



POLITECHNIKA WARSZAWSKA  
Wydział Elektroniki i Technik Informacyjnych  
Instytut Informatyki

Rok akademicki 2013/2014

# PRACA DYPLOMOWA MAGISTERSKA

Michał Aniserowicz

[TYTUŁ]

Praca wykonana pod kierunkiem  
dra inż. Jakuba Koperwasa

Ocena: .....

.....

*Podpis Przewodniczącego Komisji  
Egzaminu Dyplomowego*

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
1.1	Przedmiot pracy . . . . .	2
1.2	Dziedzina problemu . . . . .	2
<b>2</b>	<b>Plan pracy (docelowy spis treści)</b>	<b>9</b>

# Rozdział 1

## Wstęp

### 1.1 Przedmiot pracy

Przedmiotem niniejszej pracy magisterskiej jest aplikacja mobilna umożliwiająca umieszczenie wirtualnego obrazu w rzeczywistej lokalizacji. Przebieg działania aplikacji przedstawia się następująco:

- użytkownik wybiera plik obrazu i nakierowuje kamerę telefonu na miejsce (np. gniazdko elektryczne na ścianie), na której chce go umieścić;
- następnie aplikacja zapamiętuje tło obrazu (np. wspomniane gniazdko elektryczne);
- kiedy użytkownik ponownie wskaże dane miejsce kamerą telefonu, na ekranie urządzenia pojawi się - odpowiednio obrócony i przeskalowany - wybrany obraz.

Aplikacja umożliwia również przechowywanie danych na serwerze, tak aby umieszczone przez danego użytkownika obrazy mogły być oglądane także przez innych użytkowników.

### 1.2 Dziedzina problemu

Aplikacja porusza problemy zawierające się w kilku dziedzinach:

- rozpoznawanie obrazu (rozpoznanie tła, na którym powinien zostać wyświetlony obraz),
- przetwarzanie obrazu (obracanie i skalowanie obrazu),
- komunikacja klient-serwer (przesyłanie obrazu i danych dotyczących jego tła),
- przechowywanie danych (przechowywanie wyżej wymienionych danych w bazie danych).

### 1.2.1 Rozpoznawanie i przetwarzanie obrazu

Aby rozpoznać tło, na którym powinien być wyświetlony obraz, należy wykonać następującą procedurę [1]:

1. dokonać akwizycji obrazu, tj. przetworzenia obrazu obiektu fizycznego do postaci obrazu cyfrowego;
2. wykonać wstępne przetwarzanie obrazu, tj. zredukować zniekształcenia powstałe podczas akwizycji;
3. dokonać segmentacji obrazu, tj. wydzielenia poszczególnych obiektów zawartych w obrazie;
4. przeprowadzić analizę uzyskanych segmentów i rozpoznać poszukiwany obiekt.

#### Akwizycja obrazu

Istnieje kilka sposobów uzyskiwania obrazów cyfrowych. Niektóre z nich, to [2]:

- kamera video,
- cyfrowy aparat fotograficzny,
- skaner,
- ręczne stworzenie obrazu przy pomocy programu graficznego.

Jako że aplikacja uruchamiana będzie na urządzeniach typu smartfon, akwizycja obrazu będzie odbywać się poprzez wykorzystanie aparatu cyfrowego zamontowanego w danym urządzeniu.

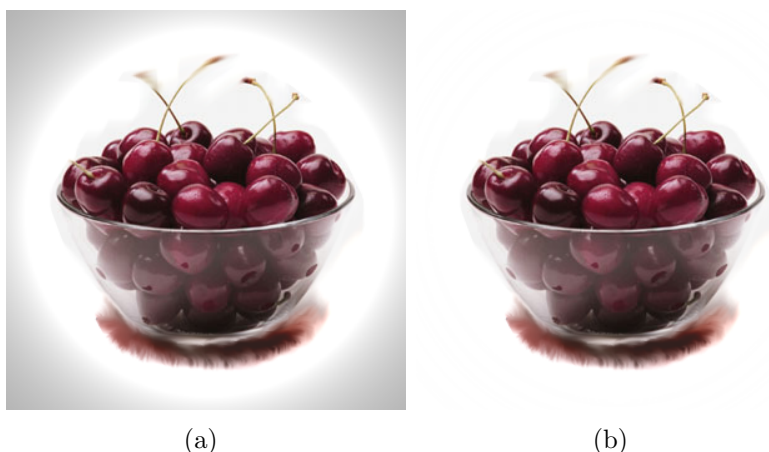
#### Przetwarzanie wstępne

Przetwarzanie wstępne polega na redukcji zniekształceń radiometrycznych i geometrycznych występujących w obrazie [2].

