

Politechnika Warszawska
Wydział Elektroniki i Technik Informatycznych
Instytut Automatyki i Informatyki Stosowanej
Zakład Sterowania i Systemów

PRACOWNIA DYPLOMOWA MAGISTERSKA I
SPRAWOZDANIE

Mateusz Zięba
Nr albumu: 188911

**Rozpoznawanie gestów twarzy w sekwencji
obrazów cyfrowych**

Opiekun naukowy
prof nzw. dr hab. inż. Włodzimierz Kasprzak

Spis treści

1. Wstęp.....	3
Zakres pracowni dyplomowej magisterskiej I.....	3
Zakres sprawozdania.....	3
2. Wybór tematyki pracy.....	3
Tematyka pracy.....	3
Implementacja aplikacji.....	3
Cele pracy dyplomowej magisterskiej.....	3
3. Zadania w bieżącym semestrze.....	4
4. Przewidywane prace w przyszłym semestrze.....	4
5. Koncepcja rozwiązania.....	4
6. Analiza obrazów twarzy.....	5
Wykrywanie oczu.....	6
7. Analiza sygnału dźwiękowego.....	8
Obliczanie częstotliwości podstawowej F_0	8
8. Środowisko implementacji.....	9
Środowisko Eclipse RCP.....	9
JFace.....	10
Java Media APIs.....	10
Java Advanced Imaging (JAI).....	11
Java MediaFramework (JMF).....	11
Java sound API.....	11
9. Plan pracy magisterskiej.....	11
10. Literatura.....	12

1. Wstęp

• **Zakres pracowni dyplomowej magisterskiej I**

Zasadniczym celem pracowni problemowej magisterskiej była analiza oraz ustalenie założeń i zakresu pracy magisterskiej. Tematyka pracowni obejmowała również rozpoznanie podstawowych problemów, analizę literatury oraz algorytmów przydatnych podczas realizacji pracy. Zakres pracowni dyplomowej objął także analizę sposobów rozpoznania gestów twarzy oraz narzędzi i bibliotek programistycznych pomocnych w implementacji tych sposobów.

• **Zakres sprawozdania**

Niniejsze sprawozdanie stanowi abstrakt najważniejszych zagadnień opracowanych w ramach pracowni problemowej magisterskiej I..

2. Wybór tematyki pracy

• **Tematyka pracy**

Tematyka pracy dyplomowej magisterskiej obejmuje analizę sekwencji obrazów cyfrowych, zawierających nagranie wypowiadającego się człowieka. W wyniku tej analizy dokonane będzie rozpoznanie uczuć mówcy. Przeprowadzać ją będzie specjalnie stworzona aplikacja, która wykryje kształt oraz położenie ważniejszych części ciała ludzkiej twarzy. Analizowane będą usta oraz oczy. Na podstawie analizy dźwięków wypowiedzi osób wyrażających różne ekspresje, wzajemnego ułożenia, ruchów oraz zmian kształtów ich części twarzy wyodrębnione zostanie kilka klas uczuć, takich jak strach, radość, podekscytowanie, smutek, złość. Następnie do wyodrębnionych cech klas ludzkich ekspresji porównywane będą nowe nagrania, w celu określenia w jakim stanie emocjonalnym znajdują się umieszczeni na nich mówcy.

Zdolność rozpoznawania uczuć rozmówcy przez komputery, może okazać się niezbędna do swobodnego prowadzenia rozmów człowieka z maszyną.

• **Implementacja aplikacji**

Aplikacja zostanie napisana przy użyciu języka Java, oraz przy wykorzystaniu technologii Eclipse Client Platform. Do analizy, przetwarzania oraz prezentacji sygnału wideo (obraz i dźwięk) zostaną użyte biblioteki Java Advanced Imaging (JAI), Java MediaFramework (JMF) oraz Java sound API. Interfejs użytkownika wykonany zostanie przy użyciu biblioteki JFace.

