

Robert Wielgat*, Marek Gorgoń**

Nauczanie technik multimedialnych w środowisku MATLAB 6.5 z wykorzystaniem aplikacji TMT

1. Wprowadzenie

Rozwój technik przetwarzania sygnału doprowadził w ostatnim 15-leciu do burzliwego rozwoju i upowszechnienia stosowania różnorodnych technik multimedialnych. Pojęcie multimediiów [3, 7] jest nie do końca sprecyzowane, jakkolwiek kryją się za tym pojęciem środki sprzętowe i programistyczne służące do integracji, przetwarzania i przekazu różnorodnych mediów, tj. tekstów i grafiki komputerowej, rzeczywistych obrazów statycznych, dźwięku, filmów i interaktywnych animacji komputerowych.

Wraz z rozwojem technik multimedialnych pojawiła się potrzeba ich efektywnego nauczania. Analiza pozycji literaturowych dotyczących zagadnień multimedialnych wskazuje, że większość z nich omawia jedynie warstwę praktyczną, związaną z wykorzystaniem gotowych narzędzi – głównie informatycznych – do tworzenia różnorodnych plików multimedialnych. Odczuwa się wyraźny brak literatury, zwłaszcza w języku polskim, omawiającej warstwę teoretyczną współczesnych multimediiów. Warstwa ta jest bowiem daleko bardziej złożona, niż wynikałoby z zakresu teorii omawianej w podstawowych podręcznikach do cyfrowego przetwarzania sygnałów i obrazów [2, 11]. Nadto podręczniki te omawiają nierzadko poszczególne zagadnienia w sposób rozcłonkowany, a istotą współczesnych multimediiów jest stosowanie wielu różnorodnych metod przetwarzania sygnału łącznie, ze szczególnym uwzględnieniem dopasowania tych metod do cech ludzkiej percepcji i sposobu analizy sygnału w ludzkim mózgu. Jako przykłady niech posłużą tu wszelkie metody kompresji obrazu dopasowane do mechanizmu ludzkiego widzenia i oparte o eliminację wyższych częstotliwości sygnału wizyjnego, czy też redukcję liczby bitów chrominancji w modelu YUV. Nie inaczej, jak tylko z uwzględnieniem psychoakustycznego modelu ludzkiego słuchu możliwe jest ograniczenie pasma kodowanego sygnału w popularnej kompresji MP3. Na gruncie wspomnianych modeli biologicznych, w nieustannym dążeniu do zachowania równowagi pomiędzy jakością sygnału a stopniem kompresji, realizowane są wciąż nowe pomysły techniczne formułowane w międzynarodowe standardy takie, jak:

* Państwowa Wyższa Szkoła Zawodowa, Tarnów

** Akademia Górniczo-Hutnicza, Kraków

JPEG, JPEG 2000, MPEG-1, MPEG-2, MPEG-4, DV (formalnie IEC-61834). Niemniej nie sposób znaleźć podręczników akademickich przystępnie i gruntownie przybliżających te zagadnienia studentom. To samo można powiedzieć o pomocach naukowych służących do nauczania wspomnianych zagadnień. Są natomiast dostępne nieliczne książki o charakterze teoretycznym [8, 10] oraz omówienia na stronach internetowych [9]. Celem niniejszej publikacji jest pokazanie narzędzia informatycznego, powstałego w Państwowej Wyższej Szkole Zawodowej w Tarnowie w okresie 2002–2005, służącego do nauczania przedmiotu techniki multimedialne. Narzędzie zrealizowano w formie programu obejmującego zestaw ćwiczeń laboratoryjnych dla studentów kierunku informatyka stosowana. Program był tworzony głównie w ramach prac dyplomowych [1, 4, 5, 6].

Na początku prac zdecydowano o stworzeniu aplikacji komputerowej w oparciu o środowisko MATLAB w wersji podstawowej – tj. bez korzystania z funkcji pakietów specjalistycznych (*toolbox*) – mającej na celu nauczanie zarówno teoretycznych podstaw, jak i aspektów praktycznych wybranych technik multimedialnych dotyczących cyfrowego przetwarzania obrazów, kompresji obrazów oraz cyfrowego przetwarzania sygnału dźwiękowego. Niniejszy artykuł przedstawia główne założenia programu oraz podaje przykłady ćwiczeń realizowanych za pomocą programu.

