

Politechnika Poznańska
Wydział Informatyki



Tomasz Merda
Praca magisterska

**Ocena jakości rozpoznawania gestów statycznych przy użyciu
technologii Microsoft Kinect.**

Promotor: dr inż. Mikołaj Sobczak

Poznań, 2012

Streszczenie

bla bla bla

Abstract

bla bla bla

Spis treści

1.Wprowadzenie.....	1
2.Cel i zakres.....	3
3.Stan wiedzy (SOA).....	4
3.1.Microsoft Kinect.....	4
3.2.Microsoft Kinect SDK.....	5
3.3.Rozwiązania oparte na Microsoft Kinect.....	6
3.3.1.Początki Microsoft Kinect.....	6
3.3.2.Prace oparte na Microsoft Kinect.....	6
4.Środowisko badawcze.....	8
5.System w użyciu.....	9
6.Eksperymenty.....	10
7.Podsumowanie.....	11
8.Literatura.....	12

1. Wprowadzenie

Rozpoznawanie gestów jest podzbiorem komputerowego rozpoznawania wzorców, które to jest jednym z najciekawszych zagadnień związanych z informatyką. Problematyką tą zajmuje się wiele ośrodków na całym świecie od czasu powstania pierwszych komputerów. Najbardziej znane podejścia do rozpoznawania wzorców to:

- rozpoznawanie mowy,
- rozpoznawanie obrazów, w tym:
 - rozpoznawanie pisma,
 - rozpoznawanie kodów (kody kreskowe, kody QR, kody pocztowe),
 - rozpoznawanie znaków drogowych,
 - rozpoznawanie twarzy,
 - rozpoznawanie gestów,
- rozpoznawanie stylu pisania.

Rozpoznawanie gestów w najprostszym scenariuszu wymaga elementu rozpoznawanego (np. człowiek), medium przechwytyjącego dane (np. kamera), systemu przeprowadzającego operacje na tych danych, oraz ośrodka decyzyjnego. W ostatnich latach rozpoznawanie gestów nabrało rozpędu dzięki systematycznie wzrastającej mocy obliczeniowej komputerów osobistych i zastosowaniu rozpoznawania gestów jako nowego kontrolera, zgodnie z maksymą „steruj swoim ciałem”. Najbardziej popularne podejścia do rozpoznawania gestów to (w porządku chronologicznym).

- Playstation EyeToy,
- Nintendo Wii,

- Playstation Move,
- Microsoft Kinect.

Playstation EyeToy było pierwszym udanym podejściem do zastosowania rozpoznawania gestów w grach komputerowych przy użyciu kamery wideo. Problemem była jednak delikatność tego systemu, ponieważ operował w całości w spektrum światła widzialnego i wszystkie jego zaburzenia odbijały się na skuteczności rozpoznawania.

Nintendo Wii zrewolucjonizowało rozpoznawanie gestów poprzez całkowite pozbycie się kamery wideo i wykorzystanie urządzeń, wyposażonych w akcelerometry, trzymane w dłoniach ((ZROBIĆ Z TEGO PRZYPIS) później wyposażone także w inne sensory (np. nacisku w Nintendo Wii Fitness)). Dzięki temu podejściu Nintendo Wii zdecydowanie prowadziło w liczbie sprzedawanych konsol nad swoimi bardziej zaawansowanymi konkurentami – Microsoft Xbox 360 oraz Playstation 3.

Po kilku latach Playstation 3 wypuściło swoje kontrolery „Move”, będące podobnym rozwiązaniem do Wii, z tą różnicą, że wykorzystywały jeszcze dodatkowe świecące kule przy kontrolerach, których poszukiwała kamera wideo. Można zaryzykować stwierdzenie, że doszło do technologicznego połączenia Playstation EyeToy oraz Nintendo Wii w Playstation Move.

Prawdziwa rewolucja rozpoczęła się jednak, gdy Microsoft wypuścił swój kontroler nazwany „Kinect”, pozbywając się „zbędnych” kontrolerów trzymanych w dłoniach. Kinect wyświetla siatkę w podczerwieni

2. Cel i zakres

Celem projektu jest określenie przydatności urządzenia Microsoft Kinect w procesie rozpoznawania gestów oraz dokładności jaką można uzyskać dzięki temu urządzeniu.