• **Cele pracy dyplomowej magisterskiej**

- zapoznanie z technikami rozpoznania obrazów ludzkiej twarzy
- zapoznanie z technikami analizy sygnału dźwiękowego ludzkiej wypowiedzi
- rozpoznanie na obrazie twarzy oczu oraz ust
- analiza zmian położenia i kształtu ust oraz oczu w sekwencji obrazów cyfrowych
- wyodrębnienie na podstawie analizy filmów wideo z wypowiedziami różnych osób kilku klas uczuć

- przyporządkowanie do jednej z wyodrębnionych klas badanej sekwencji obrazów (rozpoznanie rodzaju uczucia wyrażanego przez nagraną osobę)
- zapoznanie z technikami tworzenia aplikacji typu „gruby klient” (rich client) przy użyciu języka Java

3. Zadania w bieżącym semestrze

- Ustalenie i sprecyzowanie tematu pracy dyplomowej
- Wybór technologii realizacji pracy
- Wstępny plan pracy dyplomowej
- Analiza literatury
- Opracowanie szkieletu (wstępny projekt) aplikacji
- Implementacja podstawowych algorytmów przetwarzania obrazów
- Implementacja algorytmu pomocniczego do detekcji twarzy w obrazie – analiza histogramów
- Implementacja algorytmów analizy częstotliwości podstawowej F_0 sygnału mowy

4. Przewidywane prace w przyszłym semestrze

- Opracowanie ostatecznej koncepcji rozwiązania
- Kompletny plan pracy dyplomowej
- Kompletny projekt aplikacji
- Opracowanie planu testów aplikacji
- Implementacja podstawowej funkcjonalności aplikacji
- Przeprowadzenie doświadczeń i analiza działania systemu

5. Koncepcja rozwiązania

Analizie poddawane będą nagrania audio-wideo przedstawiające wypowiadającą się osobę. Obrazy oraz sygnał dźwiękowy poddawany będzie wstępnej obróbce. Klatki filmu przeskalowywane będą do rozdzielczości 640 na 480 pikseli. Paleta kolorów konwertowana będzie do 256 kolorowej skali szarości. Sygnał dźwiękowy konwertowany będzie do sygnału monofonicznego o 16 bitowej rozdzielczości o częstotliwości 22050 Hz.

Na wstępnie obrobionych obrazach twarzy mówcy zostaną rozpoznane jej elementy takie jak oczy, nos, usta i brwi. Analizowany będzie ich kształt, położenie oraz ruchy. Z sygnału dźwiękowego wypowiedzi wyodrębnione będą podstawowe jego cechy, takie jak częstotliwość podstawowa oraz energia dźwięku.

W serii eksperymentów z badaniem nagrań osób znajdujących się w różnych stanach emocjonalnych, na podstawie wyodrębnionych w danej chwili, wyodrębnionych z obrazu i sygnału dźwiękowego cech, utworzone zostanie kilka klas uczuć, takich jak złość, radość, smutek, zaskoczenie, wstręt.

Po wyodrębnieniu klas uczuć, możliwe będzie badanie nowych nagrań przedstawiających ludzkie

wypowiedzi. Cechy badanego nagrania porównywane będą z cechami charakterystycznymi wyodrębnionych klas, w celu ustalenia stanu emocjonalnego wypowiadającej się na nim osoby. Spodziewany wynik analizy przedstawia poniższy rysunek.



Ilustracja 1: Spodziewany wynik analizy uczuć mówcy. Rozpoznane uczucia: złość, wstręt, zadowolenie, zaskoczenie. [2]

6. Analiza obrazów twarzy

W celu umożliwienia analizy sekwencji obrazów konieczne jest spełnienie kilku warunków technicznych. Twarz osoby musi znajdować się w środku zarejestrowanego nagrania i musi być jego głównym obiektem. Musi być ona dobrze i jednolicie oświetlona. Tło nagrania musi być jednolite, jasne. Powyższe założenia eliminują konieczność wykrywania twarzy na każdym z obrazów. Konieczne jest jedynie wykrycie poszczególnych elementów twarzy.

Nagrania muszą być wykonywane z rozdzielczością co najmniej 640x480 pikseli i obejmować widok czołowy twarzy, nie obrócony w żadnej płaszczyźnie.