1.1. Założenia programu TMT

Program TMT (*Tarnów Multimedia Toolbox*) ma na celu nauczanie metod i algorytmów stosowanych w wybranych technikach multimedialnych. Program został zaimplementowany w środowisku MATLAB 6.5. Środowisko MATLAB posiada wiele swoistych zalet, które czynią go oprogramowanie szczególnie przydatnym w dydaktyce.

Wśród tych zalet należy wymienić:

- prosty i czytelny język programowania, zbliżony do zapisu matematycznego,
- brak konieczności kompilowania programu;
- ograniczenie do minimum procedur związanych ze specyfiką języka, jak np. brak konieczności alokacji tablic, deklaracji zmiennych itp.;
- duży wybór efektywnie działających wbudowanych funkcji;
- bogate możliwości i prosty sposób prezentowania wyników obliczeń.

Wszystkie te zalety sprawiają, że osoba chcąca zaimplementować konkretny algorytm w środowisku MATLAB-a, po okresie zapoznania się z funkcjonowaniem narzędzia, skupia się głównie na pokonywaniu trudności ze zrozumieniem algorytmu, a nie pokonywaniu trudności związanych ze składnią i filozofią języka programowania. Ta sama osoba dysponuje również bogatymi możliwościami usprawnienia procesu tworzenia algorytmu poprzez korzystanie z szerokiego wyboru wbudowanych funkcji oraz korzystanie z gotowych wygodnych narzędzi do liczbowej i graficznej prezentacji wyników działania algorytmu.

Głównymi wadami środowiska MATLAB jest długi czas wykonywania się zaimplementowanych procedur oraz zajmowanie dużych obszarów pamięci podczas wykonywania bardziej złożonych algorytmów. Jednak, jeżeli wziąć pod uwagę wielkości pamięci RAM

stosowane we współczesnych komputerach oraz założone cele dydaktyczne, wówczas wymienione wady środowiska MATLAB tracą na znaczeniu.

Założono, że tworzone narzędzia programowe będą przeznaczone do realizacji zajęć laboratoryjnych. Napisane oprogramowanie ma stanowić środowisko – swoiste ramy – umożliwiające implementacje algorytmów, pobranie i wizualizację danych oraz wykonanie operacji pomocniczych służących wizualnej, odsłuchowej oraz jakościowej i ilościowej ocenie wyniku. W obecnej wersji oprogramowania prowadzący zajęcia ma możliwość formułowania konkretnych zadań w formie drukowanej instrukcji do ćwiczeń laboratoryjnych. Wybrane moduły zawierają elementy interaktywnej nawigacji, odsyłacze, podpowiedzi, sygnalizację błędów (niezależnie od błędów sygnalizowanych przez środowisko MATLAB) i podobne. Formuła oprogramowania jest otwarta. Można tworzyć nowe, poszerzać istniejące i w razie potrzeby dokonywać ograniczeń tematycznych w proponowanych ćwiczeniach. Oprogramowanie (stworzony toolbox) używa języka polskiego i polskich liter.

Główne założenie dydaktyczne polega na możliwości konfrontacji przez studenta zwizualizowanych wyników działania funkcji zrealizowanej w oprogramowaniu, której składnia jest dla studenta niedostępna, z efektami działania funkcji zadanej mu do napisania. Student otrzymuje szkielet funkcji zadanej (w odpowiednim folderze toolbox) wraz z opisem lub wskazówkami dotyczącymi działania algorytmu. W realizowanym procesie dydaktycznym wiadomości teoretyczne przekazywane są w trakcie wykładu. W trakcie ćwiczeń laboratoryjnych wykonywane są 2–4 zadania o wzrastającym poziomie trudności. Oprogramowanie umożliwia również pracę w trybie sprawdzania wiadomości. W trybie tym nie jest możliwe oglądanie wyników funkcji przygotowanych w oprogramowaniu ani podpowiedzi. Jest możliwe natomiast wykorzystywanie pomocy środowiska MATLAB.

2. Zasady użytkowania aplikacji

Program posiada strukturę okienkową. Dużą uwagę zwrócono na zapewnienie możliwości pracy indywidualnej. Przy uruchamianiu programu pojawia się okienko logowania, w którym student wpisuje swoje dane (imię, nazwisko, numer albumu). Po wpisaniu tych danych jest tworzony profil użytkownika, tzn. folder o nazwie takiej jak nr albumu w folderach studenckich. We wspomnianym folderze są umieszczane m-pliki studenta pisane przez niego w trakcie zajęć. Przy normalnej pracy laboratorium zakłada się możliwość dostępu do stworzonych danych przez użytkownika folderu w ciągu całego semestru.

