# System do sterowania ruchem kamery przemysłowej za pomocą komend głosowych

Dariusz Krala[1]

[1]Wydział Inżynierii Mechanicznej i Informatyki
Kierunek Informatyka, Rok V
{dariusz.krala}@gmail.com

**Streszczenie**

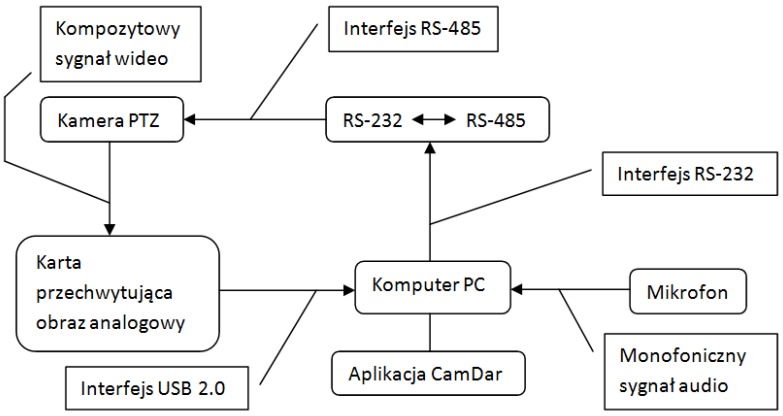
System do sterowania kamerą przemysłową za pomocą komend głosowych (system *CamDar*) to aplikacja napisana w oparciu o bibliotekę Qt. Umożliwia przechwytywanie obrazu analogowego z kamery i wyświetlanie go w oknie aplikacji. Obraz analogowy z kamery jest zamieniany na postać cyfrową przy użyciu tunera wideo podłączonego do komputera za pomocą interfejsu USB 2.0. Kamerą aplikacja steruje z komputera poprzez port szeregowy RS-232. Kamera posiada głowicę, która ma możliwość zmiany ostrości obrazu, powiększenia oraz jasności. Możliwe jest również sterowanie głowicą kamery w płaszczyźnie poziomej oraz pionowej. Aplikacja umożliwia sterowanie kamerą zarówno ręcznie przy użyciu myszki i panelu sterowania dostępnego w oknie programu, jak również poprzez wypowiadanie poleceń głosowych, które aplikacja potrafi rozpoznawać. Komendy wypowiadane są do mikrofonu podłączonego do karty dźwiękowej komputera. Program przechwytuje wypowiadaną komendę, następnie poddaje ją skomplikowanemu procesowi obróbki, którego celem jest dokonanie parametryzacji wypowiedzianej komendy i określenie jej znaczenia.

## 1 Wstęp

System *CamDar* to aplikacja okienkowa działająca pod kontrolą systemu Windows na komputerze klasy PC. Umożliwia sterowanie kamerą przemysłową oraz przechwytywanie z niej obrazu. Sterowanie kamerą możliwe jest przy pomocy przycisków znajdujących się w oknie aplikacji lub głosowo wypowiadając komendy do mikrofonu. Schemat blokowy systemu przedstawiono na rys.1.

## 2 Komunikacja z kamerą przemysłową

Użyta kamera wykorzystuje do komunikacji z urządzeniem sterującym protokół *PELCO*. Kamera obsługuje dwie wersje tego protokołu: 'P' [6] oraz 'D' [5]. W systemie zaimplementowano obsługę każdej z wymienionych wyżej wersji. Kamera automatycznie rozpoznaje wersję użytego protokołu bez potrzeby dodatkowego konfigurowania jej. Program komunikuje się z kamerą poprzez port szeregowy RS-232 z zastosowaniem konwertera RS-232C $\rightleftharpoons$ RS-485.




Rys. 1: Elementy wchodzące w skład systemu *CamDar*.

# 3 Przechwytywanie obrazu

Zastosowana w systemie *CamDar* kamera przemysłowa dostarcza obraz w postaci analogowej. W systemie *CamDar* odczyt obrazu z kamery PTZ zrealizowano poprzez zastosowanie karty przechwytującej obraz w postaci analogowej. Obraz transmitowany jest z kamery do karty wideo poprzez kabel kompozytowy. Karta przeprowadza digitalizację obrazu i wysyła go do komputera poprzez interfejs USB 2.0.

Od strony programowej do przechwytywania obrazu z konwertera zastosowano bibliotekę *OpenCV*[3]. Jest to biblioteka funkcji wykorzystywanych podczas obróbki obrazu, oparta o otwarty kod. Jest wieloplatformowa, można z niej korzystać w Mac OS X, Windows jak i Linux. W systemie skorzystano z biblioteki w wersji 2.0.