Przed przystąpieniem do analizy sekwencji obrazów konieczna jest wstępna obróbka każdego z nich. Każdy obraz jest (w razie konieczności) przeskalowywany do rozmiarów 640 x 480 pikseli. Skala kolorów jest konwertowana na 256 stopniową skalę szarości. Obróbka wstępna obrazu obejmuje również zastosowanie filtra medianowego, w celu usuwania zakłóceń losowych, których poziom intensywności znacznie odbiega od poziomu intensywności punktów sąsiednich.. Kształt

(X, plus, kwadrat) i rozmiar filtra (3x3 lub 5x5 pikseli) zostanie dobrany doświadczalnie w miarę postępu prac. Przykładowe działanie filtra medianowego pokazują poniższe ilustracje:



Ilustracja 2: obraz zaszumiony



Ilustracja 3: obraz po zastosowaniu filtra medianowego 5x5

• Wykrywanie oczu

Znalezienie środków oczu na obrazie twarzy odbywa się poprzez obliczenie gradientu wartości pikseli całego obrazu oraz zsumowanie tych wartości w pionie i w poziomie:

$$H_j = \sum_{i=1}^k |o_{i,j} - o_{i,j+1}|$$

$$V_i = \sum_{j=1}^l |o_{i,j} - o_{i+1,j}|$$

gdzie:

o – obraz

H_j – suma gradientów wartości pikseli w poziomie, dla j-tego wiersza

V_i – suma gradientów wartości pikseli w pionie, dla i-tej kolumny

i – indeks numeru wiersza

j – indeks numeru kolumny

k – wysokość obrazu

l – szerokość obrazu

Poprzez wyznaczenie wartości maksymalnej wektora H otrzymujemy współrzędną pionową oczu. Wektor V dzielony jest na 2 połowy. Wartość maksymalna każdej z nich wyznacza współrzędną poziomą odpowiednio lewego i prawego oka.

$$y = \max_{i \in \{1 \dots k\}} (H)$$

$$x_1 = \max_{j \in \{1 \dots \frac{l}{2}\}} (V)$$

$$x_2 = \max_{j \in \{\frac{l}{2} + 1 \dots l\}} (V)$$

H – wektor suma gradientów wartości pikseli w poziomie

V – wektor suma gradientów wartości pikseli w pionie

i – indeks numeru wiersza

j – indeks numeru kolumny

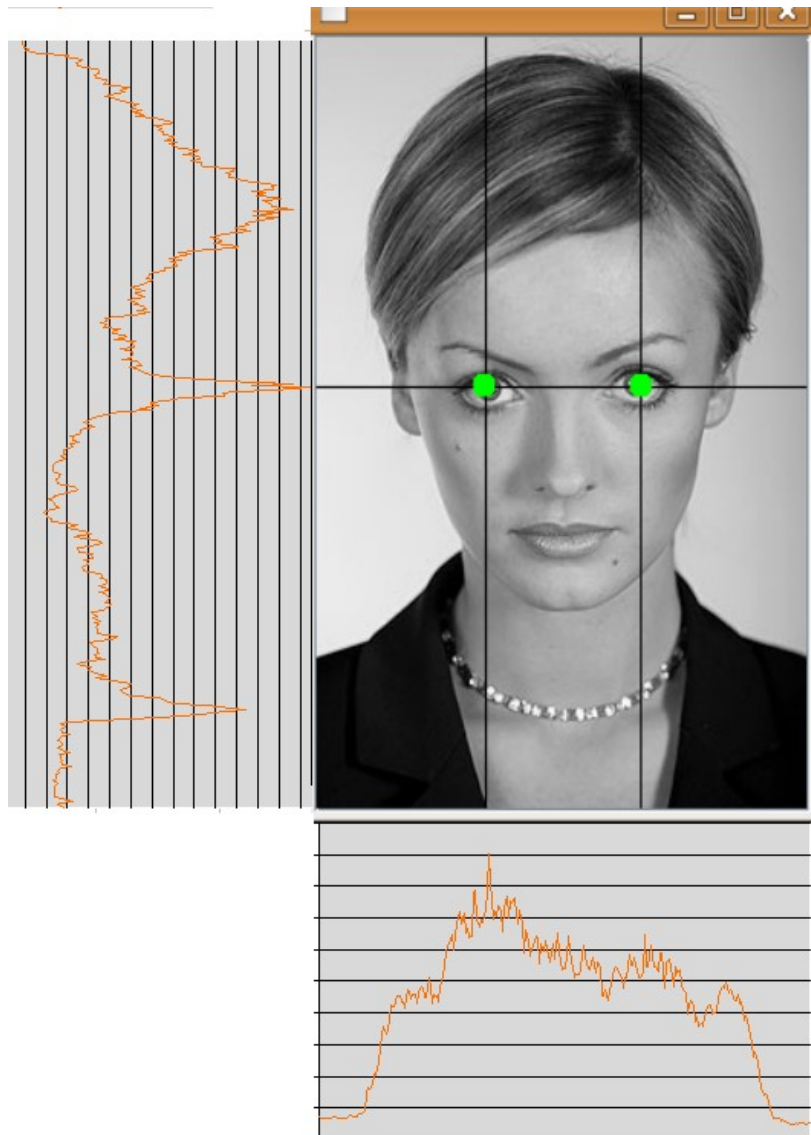
k – wysokość obrazu

l – szerokość obrazu

$[x_1, y]$ – współrzędne prawego oka

$[x_2, y]$ – współrzędne lewego oka

Działanie algorytmu przedstawia poniższa ilustracja:



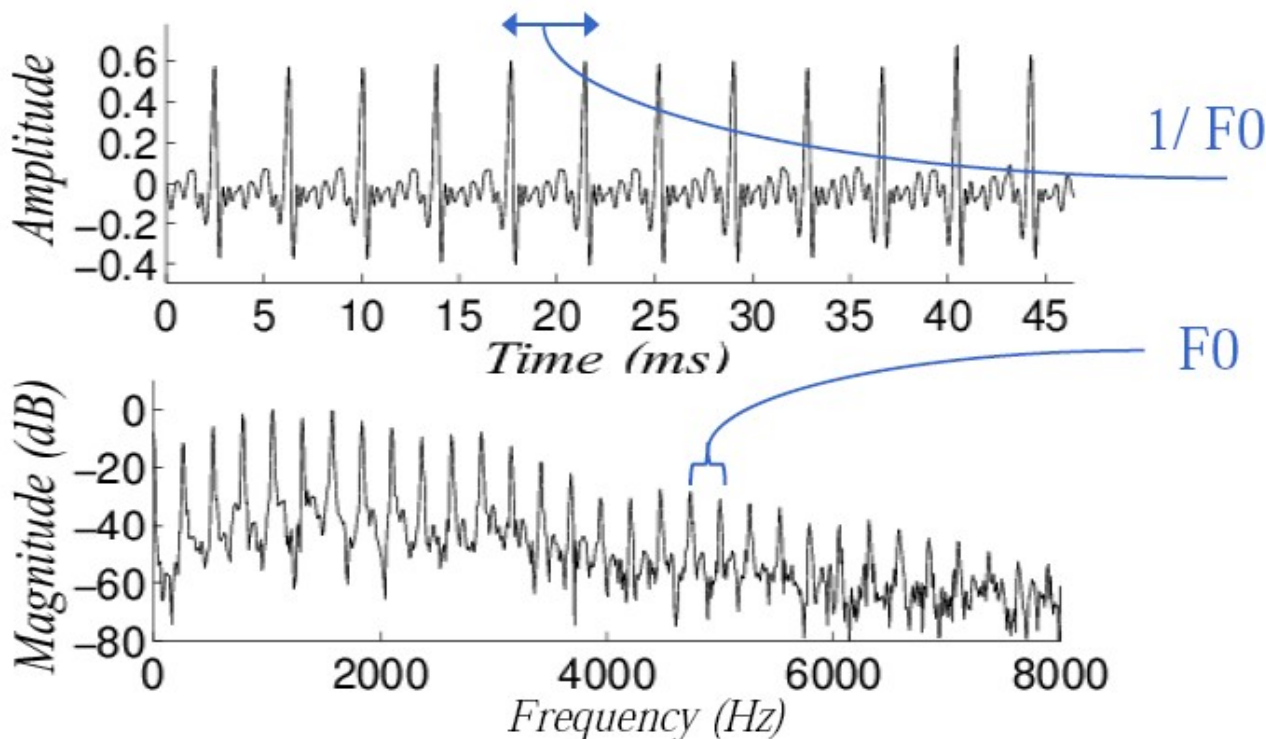
*Ilustracja 4: działanie algorytmu wykrywania oczu.
Na wykresach sumy gradientów dla poszczególnych wierszy
i kolumn.*

W przyszłej pracy przedstawiona metoda zostanie prawdopodobnie zastąpiona inną, bazującą na kształcie oczu. Przedstawiona metoda ma charakter pomocniczy.

7. Analiza sygnału dźwiękowego

• Obliczanie częstotliwości podstawowej F_0

Mowa ludzka jest okresowym sygnałem dźwiękowym. Jej częstotliwość podstawowa ograniczona jest do zakresu od 50 do 400 Hz. Jest ona charakterystyczna dla wypowiadającej go osoby, ale może ulegać zmianom, wraz ze zmianami stanu emocjonalnego mówcy. Częstotliwość podstawową F_0 w dziedzinie czasu i częstotliwości przedstawia poniższa ilustracja.