Po zalogowaniu się student przechodzi do okienka wyboru ćwiczenia. Jak już wspomniano dobór zadań w poszczególnych ćwiczeniach określa prowadzący w instrukcji do ćwiczenia.

Zadania zrealizowane w aplikacji na obecnym etapie prac są związane z następującymi zagadnieniami technik multimedialnych:

- zmianą rozdzielczości obrazu,
- zmianą liczby poziomów szarości obrazu,
- binaryzacją obrazu,
- tworzeniem, wyrównywaniem i rozciąganiem histogramu obrazu,

- filtracją liniową obrazu,
- filtracją nieliniową obrazu,
- dwuwymiarową dyskretną transformatą Fouriera,
- odwrotną dwuwymiarową transformatą Fouriera,
- standardami kolorów (RGB, CMY, YUV),
- zmianą palety kolorów w obrazach indeksowanych,
- dwuwymiarową dyskretną transformatą kosinusową,
- odwrotną dwuwymiarową transformatą kosinusową,
- kompresją JPEG,
- kompresją GIF,
- kwantyzacją równomierną, nierównomierną i różnicową sygnału dźwiękowego,
- cyfrową zmianę częstotliwości próbkowania sygnału dźwiękowego,
- kodowaniem DPCM sygnału dźwiękowego,
- kompresją MP3.



Rys. 1. Standardowa plansza programu do przetwarzania obrazów

Po przejściu do wybranego ćwiczenia pojawiają się plansze umożliwiające wykonanie zadań objętych danym ćwiczeniem. Zaprojektowano szereg plansz adekwatnych do poszczególnych grup ćwiczeń. Przykładową planszę do ćwiczeń z zakresu przetwarzania ob-

razów pokazano na rysunku 1. Plansza umożliwia wizualizację danych wejściowych i przetworzonych, uruchomienie funkcji oprogramowania oraz funkcji studenta, wykonywanie czynności pomocniczych i używanie narzędzi toolboxa. Menu oraz inne elementy planszy zostały zaplanowane w taki sposób, aby przypominały elementy znane z innych popularnych programów komputerowych, dzięki czemu obsługa programu TMT może być w wysokim stopniu intuicyjna.

2.1. Grupa ćwiczeń do przetwarzania i kompresji obrazów

Dla pokazanej na rysunku 1 planszy do przetwarzania obrazu w obszarze roboczym są widoczne dwa pola:

- 1) pole obrazu wejściowego, w którym wyświetla się obraz wczytany z pliku lub macierzy;
- 2) pole obrazu wyjściowego, w którym jest wyświetlany obraz przetworzony za pomocą danego przekształcenia.

Przekształcenia obrazu dokonuje się za pomocą zielonej lub niebieskiej strzałki. Niebieską strzałką uruchamia się funkcję napisaną przez studenta. Jak już wspomniano, każdej funkcji pisanej przez studenta odpowiada analogiczna funkcja zaimplementowana w programie TMT, która jest uruchamiana za pomocą zielonej strzałki. Jeżeli efekty przetwarzania za pomocą funkcji studenta są identyczne z efektami przetwarzania za pomocą funkcji programu TMT, wówczas można uznać, że zadanie zostało wykonane poprawnie. Ocena zgodności może być wykonana za pomocą narzędzi pomocniczych takich, jak: histogram, bezpośredni odczyt pikseli, porównywanie dwóch obrazów [4]. Na dokładniejsze omówienie zasługuje narzędzie do porównywania obrazów. W swej obecnej formie umożliwia ono odczyt wartości odpowiadających sobie pikseli dwóch obrazów, wyświetlanie obrazu błędu między obrazami, wyświetlanie histogramu różnicy obrazów, wyświetlanie różnicy histogramów, obliczanie PSNR między dwoma obrazami.

W celu lepszej wizualizacji obrazu zastosowano jego normalizację według wzoru

$$E(i, j) = M * [G(i, j) - F(i, j)] + shift \quad (1)$$

gdzie:

- $F(i, j)$ – obraz pierwszy,
- $G(i, j)$ – obraz drugi,
- $E(i, j)$ – obraz błędu,
- M – mnożnik różnicy,
- $shift$ – przesunięcie pikseli w skali szarości.