# 4 Przechwytywanie dźwięku

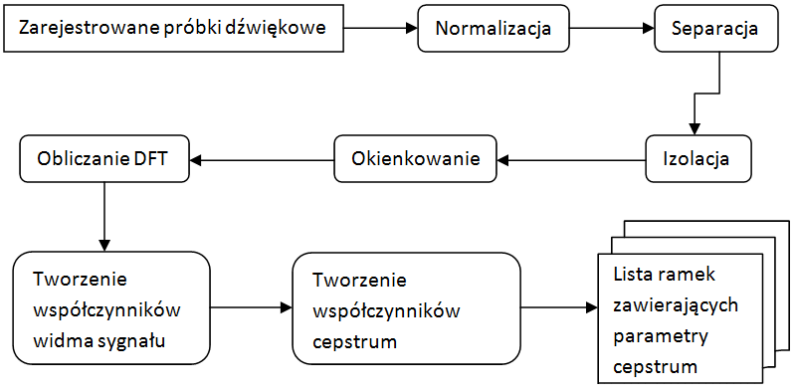
W systemie *CamDar* dźwięk pobierany jest z monofonicznego mikrofonu za pośrednictwem karty dźwiękowej. Dane są przechwytywane z domyślnego urządzenia rejestrującego w systemie.

Odczyt danych audio z urządzenia rejestrującego w aplikacji zrealizowano za pomocą funkcji API systemu Windows. Dane dźwiękowe trafiają do programu w formacie MONO PCM, a więc w postaci 1-kanałowego nieskompresowanego ciągu próbek dźwiękowych.

# 5 Proces obróbki dźwięku

Celem procesu obróbki dźwięku jest przetworzenie danych wejściowych, jakimi są kolejne próbki dźwięku zapisane z określoną rozdzielczością zależną od urządzenia rejestrującego i pobierane w równomiernych odstępach czasu, na ciąg parametrów możliwy do zinterpretowania i porównania z innymi parametrami, pochodzącymi z innych nagrań.

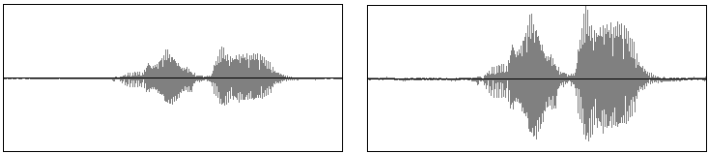
Po zakończeniu tego procesu nagranie dźwiękowe podzielone jest na stacjonarne ramki równej długości, a każda ramka sygnału dźwiękowego opisywana jest przez 20 współczynników MFCC (ang. *Mel Frequency Cepstral Coefficients*).




Rys. 2: Schemat blokowy ilustrujący kolejne kroki obróbki próbek dźwiękowych.

## Normalizacja

Normalizacja jest to proces polegający na sprowadzeniu parametrów dźwięku do odpowiednich stałych wartości. Parametrem takim może być np. maksymalna amplituda sygnału dźwiękowego.

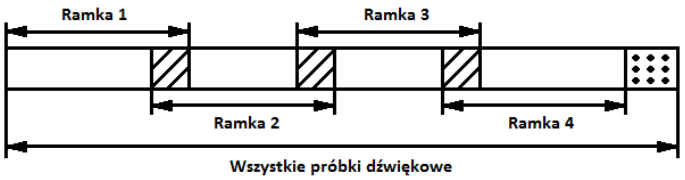


(a) (b)

Rys. 3: Wykres amplitudowo czasowy komendy *lewo*: (a) przed normalizacją; (b) po normalizacji.

## Separacja

Proces separacji polega na podzieleniu całego nagrania dźwiękowego na równe części zwane stacjonarnymi ramkami.



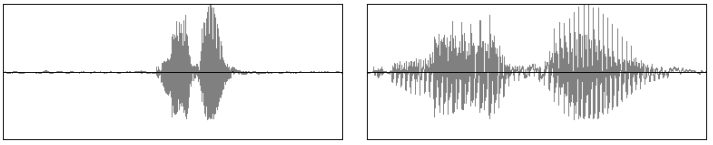

Rys. 4: Schemat podziału nagrania na stacjonarne ramki.