Ilustracja 5: Częstotliwość podstawowa w dziedzinie czasu i częstotliwości. [6]

Jedną z metod wyznaczania podstawowej częstotliwości sygnału jest metoda autokorelacji. Jest to metoda, która wylicza korelację sygnału z tym samym sygnałem przesuniętym o φ chwil czasowych. Autokorelację sygnału można obliczyć zgodnie ze wzorem:

$$r_t(\varphi) = \sum_{j=t+1}^{t+W} x_j + x_{j+\varphi}$$

gdzie:

φ - przesunięcie sygnału

t - początek okna sygnału

W - szerokość okna sygnału

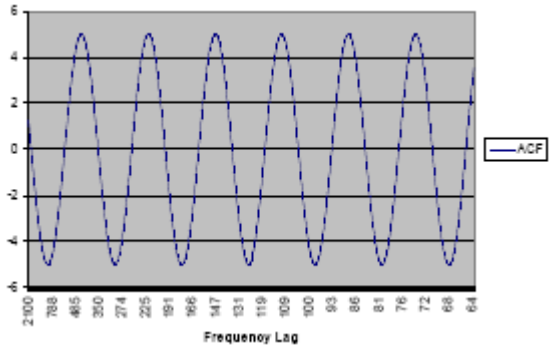
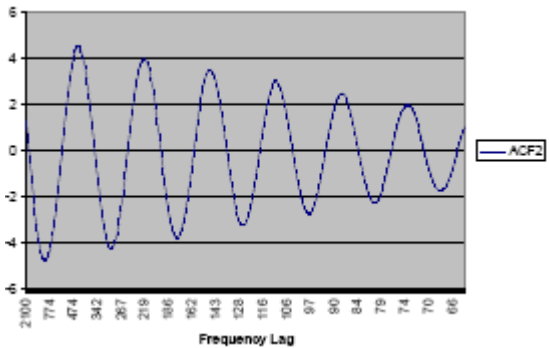
x - dyskretna wartość sygnału

Funkcja autokorelacji największą wartość przybiera dla $\varphi = 0$ i odpowiada wówczas krótkookresowej wartości energii którą wyliczono wcześniej

Niewielka modyfikacja powyższej funkcji:

$$r'_t(\varphi) = \sum_{j=t+1}^{t+W-\varphi} x_j + x_{j+\varphi}$$

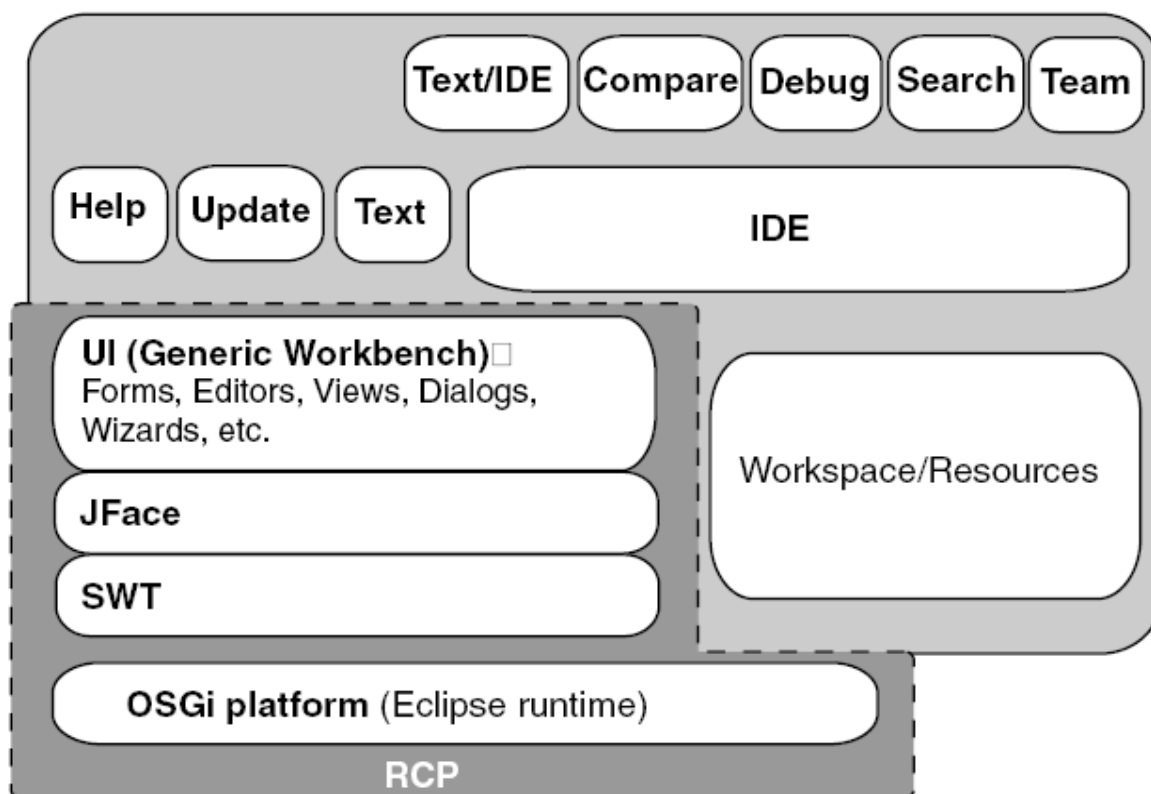
nadaje niższym częstotliwościom wyższe wagi. Porównanie wyników generowanych przez obie funkcje [8] oraz przykładowe implementacje powyższych algorytmów w języku Java prezentuje poniższa tabela:

$r_t(\varphi) = \sum_{j=t+1}^{t+W} x_j + x_{j+\varphi}$	$r'_t(\varphi) = \sum_{j=t+1}^{t+W-\varphi} x_j + x_{j+\varphi}$
<pre> int length = data.length; int size = length / 2; double[] r = new double[size]; for (int t = 0; t < size; t++) { double sum = 0; for (int n = 1; n < length-t; n++) { double val = (data[n]*data[n+t])/(double)length; sum += val; } r[t] = sum; } return r; </pre>	<pre> int length = data.length; int size = length / 2; double[] r = new double[size]; for (int t = 0; t < size; t++) { double sum = 0; for (int n = 1; n < size; n++) { double val = (data[n]*data[n+t])/(double)length; sum += val; } r[t] = sum; } return r; </pre>
	

8. Środowisko implementacji

• Środowisko Eclipse RCP

Eclipse Software Development Kit to jedno z najlepszych i najpopularniejszych zintegrowanych środowisk programistycznych (IDE). Eclipse SDK składa się z rdzenia, mającego za zadanie ładowanie i uruchamianie wtyczek oraz zbioru samych wtyczek, takich jak Java Development Tools (zestaw narzędzi pomocnych przy pisaniu aplikacji w języku Java <http://eclipse.org/jdt>) czy Plugin Development Environment (zestaw narzędzi ułatwiający tworzenie wtyczek <http://eclipse.org/pde>). Ilustracja 6 obrazuje architekturę środowiska Eclipse . Zbiór komponentów niezbędnych do ładowania i uruchamiania wtyczek wyodrębniono w projekcie Eclipse Rich Client Platform (RCP <http://eclipse.org/rcp>). System wtyczek stosowany w środowisku Eclipse bazuje na specyfikacji OSGi (Open Service Gateway Initiative <http://osgi.org>).



Ilustracja 6: architektura środowiska Eclipse [7]

Na bazie RCP możliwe jest nie tylko utworzenie środowiska programistycznego, ale także innych aplikacji typu „gruby klient”, nie mających nic wspólnego z programowaniem. Niewątpliwą zaletą takich aplikacji jest ich modułarna budowa oraz możliwość integracji z innymi aplikacjami napisanymi przy wykorzystaniu tej technologii. W ten sposób końcowi użytkownicy mogą łatwo dostosowywać produkt do swoich indywidualnych wymagań.

Ułatwia to sam proces tworzenia takiego oprogramowania, zwłaszcza dużych aplikacji klienckich. Poszczególne komponenty programu mogą być projektowane niezależnie, przez różne zespoły w tym samym czasie, a ich integracja przeprowadzona dopiero w końcowych fazach projektu. Eclipse RCP pełni nie tylko rolę serwera OSGi. Dostarcza także kompleksowy system zarządzania interfejsem użytkownika, umożliwiającą przenoszenie i przyklejanie poszczególnych elementów, dostarczając komponenty interfejsu użytkownika oraz system zarządzający cyklem życia całej aplikacji.