Wprowadzenie funkcji studenta następuje poprzez wywołanie standardowego okna użytkownika oprogramowania MATLAB. Po wywołaniu student ma możliwość wyboru funkcji do napisania. Po wybraniu funkcji do napisania pojawia się okienko edycyjne, w którym zawarty jest szablon funkcji, ewentualnie ostatnio zachowana wersja tworzonej funkcji. W szablonie funkcji oprócz narzuconego przez program TMT nagłówka znajduje

się również obszerny komentarz wyjaśniający działanie funkcji, a czasami wskazówki do jej tworzenia. W przypadku bardziej złożonych procedur w szablonie może znaleźć się mniej lub bardziej obszerny fragment kodu. Przykładowe okno do tworzenia funkcji na filtrację liniową obrazu za pomocą maski 3×3 jest pokazane na rysunku 2.

```

1  function obraz_wy=stud_Maska9(obraz_we,maska9)
2
3  %Tutaj proszę napisać funkcję realizującą:
4  %FILTR LINIOWY - MASKA 3x3
5  %Zmienne w programie:
6  %obraz_we - obraz wejściowy
7  %maska9 - maska 9 elementowa (macierz [1,9]). Elementy w macierzy [1,9] są ustawione w następujący
8  %sposób: [pierwszy_wiersz_maski_3x3  drugi_wiersz_maski_3x3  trzeci_wiersz_maski_3x3];
9  %obraz_wy - obraz wyjściowy
10
11 obraz_we=double(obraz_we);           %zamiana liczb typu uint8 na liczby typu double
12 LW=size(obraz_we,1);                 %LW - liczba wierszy obrazu wejściowego
13 LK=size(obraz_we,2);                 %LK - liczba kolumn obrazu wejściowego
14 temp=zeros(LW+2,LK+2);               %zadeklarowanie powiększonej macierzy z obrazem
15 temp(2:LW+1,2:LK+1)=obraz_we;       %wpisanie obrazu wejściowego w środek powiększonej macierzy
16 temp(1,2:LK+1)=obraz_we(1,:);       %powielenie pierwszego wiersza
17 temp(LW+2,2:LK+1)=obraz_we(LW,:);   %powielenie ostatniego wiersza
18 temp(:,1)=temp(:,2);                 %powielenie pierwszej kolumny
19 temp(:,LK+2)=temp(:,LK+1);          %powielenie ostatniej kolumny
20
21 %*****
22 % ***** MIEJSCE DO PISANIA DLA STUDENTA *****
23
24
25
26
27
28 % ***** KONIEC NIC WIECEJ NIE ZMIENIAC *****
29 %*****

```

Rys. 2. Okno edytora funkcji studenta z z wyświetlonym kodem filtra liniowego z maską o rozmiarze 3×3