W zakres projektu wpisują się następujące punkty:

1. Zapoznanie się z dokumentacją urządzenia Kinect oraz wybór technologii do wykonania aplikacji badawczych,
2. Zapoznanie się z dostępnymi technikami rozpoznawania gestów i wybór odpowiedniego rozwiązania,
3. Przygotowanie aplikacji umożliwiającej rozpoznawanie gestów oraz zbadanie ich jakości,
4. Przeprowadzenie badań określających przydatność i skuteczność urządzenia Kinect w rozpoznawaniu gestów statycznych.
5. Sporządzenie pracy dokumentującej wykonane badania.

Dodatkowo programy rozwijane w ramach projektu mają być demonstracją możliwości Kineka w zakresie:

- sterowania aplikacjami przy użyciu dłoni,
- przygotowania uniwersalnych kontrolerek do obsługi przy pomocy ruchów dłońmi,
- rozpoznawania gestów statycznych opartych na ruchach rąk,
- możliwości manipulowania parametrami rozpoznawania gestów w celu jego kalibracji,
- przygotowania przepływu sterowania w aplikacji, tak aby zminimalizować wykorzystanie standardowych urządzeń wejścia/wyjścia w całym procesie przygotowania i rozpoznania gestów.

3. Stan wiedzy (SOA)

3.1. Microsoft Kinect

Macierz sensorów w urządzeniu Microsoft Kinect zawiera[1, 2]:

- kamerę VGA o rozdzielczości 1280x1024 pikseli i szybkości 30 klatek na sekundę,
- kamerę QVGA o rozdzielczości 320x240 i szybkości 30 klatek na sekundę do pomiaru głębokości,
- promiennik podczerwieni,
- macierz 4 mikrofonów kierunkowych,
- napęd pozwalający na uchylenie głowicy w promieniu $\pm 28^\circ$,
- akcelerometr pracujący w 3 wymiarach.



Rysunek 1: Macierz sensorów w urządzeniu Kinect [3]

3.2. Microsoft Kinect SDK

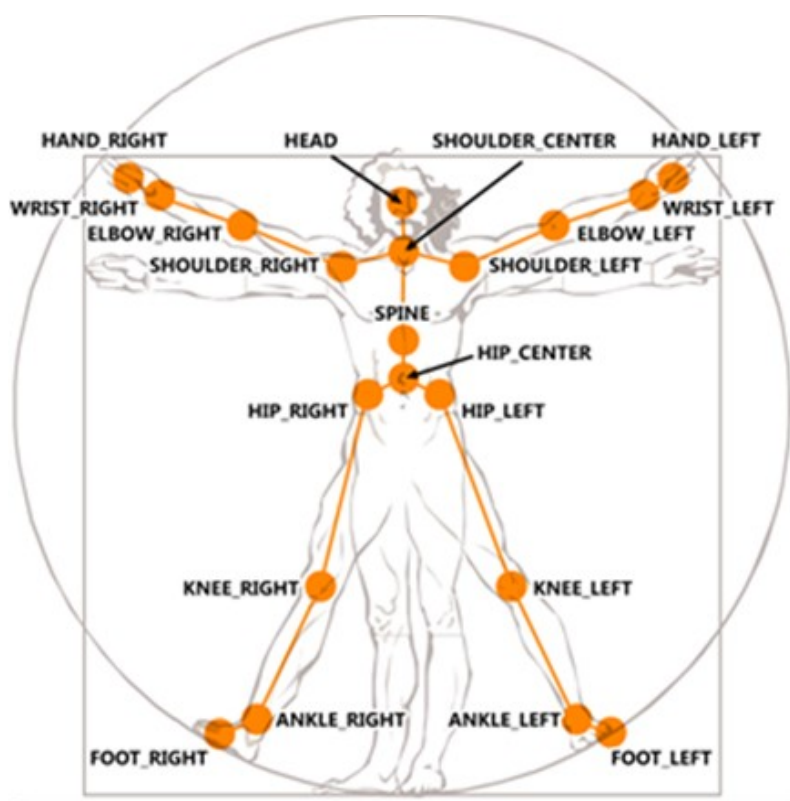
Microsoft Kinect SDK to zestaw narzędzi i bibliotek pozwalający na wykorzystanie urządzenia Kinect w aplikacji. Microsoft wypuścił SDK 16 czerwca 2011 roku, wspierając języki C#, C++ oraz Visual Basic [4].

Najważniejszą częścią SDK są strumienie danych zwracane z macierzy sensorów.

Podstawowy strumień danych pobierany jest z kamery VGA z szybkością 30 FPS. Obraz przekazywany może być w formacie kolorów sRGB, lub YUV i w jednym z kilku wielkości ramek (maksymalnie 1280x1024 pikseli).