Przekształcenia radiometryczne powstają na skutek nierównomiernego oświetlenia fotografowanego obiektu lub są wynikiem błędów występujących podczas akwizycji obrazu. Można je korygować stosując korekcję sumacyjną lub korekcję iloczynową [1]. Rysunek 1.1 przedstawia przykład zniekształconego obrazu.

Przyczyną powstawania zniekształceń geometrycznych jest najczęściej nieprawidłowy sposób wykonania zdjęcia (np. zbyt ostry kąt “patrzenia” kamery lub zbyt mała odległość pomiędzy obiektywem a fotografowanym obiektem) lub ekstremalne warunki towarzyszące jego wykonaniu (np. zdjęcia satelitarne) [4].

Najczęściej występującymi zniekształceniami będą z pewnością zniekształcenia geometryczne spowodowane różnicą pomiędzy orientacją urządzenia (tj. obrotem kamery) w chwili umieszczenia, a jego orientacją w chwili wyświetlenia obrazu. W celu niwelacji tych zniekształceń wykorzystane zostaną dane dotyczące orientacji telefonu w obu tych chwilach.



Rysunek 1.1: Przykład zniekształceń radiometrycznych: (a) obraz zniekształcony; (b) obraz skorygowany.

## Segmentacja

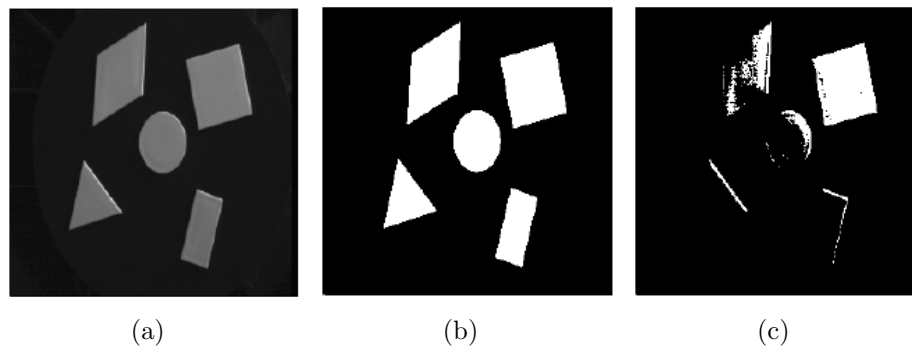
Segmentacja obrazu to operacja polegająca na wydzieleniu z obrazu obszarów, które cechują się pożądanymi właściwościami (np. mają jednolitą barwę) [2]. Dostępne metody segmentacji to [3]:

- metody punktowe:
  - progowanie,
  - klasteryzacyjne,
  - metody krawędziowe,
- metody obszarowe:
  - rozrost obszarów,
  - łączenie obszarów,
  - podział obszarów,
  - metoda podziału i łączenia,
  - segmentacja wododziałowa,
- metody hybrydowe (złożenie kilku metod).

Przykład poprawnej i niepoprawnej segmentacji obrazuje Rysunek 1.2.

## Analiza i rozpoznanie

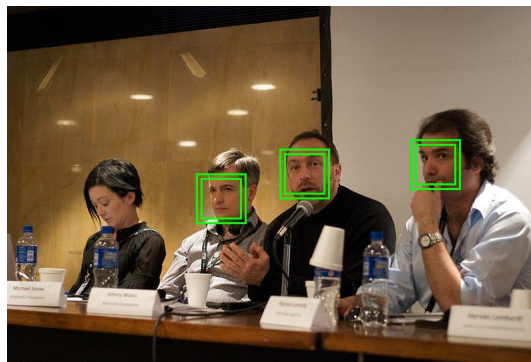
Analiza i rozpoznanie obrazu polega na przeanalizowaniu wydzielonych segmentów obrazu i zidentyfikowaniu wśród nich tego zawierającego poszukiwany obiekt. Klasyfikacja metod rozpoznawania obrazów wyszczególnia następujące grupy metod [3]:



Rysunek 1.2: Przykład segmentacji: (a) obraz oryginalny; (b) obraz po poprawnej segmentacji; (c) obraz po niepoprawnej segmentacji. Źródło: [2].