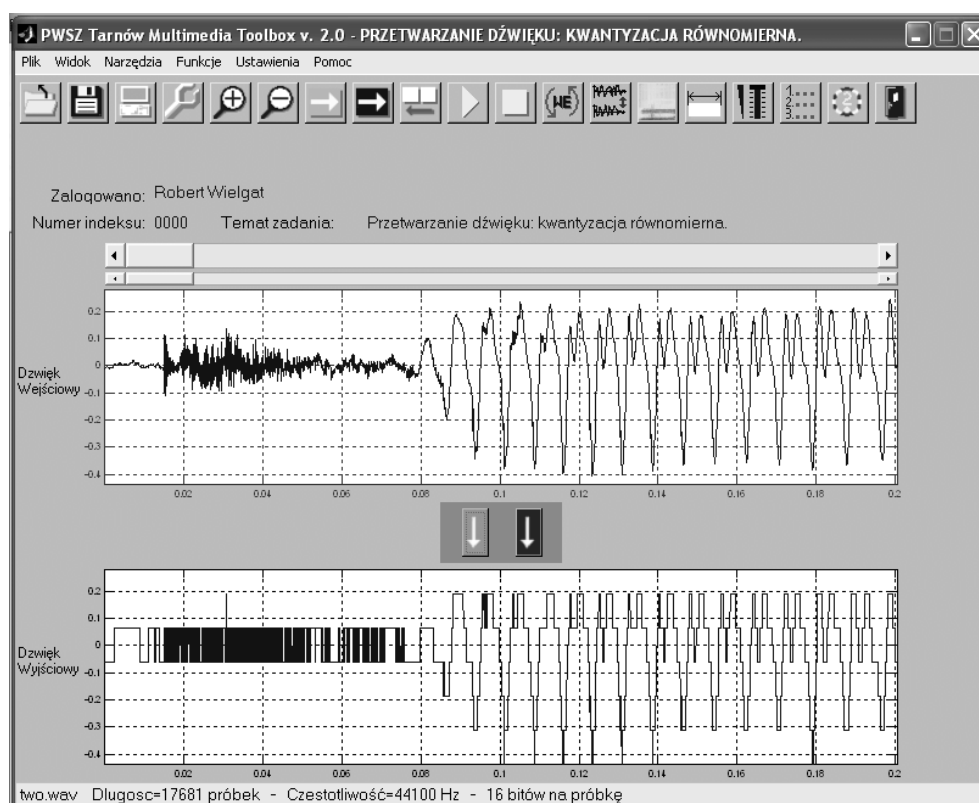
Tworzone przez studenta funkcje są zapisywane w jego osobistym katalogu o unikalnej nazwie będącej numerem albumu studenta. Wiele tworzonych przez studenta na danym etapie nauki funkcji może być wykorzystanych w późniejszych etapach. Przykładowo funkcję splotu (konwolucji) można wykorzystać do budowy filtra kombinowanego, funkcję obliczającą DCT obrazu 8×8 można wykorzystać w procedurze dokonującej kompresji JPEG, itp.

W procesie tworzenia funkcji nie do uniknięcia są różnego rodzaju błędy. W programie TMT dołożono wielu starań, aby zapewnić w miarę wygodną obsługę błędów. W obecnej formie programu TMT większość błędów jest obsługiwana za pomocą okienek ostrzegawczych. Komunikaty pojawiające się w okienkach mają naprowadzać studenta na właściwe rozwiązanie.

2.2. Grupa ćwiczeń do przetwarzania dźwięku

Program TMT umożliwia również prowadzenie ćwiczeń z przetwarzania dźwięku, jakkolwiek lista możliwych do wykonania ćwiczeń jest na obecnym etapie prac znacznie uboższa w porównaniu z zestawem ćwiczeń do przetwarzania obrazów. Ćwiczenia związane z przetwarzaniem dźwięku są wykonywane w oparciu o planszę pokazaną na rysunku 3.

W przypadku ćwiczeń dotyczących przetwarzania dźwięku konieczne jest wyposażenie stanowiska ćwiczeniowego w słuchawki. Stosowanie głośników multimedialnych nie jest wskazane, ze względu na możliwość wzajemnego zagłuszenia się odtwarzanych dźwięków na sąsiednich stanowiskach.



Rys. 3. Standardowa plansza programu do przetwarzania sygnału dźwiękowego

Na planszy do przetwarzania sygnału dźwiękowego pojawiają się ikony uruchamiające narzędzia do przetwarzania dźwięku. Narzędzia te dają możliwość sterowania odtwarzaniem: odsłuchiwanie wybranego kanału lub obydwu kanałów, oryginalnego i przetworzonego sygnału dźwiękowego, zatrzymywanie odtwarzania dźwięku, regulacji głośności i za-

znaczenia fragmentu dźwięku w celu jego odsłuchania lub późniejszego zapisania. Ponadto stworzono narzędzia do wizualizacji spektrogramu sygnału dźwiękowego oraz do porównywania sygnałów dźwiękowych.

2.3. Prezentacja algorytmów „krok po kroku”

Oprócz wykonania ćwiczeń z zakresu algorytmów wykorzystywanych w technikach multimedialnych studenci mają również możliwość zapoznania się z bardziej złożonymi procesami jak np. metody kompresji obrazów i dźwięków. Obecnie studenci mają możliwość zapoznania się z kompresją JPEG metodą „krok po kroku” (rys. 4) [6] oraz w ograniczonym zakresie z kompresją MP3 [5]. Ponieważ plansza posiada jak na razie charakter prezentacyjny, nie ma na niej ikon pozwalających studentowi pisać własne funkcje.