- **JFace**

<http://wiki.eclipse.org/index.php/JFace>

Biblioteka pozwalająca na tworzenie interfejsu użytkownika w języku Java. Aplikacje wykonane przy użyciu tej biblioteki łączą przenośność oferowaną przez sam język, z wysoką wydajnością, ponieważ wykorzystują natywne komponenty systemu operacyjnego na którym działa aplikacja.

- **Java Media APIs**

<http://java.sun.com/javase/technologies/desktop/media/>

Technologia opracowana przez firmę Sun w skład której wchodzi interfejsy programowania

aplikacji zwiększające możliwości operowania na multimediami w języku Java. W jej skład wchodzi pakiety przeznaczone do obsługi dwu i trój-wymiarowej grafiki (Java 2D i Java 3D), zapisu i odczytu obrazów ([Java Image I/O](#)), przetwarzania obrazów ([Java Advanced Imaging](#)) oraz przetwarzania filmów i dźwięku (Java Media Framework). Podczas realizacji pracy dyplomowej magisterskiej szczególnie pomocne mogą okazać się technologie: Java Advanced Imaging oraz Java Media Framework.

–Java Advanced Imaging (JAI)

<http://java.sun.com/javase/technologies/desktop/media/jai/>

Interfejs programowania aplikacji (API) wprowadzający obiektowe metody umożliwiające przeprowadzanie zaawansowanych operacji na obrazach. JAI wprowadza funkcjonalność, która pozwala wydajne, intuicyjne oraz łatwo rozszerzalne przetwarzanie obrazów. JAI potrafi, w celu zwiększenia wydajności przetwarzania korzystać z natywnych funkcji systemu, na którym przetwarzanie jest wykonywane.

–Java MediaFramework (JMF)

<http://java.sun.com/products/java-media/jmf>

Interfejs programowania aplikacji (API) wprowadzający obsługę dźwięku oraz obrazu wideo w aplikacjach napisanych w języku Java. Pakiet umożliwia nagrywanie i odtwarzanie multimediiów w wielu formatach. JMF wprowadza także obsługę mediów strumieniowych. JMF, podobnie jak JAI potrafi, w celu zwiększenia wydajności przetwarzania korzystać z natywnych funkcji systemu, na którym to przetwarzanie jest wykonywane.

• *Java sound API*

<http://java.sun.com/products/java-media/sound/>

Interfejs programowania (API) niskiego poziomu umożliwiający kontrolę nad procesami nagrywania i odtwarzania dźwięku. API wprowadza również zestaw funkcji pozwalających na generowanie oraz przetwarzanie sygnału audio oraz dźwięków MIDI.

9. Plan pracy magisterskiej

1. Wstęp
2. Środowisko implementacji
3. Wykrywanie elementów twarzy
4. Analiza sygnału dźwiękowego
5. Projekt aplikacji
6. Prezentacja (opis instalacji i przykład użycia)
7. Testowanie aplikacji
8. Opracowanie systemu rozpoznającego stan emocjonalny badanej osoby
9. Przebieg doświadczeń i analiza działania systemu
10. Wnioski końcowe

Dodatek A: Literatura

Dodatek B: Zawartość płyty CD

10.Literatura

- [1] W. Kasprzak: Rozpoznawanie obrazów i sygnałów mowy.
- [2] Stan Z. Li, Anil K. Jain: „Handbook of Face Recognitions”, Springer 2005
- [3] Matthew Scarpino, Stephen Holder, Stanford NG i Laurent Mihalkovic; „Swt JFace In Action” Manning Publications 2005
- [4] Janusz Bobulski: „Metoda identyfikacji użytkownika w oparciu o fuzję transformacji falkowej i ukrytych modeli Markowa”, praca doktorska, Politechnika Częstochowska
- [5] Paris Smaragdis, Michael Casey: „Audio/visual independent components”, 4th International Symposium of Independent Component Analysis and Blind Signal Separation (ICA 2003)
- [6] Klapuri: „Fundamental frequency estimation” ISMIR Graduate School, October 2004
- [7] Berthold Daum: „Eclipse 3 for Java Developers”; ISBN: 0-470-02005-9; Wrox 2005,
- [8] Abdreu Moedinger: „Sing-a-ring: a custom ring tone creation system for mobile platforms”; University DeLand, Florida