Nowością wprowadzoną przez Microsoft jest „strumień głębokości” pobierany z kamery QVGA w ramach wielkości do 640x480 pikseli i z szybkością 30 FPS. Kamera ta działa w spektrum podczerwieni. Specjalne źródło światła w podczerwieni oświetla przestrzeń przed Kinectem, a procesor przy pomocy danych z kamery oblicza odległość w każdym z odczytanych pikseli. Odległość jaką może odczytać sensor wynosi od 0,7 do 6 metrów od urządzenia.

Kolejnym strumieniem jest „strumień szkieletowy”. Kinect na podstawie pierwszych dwóch strumieni rozpoznaje ludzką sylwetkę, nadaje jej identyfikator oraz zwraca pozycję 20 części ciała.



Rysunek 2: Punkty na ciele użytkownika zwracane jako strumień danych przez Kinecta[5]

Współrzędne punktów zwracane są w 3 wymiarach dzięki czemu bez żadnych dodatkowych operacji można uzyskać umiejscowienie sylwetki użytkownika w przestrzeni. Kinect nie pozwala jednak na rozpoznawanie pojedynczych palców, a jedynie dłoni jako całości.

Ostatni strumień danych to strumień dźwięku pochodzący z macierzy mikrofonów. Wbudowane możliwości sensora pozwalają na poprawę jakości dźwięku oraz określenia kierunku jego pochodzenia. Microsoft dostarcza także biblioteki pozwalające na rozpoznawanie głosu¹.

3.3. Rozwiązania oparte na Microsoft Kinect

3.3.1. Początki Microsoft Kinect

Pierwsze próby przejęcia kontroli nad świeżo wypuszczonym urządzeniem w listopadzie 2010 roku zostały podjęte natychmiast po premierze. W przeciągu tygodnia powstały nieoficjalne sterowniki to kontrolowania podstawowych możliwości Kinektu [6]. Następnie zaczęła rozwijać się społeczność wspierająca otwarte oprogramowanie OpenKinect oparte na języku Java. Nie było ono jednak oficjalnie wspierane przez Microsoft przez co nie można było liczyć na reklamację w przypadku uszkodzenia sprzętu. 16 czerwca 2011 roku Microsoft wypuścił własne oprogramowanie – Microsoft Kinect SDK Beta [4]. Jest to jedyne licencjonowane oprogramowanie dla Kinektu i co za tym idzie – nie wiąże się z utratą gwarancji. Od tego czasu lawinowo rośnie liczba projektów opartych na Kineckie. Użytkownicy mogą chwalić się i udostępniać swoje aplikacje na portalach takich jak np. Coding4Fun [7] wspierany przez Microsoft. Licencja dla wersji beta nie pozwala jednak na wykorzystanie Kinektu w celach komercyjnych [8].

3.3.2. Prace oparte na Microsoft Kinect

Microsoft zachęca ośrodki badawcze do rozwijania możliwości Kinektu, gdyż oprogramowanie posiada jeszcze wiele niedociągnięć. Jednym z problemów jest optymalizacja wykorzystania zasobów, gdyż Kinect do obsługi wymaga bardzo mocnej maszyny – dwurdzeniowy procesor o taktowaniu 2,66 GHz i 2 GB pamięci RAM. Microsoft Kinect SDK nie dostarcza też żadnego mechanizmu do rozpoznawania gestów – jedynymi informacjami są strumienie danych opisane w rozdziale 3.2. Właśnie w tym miejscu wpisuje się niniejsza praca, próbując wypracować metodę rozpoznawania gestów.

TODO: napisać coś mądrego na bazie artykułów... np. to

<http://www.creativedistraktion.com/demos/gesture-recognition-kinect-with-hidden-markov-models->

¹ W czasie pisania pracy przez autora dostępny był jedynie język angielski.

[hmms/](#)

4. Środowisko badawcze

Zgodnie z założeniami podanymi w rozdziale 2. wykonano dwie aplikacje Gestures Editor oraz Gestures Recognizer.

4.1. Gestures Editor

Opis

Aplikacja służąca do przygotowania metadanych na temat zbiorów gestów.