The screenshot shows a software interface for demonstrating the JPEG compression process step-by-step. It includes:

- Image and Block Selection:** Two windows showing the original image and a selected 8x8 pixel block. A 'Zawartość wybranego bloku' table shows the pixel values.
- DCT Coefficients:** A table labeled 'Współczynniki DCT' showing the transformed coefficients.
- Quantization:** A 'Wybrano blok nr.: 786' section with a 'Wskaż blok' button and a 'Kwantyzacja z zadaną jakością' dropdown. Below it is a 'Tablica kwantyzacji' and a 'Wynik kwantyzacji' table.
- Huffman Coding:** A 'Rekonstrukcja' section with a list of Huffman codes, a 'Zygak' list, and parameters for DC and AC coefficients.

Rys. 4. Plansza do prezentacji kompresji JPEG „krok po kroku”

Jednym z podstawowych celów prezentacji algorytmów metodą „krok po kroku” jest przedstawienie danej metody w postaci schematu blokowego oraz sugestywna wizualizacja kolejności wykonywania poszczególnych operacji. W związku z powyższym poszczególne bloki, ewentualnie pośrednie wyniki przetwarzania, pojawiają się na ekranie sekwencyjnie. Każdy kolejny blok przetwarzania jest wywoływany po naciśnięciu przez studenta wy-

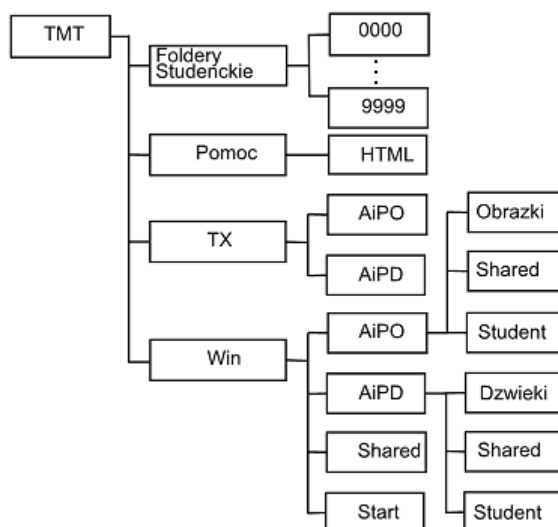
świecanych kolejno ikon z zieloną strzałką. Pod strzałką znajduje się przycisk z interaktywnym znakiem zapytania, po naciśnięciu którego pojawia się informacja na temat kolejnego etapu przetwarzania.

Planszę do kompresji JPEG metodą „krok po kroku” przedstawiono na rysunku 4. Po naciśnięciu zielonej strzałki wykonywany jest kolejny etap kompresji oraz pojawiają się na planszy pośrednie wyniki przetwarzania. Na nieco inną technikę prezentacji zdecydowano się przy realizacji narzędzia „krok po kroku” dla kompresji MP3. Różnica polega na prezentacji poszczególnych bloków kompresji, a nie pośrednich wyników przetwarzania, ponieważ wyników przetwarzania jest zbyt wiele, żeby można było zamieścić je na jednej planszy. Obserwacja pośrednich wyników przetwarzania jest możliwa dopiero po naciśnięciu odpowiedniego aktywnego przycisku na danym bloku kompresji MP3.

3. Struktura aplikacji

3.1. Struktura katalogów i nazewnictwo plików

Strukturę katalogów programu TMT przedstawiono na rysunku 5. Katalog główny *TMT* umieszcza się w katalogu *Toolbox* programu MATLAB 6.5.



Rys. 5. Struktura katalogów programu TMT

W katalogu *Foldery studenckie*, każdy student umieszcza pliki stworzone w trakcie zajęć w swoim osobistym katalogu. Pliki tworzone przez studenta mają z góry narzucone nazwy i zaczynają się od przedrostka *_stud*. Student może również w swoim osobistym folderze przechowywać przetworzone obrazy, dźwięki lub inne zmienne. W programie TMT zostały częściowo zaimplementowane mechanizmy zapobiegające przed

omyłkowym zapisywaniem plików stworzonych przez studenta do cudzego katalogu. Program nie jest natomiast zabezpieczony przed celowym zapisywaniem plików w lokalizacji innej niż osobisty folder studenta.