- metody minimalnoodległościowe,
- metody wzorców,
- metody aproksymacyjne,
- metody specjalne,
- metody probabilistyczne.

Przykład fotografii, na której rozpoznano ludzkie twarze, przedstawia Rysunek 1.3.



Rysunek 1.3: Przykład rozpoznawania ludzkich twarzy na fotografii. Źródło: [http://en.wikipedia.org/wiki/File:Face\\_detection.jpg](http://en.wikipedia.org/wiki/File:Face_detection.jpg)

Aplikacja korzystać będzie z jednej lub kilku metod w celu podjęcia decyzji, czy (i gdzie) na obrazie z kamery urządzenia wyświetlić zapamiętany obraz.

Dodatkowo, wyświetlany obraz powinien być odpowiednio przetworzony, tak aby użytkownik miał wrażenie, że rzeczywiście znajduje się on na wskazanej powierzchni. Aby uzyskać pożądany efekt, obraz należy poddać odpowiednim przekształceniom geometrycznym. Dostępne przekształcenia to między innymi [2]:

- przesunięcia,

- obroty,
- skalowanie,
- odbicia.

W tym przypadku wykorzystać należy obroty i skalowanie.

### Obracanie obrazu

Wyświetlany obraz powinien zostać obrócony, tak aby zniwelować różnicę pomiędzy orientacją urządzenia w chwili umieszczenia, a jego orientacją w chwili wyświetlenia obrazu.

### Skalowanie obrazu

Skalowanie, czyli zmiana rozmiaru wyświetlanego obrazu [4], powinno zostać wykonane, aby użytkownik mógł odnieść wrażenie zbliżania się do obrazu lub oddalania się od niego.

Współczynnik przeskalowania obrazu powinien być ustalony na podstawie porównania rozmiarów obiektów oryginalnego tła obrazu z ich rozmiarami na obrazie odczytanym przez kamerę urządzenia - współczynnik przeskalowania wyświetlanego obrazu powinien być równy stosunkowi tych rozmiarów.

Przykład obróconego i przeskalowanego obrazu widoczny jest na Rysunku 1.4.



Rysunek 1.4: Przykład obróconego i przeskalowanego obrazu: (a) obraz oryginalny; (b) obraz obrócony i przeskalowany.

## 1.2.2 Komunikacja klient-serwer

Komunikacja pomiędzy klientem a serwerem może być zrealizowana przy użyciu różnych protokołów, takich jak:

- UDP (User Datagram Protocol) [5] - bezpołączeniowy (a więc zawodny) protokół komunikacyjny,

- TCP (Transmission Control Protocol) [6] - połączeniowy (niezawodny) protokół komunikacyjny,
- HTTP (Hypertext Transfer Protocol) [7, 8], jako *web service*<sup>1</sup> wykorzystujący jeden z poniższych sposobów dostępu do danych:
  - SOAP (Simple Object Access Protocol) [9] - protokół wykorzystujący język XML, nie posiada ustandaryzowanego interfejsu (API),
  - REST (Representational State Transfer) [10] - wzorzec definiujący ustandaryzowany interfejs dostępu do danych, wspiera przesyłanie danych w języku XML lub formacie JSON,
  - OData (Open Data Protocol) [11] - protokół umożliwiający wygodne konstruowanie ustandaryzowanych zapytań o dane.

Ponieważ komunikacja wewnątrz aplikacji odbywać się będzie według schematu żądanie-odpowiedź, strona serwerowa z powodzeniem może zostać zrealizowana jako *web service*. W chwili obecnej najbardziej popularne platformy umożliwiające tworzenie tego typu rozwiązań to .NET Framework (Microsoft)<sup>2</sup> i Java EE (Oracle)<sup>3</sup>.

## .NET

Technologia WCF będąca częścią platformy .NET wspiera zarówno SOAP, jak i REST wraz z OData (jako rozszerzenie REST).

## Java EE

Platforma Java EE posiada dwa narzędzia przeznaczone do budowy serwisów:

- JAX-WS (Java API for XML Web Services) - wspiera serwisy wykorzystujące protokół SOAP,
- JAX-RS (Java API for RESTful Web Services) - służy to tworzenia serwisów opartych na REST (ang. *RESTful*).

