, LNCS 3852, Springer-Verlag, Berlin/Heidelberg 2006, s. 551÷560.
4. Chen J.-Y., Chang Y.-H.: A hand-pose recognition system using a combined classifier of shift distances and Fourier features. The 20th IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP), 2007.
5. Wysocki M., Kapuściński T., Marnik J., Oszust M.: Rozpoznawanie gestów wykonywanych rękami w systemie wizyjnym. Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 2011.
6. Chaudhary A., Raheja J. L., Das K., Raheja S.: Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way: A Survey. International Journal of Computer Science & Engineering Survey (IJCSES), Vol. 2, No. 1, 2011.

7. Stergiopoulou E., Papamarkos N.: Hand gesture recognition using a neural network shape fitting technique. Engineering Applications of Artificial Intelligence, Vol. 22, Issue 8, 2009, s. 1141÷1158.
8. Bradski G.: The OpenCV Library, Dr. Dobb's J. of Software Tools, Vol. 25, No. 11, 2000, s. 120÷126.
9. Koszowski G., Michał K.: Modelowanie ludzkiej dłoni na potrzeby rozpoznawania gestów. Studia Informatica, Vol. 33, No. 2B (106), 2012, s. 25÷37.
10. Chetrikov D., Szabo Z.: A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. Proc. 23rd Workshop of the Austrian Pattern Recognition Group, 1999, s. 175÷184.

Wpłynęło do Redakcji 9 stycznia 2013 r.

Abstract

We noticed that in common electronic devices hand pose recognition is often omitted. We have plenty of application for voice recognition, speech recognition, face detection and also building detection based on GPS. There are also games controllers like Kinect or Movie working upon human skeleton or intelligent sensor movements. From that reason in this paper we checked some of the found in literature approaches to human hand pose recognition based on single image.

We have created 3 databases split for reference and test database. Reference database of one the main databases was created with application using only one 3D model. Overall we built databases from 180 images obtained with a help of 9 persons in age between 15 and 79 and 129 images generated from 3D model hand. Databases 1 and 2 contain pictures of 14 gestures, where one reference database is made only from 3D model pictures. Databases 1 and 2 contained visually more different gestures then database 3. Number of images in reference databases were between 28 to 48% of all images creating given database.

In test phase we used algorithms from 3 different approaches: Template Matching, points representing highest curvatures of hand contours and contours distance based methods also a vectors of distances from contour points to palm mass center and Fast Discrete Fourier Transform based upon those vectors. Classification was done with simple Nearest Neighbor algorithms due to scalar result giving back from each feature vectors comparison method. We believe that it`s enough to show a differences between algorithms. In future we will understand an accuracy as a number of properly classified gestures to a number of all gestures.

Presented results shown accuracy of different methods for different kind of databases. In our case reference database containing more representations of gestures then reference database with more models shows a drop in accuracy from 5 to 30% depending on method. For 14 gestures databases we reached accuracy of 86%. We believe it's a good result for such as small reference database (32 images). For 10 times less time consuming algorithm and the same database we obtained accuracy of 84%. For the most populated database the same difference was a 6%.

Overall we think that's presented tests are a good start before more advanced future works with a goal to achieve optimal between speed and accuracy algorithm. Right now we are decided to use vectors from contours to another characteristic hand points, but the next step in our works will be concentrated on fast image characteristic based algorithms used to hand pose recognition. In our opinion they can be used as minor methods to reject part of reference database.

Adres

Grzegorz KOSZOWSKI: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Poland, grzegorz.koszowski@polsl.pl.