W katalogu *Pomoc* znajdują się m-pliki będące plikami pomocy w programie. Znajduje się tam również folder *HTML*, gdzie w przyszłości będą umieszczone strony internetowe działające w trybie off-line i stanowiące multimedialną pomoc do programu TMT.

W katalogu *TX* są zawarte pliki, które zawierają funkcje programu – odpowiedniki tworzonych przez studenta funkcji. Wspomniane pliki zaczynają się od przedrostka *tx_*. Katalog *TX* podzielono na dwa podkatalogi. W podkatalogu *AiPO* znajdują się funkcje do analizy i przetwarzania obrazów. Natomiast w podkatalogu *AiPD* do analizy i przetwarzania dźwięku.

W katalogu *Win* znajdują się pozostałe pliki, z których większość jest wykorzystywana do tworzenia struktury okienkowej programu. W podkatalogu *Start* umieszczono pliki umożliwiające uruchomienie aplikacji. Katalog *Shared* zawiera funkcje wykorzystywane do obsługi okienek programu oraz narzędzia zarówno do analizy i przetwarzania obrazów, jak i do analizy i przetwarzania dźwięku. Nazwy wszystkich plików znajdujących się w tym folderze zostały poprzedzone przedrostkami *tmt*, podobnie jak nazwy plików znajdujące się w katalogu *Pomoc* oraz niektórych plików w folderze *Start*.

Wewnątrz folderu *Win* znajdują się także katalogi *AiPO* i *AiPD*, zawierające pliki wykorzystywane przez poszczególne moduły. Znajdują się w nich pliki tworzące główne plansze programu. Wszystkie z nich rozpoczynają się od tekstu *Okno...* Pliki tworzące wszystkie okna pomocnicze rozpoczynają się od tekstu *pOkno...* Pliki tworzące narzędzie do wczytywania funkcji użytkownika noszą nazwę *WyborFunkcjiUzytkownika - kaxxxx*, gdzie *xxxx* stanowi skrót od nazwy dla planszy głównej ćwiczenia. W katalogach tych znajdują się także foldery *Student*, które zawierają wszystkie szablony funkcji pisanych przez użytkownika. W katalogach *Shared* znajdują się te pliki, które są wykorzystywane przez wszystkie plansze wchodzące w skład jednego modułu. Pozostałe katalogi – *Obrazki*, *Dźwięki* – zawierają pliki, które użytkownik może przetwarzać w ramach ćwiczeń.

Wprowadzenie uporządkowanej struktury katalogów oraz standaryzowane nazewnictwo funkcji ma na celu ułatwienie przyszłej rozbudowy programu TMT.

4. Rozbudowa programu TMT

4.1. Zasady rozbudowy programu TMT

Jak już wspomniano w poprzednim rozdziale, struktura programu TMT została tak zaplanowana, aby w maksymalnym stopniu ułatwić proces rozbudowy programu. Wszystkie główne plansze programu zostały stworzone za pomocą *GUI Design Environment*, pozwalającym na zbudowanie w prosty sposób interfejsu użytkownika. Rozbudowa programu polega zatem na modyfikacji szablonu planszy, który posiada już wiele gotowych narzędzi, jak np. zoom, wyświetlanie histogramu, porównywanie obrazu itp. w przypadku planszy do przetwarzania obrazu; odtwarzanie dźwięku, wyświetlanie spektrogramu, regulację głośności itp. w przypadku planszy do przetwarzania dźwięku.

4.2. Perspektywiczne plany rozbudowy

Pomimo niewątpliwych zalet programu TMT pozostało jeszcze wiele zadań do zrealizowania zarówno dotyczących obsługi programu, jak i objętości prezentowanego materiału.

Najważniejszym zadaniem do zrealizowania związanym z podniesieniem jakości obsługi programu jest zbudowanie sprawniejszego systemu pomocy w porównaniu z już istniejącym. W systemie pomocy znajdzie się przede wszystkim zestaw instrukcji do ćwiczeń oraz ogólna pomoc, w której ważnym elementem będzie wykonanie multimedialnych animacji (z wykorzystaniem technologii FLASH, JAVA) prezentujących poszczególne algorytmy.

Kolejnym zadaniem związanym z obsługą programu będzie zbudowanie sprawnego narzędzia do testowania wiedzy i umiejętności studentów, które w swych założeniach ma umożliwiać automatyczne podsumowywanie i ewidencjonowanie wyników oraz będzie zabezpieczone przed nieuczciwymi sposobami rozwiązywania testów.