Nie posiada ona natywnego wsparcia dla protokołu OData - istnieją jednak implementacje przygotowane przez firmy trzecie: Google<sup>4</sup> i Apache<sup>5</sup>.

### 1.2.3 Przechowywanie danych

Aplikacja będzie przechowywać dane różnego rodzaju. Będą to między innymi:

- pliki obrazów umieszczanych przez użytkowników w wybranych miejscach,

---

<sup>1</sup>[https://pl.wikipedia.org/wiki/Usługa\\_internetowa](https://pl.wikipedia.org/wiki/Usługa_internetowa)

<sup>2</sup>Narzędzie Windows Communication Foundation (WCF)

<sup>3</sup>Narzędzia: Java API for XML Web Services (JAX-WS) i Java API for RESTful Web Services (JAX-RS)

<sup>4</sup>OData4J (<http://code.google.com/p/odata4j/>)

<sup>5</sup>Apache Olingo (<http://olingo.incubator.apache.org/documentation.html>)



- pliki zawierające tła tych obrazów,
- dane dotyczące właścicieli tych obrazów,
- dane dotyczące lokalizacji i orientacji telefonu w chwili umieszczenia obrazu w wybranym przez użytkownika miejscu.

O ile pliki graficzne mogą rezydować na dysku serwera bez konieczności utrzymywania ich w uporządkowanej strukturze, o tyle pozostałe dane powinny być zgrupowane i uporządkowane. Najpopularniejszym sposobem przechowywania uporządkowanej struktury danych w tego typu aplikacjach jest wykorzystanie bazy danych.

# Rozdział 2

## Plan pracy (docelowy spis treści)

1. Wstęp
2. Przegląd literatury (i dostępnych aplikacji tego typu)
3. Metody wstępnego przetwarzania obrazu
  - (a) Metoda 1
  - (b) Metoda 2
  - (c) Metoda 3
4. Metody segmentacji
  - (a) Metoda 1
  - (b) Metoda 2
  - (c) Metoda 3
5. Metody rozpoznania obrazu
  - (a) Metoda 1
  - (b) Metoda 2
  - (c) Metoda 3
6. Wybór markera
  - (a) Metoda 1
  - (b) Metoda 2
  - (c) Metoda 3
  - (d) Własny algorytm
  - (e) Porównanie skuteczności metod
7. Rozpoznanie markera
  - (a) Metoda 1

- (b) Metoda 2
- (c) Metoda 3
- (d) Własny algorytm
- (e) Porównanie skuteczności metod

8. Projekt aplikacji

- (a) Architektura aplikacji
- (b) Użyte algorytmy wyboru i rozpoznania markera
- (c) Komunikacja klient-serwer
- (d) Przechowywanie danych
- (e) Skuteczność aplikacji w porównaniu z innymi aplikacjami tego typu

9. Podsumowanie

10. Bibliografia

# Bibliografia

- [1] T. Pavlidis. (Warszawa 1987). *Grafika i przetwarzanie obrazów: algorytmy*. WNT.
- [2] Tadeusiewicz R., Korohoda P. (Kraków 1997). *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji. ISBN 83-86476-15-X.
- [3] Tadeusiewicz R., Flasiński M. (Warszawa 1991). *Rozpoznawanie obrazów*. PWN. ISBN 83-01-10558-5.
- [4] Schneider P. J., Eberly D. H. (USA 2003). *Geometric Tools for Computer Graphics*. Morgan Kaufmann Publishers. ISBN 1-55860-594-0.
- [5] *RFC 768 - User Datagram Protocol*; tools.ietf.org; dostęp: sierpień 2013; w:  
<http://tools.ietf.org/html/rfc768>
- [6] <http://tools.ietf.org/html/rfc793>
- [7] <http://www.w3.org/Protocols/>
- [8] <http://tools.ietf.org/html/rfc2616>
- [9] <http://www.w3.org/TR/soap12-part1/>
- [10] [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- [11] <http://www.oodata.org/>

## OŚWIADCZENIE

Oświadczam, że Pracę Dyplomową pod tytułem “[TYTUŁ]”, którą kierował dr inż. Jakub Koperwas, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....

Michał Aniserowicz