Cel programu

Celem programu Gestures Editor jest przygotowanie zbioru gestów do nauki i rozpoznawania gestów, które są przeprowadzane w programie Gestures Recognizer. Szczegółowa funkcjonalność prezentuje się następująco:

- Utworzenie nowego zbioru gestów,
- Nadanie nazwy zbiorowi gestów,
- Wybór liczby gestów w ramach zbioru,
- Wybór liczby wymiarów (2 – X, Y, lub 3 – X, Y, Z) wykorzystywanych do rozpoznania gestu,
- Wybór liczby punktów na ciele zawartych w geście (Gestures Recognizer aktualnie obsługuje jedynie 4 konkretne punkty na ciele – obie dłonie oraz łokcie. Powód takiego stanu rzeczy opisany został w rozdziale [TODO: Podać rozdział]),
- Przyporządkowanie nazwy do konkretnego gestu,
- Zapisanie zbioru gestów do bazy danych,
- Modyfikacji nazw zbiorów gestów oraz gestów w ramach tych zbiorów zapisanych w bazie

danych.

Dane wejściowe

Koncepcja na temat zbioru gestów, które mają być wynikiem wykonania aplikacji Gestures Recognizer.

Dane wyjściowe

Wyjściem aplikacji jest zbiór metadanych na temat gestów, do których w programie Gestures Recognizer przyłączone zostaną informacje identyfikujące same gesty.

Przykładowe dane wyjściowe

Przykładowe dane wyjściowe wyabstrahowane z formatu danych w jakim zostały zapisane prezentować mogą się następująco:

- Zbiór gestów: „Ręce wysoko”,
- Liczba gestów: 3,
- Liczba wymiarów: 3,
- Liczba punktów: 4,
- Nazwa gestu 1: Lewa ręka wysoko,
- Nazwa gestu 2: Prawa ręka wysoko,
- Nazwa gestu 3: Obie ręce wysoko.

Wynikiem będzie zbiór gestów o nazwie „Ręce wysoko” wykorzystująca 4 punkty na ciele (lewa dłoń, lewy łokieć, prawa dłoń, prawy łokieć), w 3 wymiarach (X, Y, Z) i składająca się z 3 gestów.

4.2. Gestures Recognizer

Opis

Aplikacja pozwala na naukę sieci neuronowej gestów na bazie metadanych podanych w aplikacji Gestures Editor, na rozpoznawanie nauczonych gestów oraz ustawienia parametrów pozwalających na ocenę dokładności rozpoznania.

Cel programu

Celem aplikacji jest:

- demonstracja możliwości wykorzystania sensora Kinect do sterowania aplikacją,

- zaproponowanie uniwersalnych kontrolerek do sterowania aplikacją przy użyciu Kinecta,
- możliwość nauczania sieci neuronowej konkretnych gestów,
- możliwość rozpoznania gestu przy pomocy nauczanej sieci neuronowej,
- możliwość manipulowania parametrami określającymi dokładność rozpoznania gestu,
- możliwość określenia dokładności rozpoznania gestu.

Dodatkowe możliwości

- możliwość wyeksportowania nauczanej sieci neuronowej,
- możliwość manipulacji parametrami sieci neuronowej,
- możliwość zmiany języka aplikacji (polski, lub angielski),
- możliwość zapisania zmian w zbiorze gestów w bazie danych,

Dane wejściowe

Metadane dotyczące zbiorów gestów, pobierane z bazy danych.

Dane wyjściowe

Sieci neuronowe nauczone rozpoznawania gestów.

5. System w użyciu

6. Eksperymenty

7. Podsumowanie

8. Literatura

1. ProgrammingGuide_KinectSDK
<http://wenku.baidu.com/view/1e17d3a3b0717fd5360cdce9.html>
2. What's inside a Kinect? <http://kotaku.com/5682075/whats-inside-a-kinect>
3. Macierz sensorów urządzenia Kinect <http://www.generationrobots.com/microsoft-kinect-sensor.us,4,Kinect-Microsoft-Sensor.cfm>
4. Official Kinect SDK released <http://hackaday.com/2011/06/16/official-kinect-sdk-released/>
5. Kinect for Silverlight 5 – Part 2: Skeletal Tracking
<http://www.fdesimoni.ch/post/2011/10/03/Kinect-for-Silverlight-5-Part-2-Skeletal-Tracking.aspx>
6. Microsoft Kinect hacked
<http://www.telegraph.co.uk/technology/microsoft/8129616/Microsoft-Kinect-hacked.html>
7. Coding4Fun <http://channel9.msdn.com/coding4fun>
8. Things you can't do with the Microsoft Kinect SDK
<http://blog.makezine.com/2011/06/17/things-you-cant-do-with-the-microsoft-kinect-sdk/>