Plany rozbudowy programu obejmują również opracowywanie kolejnych zagadnień merytorycznych. Przewidywane jest dodanie kolejnych plansz dotyczących standardów kompresji audio i wideo, jak również poszerzenie już istniejących plansz o nowe funkcje.

5. Podsumowanie

W artykule przedstawiono oryginalną koncepcję nauczania technik multimedialnych w środowisku MATLAB 6.5 z wykorzystaniem programu zrealizowanej aplikacji TMT. Aplikacja zapewnia możliwość nauczania zarówno podstaw teoretycznych, jak i aspektów praktycznych szerokiego spektrum zagadnień multimedialnych, poczynając od najprostszych algorytmów takich jak np. binaryzacja jednoprogowa, a kończąc na zaawansowanych technikach kompresji obrazów i dźwięków.

Aplikacja umożliwia prezentację działania algorytmów, daje możliwość pisania własnych funkcji, umożliwia obserwację sygnałów źródłowych i przetworzonych. Możliwe jest porównanie wyników działania algorytmów zaimplementowanych we wbudowanych funkcjach programu TMT, jak również napisanych przez studenta. Zrealizowano narzędzia umożliwiające jakościową i ilościową ocenę zgodności sygnałów przetworzonych, przez algorytmy wbudowane i funkcje studenta.

Student, w procesie doskonalenia kodu własnej funkcji, ma możliwość porównania wyników działania tworzonego algorytmu z algorytmem wbudowanym w celu oceny poprawności swojej implementacji. Służy to poznawaniu, rozumieniu i zapamiętaniu przekazanych wiadomości. Równocześnie kształtowana jest umiejętność posługiwania się pakietem MATLAB oraz zasad programowania. W aplikacji TMT dołożono wiele starań, aby prezentacja poszczególnych algorytmów i wyników ich działania była wykonana w sposób jak najbardziej przekonujący. Ponadto stworzono zestaw narzędzi ułatwiających pisanie przez studentów własnych procedur.

Program TMT jest obecnie wykorzystywany do celów dydaktycznych w ramach przedmiotu techniki multimedialne na kierunku Informatyka Stosowana w Państwowej

Wyższej Szkole Zawodowej w Tarnowie. Aplikacja spotkała się z pozytywnymi reakcjami zarówno ze strony studentów, jak i wykładowców. Planuje się dalszą rozbudowę i udoskonalanie programu TMT.

Podziękowania

Autorzy artykułu chcieliby złożyć najserdeczniejsze podziękowania wszystkim Dyplomantom, których trud, inwencja twórcza oraz pomysły przyczyniły się do powstania i udoskonalenia programu TMT.

Literatura

- [1] Czuba W., Kara P.: *Wybrane metody cyfrowego przetwarzania obrazów z wykorzystaniem środowiska Matlab 6.5*. Tarnów, PWSZ 2003 (Praca dyplomowa)
- [2] Czyżewski A.: *Dźwięk cyfrowy*. Warszawa, Akademicka oficyna wydawnicza EXIT 2001
- [3] *Wielka Encyklopedia PWN*. Warszawa, Wydawnictwo Naukowe PWN 2005
- [4] Jop R.: *Projekt i implementacja interfejsu użytkownika w programie TMT*. Tarnów, PWSZ 2005 (Praca dyplomowa)
- [5] Jop T.: *Realizacja wybranych elementów kompresji MP3 w środowisku Matlab 6.5*. Tarnów, PWSZ 2005 (Praca dyplomowa)
- [6] Kościółek K.: *Wybrane metody kompresji obrazu i cyfrowego przetwarzania dźwięku w środowisku Matlab 6.5*. Tarnów, PWSZ 2004 (Praca dyplomowa)
- [7] Netopedia 2005, strona internetowa: <http://www.ws-webstyle.com/cms.php/en/netopedia/multimedia/multimedia>
- [8] Sayood K.: *Kompresja danych. Wprowadzenie*. Wydawnictwo Read Me, 2002
- [9] Semmix 2005, strona internetowa: <http://semmix.pl/>
- [10] Skarbek W.: *Algorytmy i standardy kompresji*. Warszawa, Akademicka oficyna wydawnicza PLJ 1998
- [11] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Kraków, Wydawnictwo Fundacji Postępu Telekomunikacji 1997