Rysunek 4 przedstawia zbiór wszystkich próbek dźwiękowych jakie zostały zarejestrowane w określonym czasie i znormalizowane. Po procesie separacji dane dźwiękowe zostały podzielone na równe ramki o określonym czasie trwania. Ramki te zachodzą

na siebie przez okres czasu określony przez parametr przesunięcia. Na rysunku jest to zaznaczone obszarem zakreskowanym. Jeżeli obszar danych nie da się podzielić równo pomiędzy wszystkie ramki, a na końcu zostaje pewna ilość próbek dźwięku o liczbie nie wystarczającej do utworzenia kolejnej ramki, to te próbki są porzucane. Na rysunku próbki porzucone są zaznaczone obszarem zakropkowanym.

## Izolacja

Celem procesu izolacji jest wydzielenie sygnału użytecznego (w tym przypadku jest to wypowiadana komenda) z pozostałego sygnału, który nie niesie żadnej użytecznej informacji.



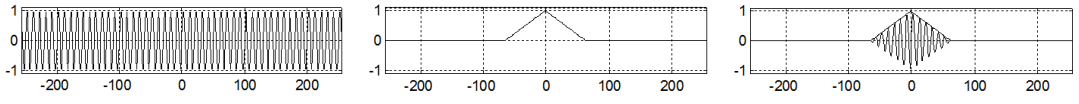
(a) (b)

Rys. 5: Wykres amplitudowo czasowy komendy *lewo*: (a) przed izolacją wraz z szumem otoczenia na początku i na końcu nagrania; (b) po izolacji.

## Okienkowanie

Ponieważ sygnał dźwiękowy jest na tym etapie podzielony na ramki powstaje niebezpieczeństwo zakłócenia właściwych parametrów sygnału. Każda ramka powoduje powstawanie nieciągłości przetwarzanego sygnału, prowadząc do powstawania niepotrzebnych wysokich częstotliwości w widmie sygnału. W celu wygładzenia nieciągłości i usunięcia z widma fałszywych prążków zastosowano dla wydzielonych ramek zawężające okna tłumiące skrajne próbki. Dodatkową zaletą stosowania procesu okienkowania jest redukcja przecieku DFT.

Proces okienkowania polega na przemnożeniu wartości próbek dźwiękowych zawartych w pojedynczej ramce przez wartości okna czasowego.




(a) (b) (c)

Rys. 6: Wykres: (a) Spróbkowanego sygnału $X(t)$; (b) okna trójkątnego; (c) sygnału po nałożeniu trójkątnego okna czasowego [7].

## Obliczanie DFT

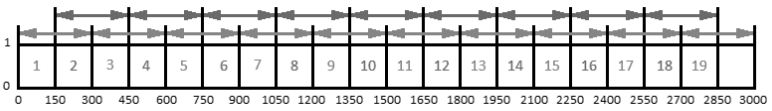
Celem tego procesu jest obliczenie widma amplitudowo częstotliwościowego sygnału zawartego w pojedynczej ramce. W tym celu zastosowano szybką transformatę Fouriera

(ang. *Fast Fourier Transform*, FFT). FFT to szybka odmiana algorytmu obliczania DFT czyli dyskretnego przekształcenia Fouriera (ang. *Discrete Fourier Transform* - DFT).

W aplikacji, aby zrealizować obliczanie DFT skorzystano z biblioteki FFTW [4]. Biblioteka FFTW została napisana w języku C i przeznaczona jest do obliczania DFT w tablicach o jednym lub większej liczbie wymiarów.

## Tworzenie współczynników widma mocy

Bazując na skali *mel* tworzony jest bank filtrów, dokonujący nieliniowej analizy częstotliwościowej sygnału, analogicznej do realizowanej przez ludzkie ucho. Filtry są równomiernie rozłożone w częstotliwościowej skali mel i mają prostokątne charakterystyki. W pracy zastosowano bank nakładających się filtrów o szerokości pasma dwa razy większego od przesunięcia względem siebie filtrów w skali *mel*.




Rys. 7: Charakterystyki banku 19 filtrów o szerokości 300*meli* przesuniętych względem siebie o 150*meli*.

Bank filtrów dla każdej ramki tworzy się na podstawie widma amplitudowego (lub mocy) obliczonego za pomocą DFT. Następnie wyznacza się oddzielnie dla każdego filtru z banku sumy współczynników widma amplitudowego (lub mocy) ważonych odpowiadającymi im wartościami charakterystyk amplitudowych filtru prostokątnego. Otrzymane sumy ważone nazywane są *parametrami banku filtrów*.

$$S(k) = \sum_{n=0}^{N-1} P(n)A(k,n), \text{ dla } k = 0,1,2,\ldots,K-1 \tag{1}$$

gdzie: $S(k)$ – współczynniki widma mocy; $P(n)$ – wartości DFT; $A(k,n)$ – amplituda filtra; $N$ – całkowita liczba próbek w ramce DFT; $K$ - liczba filtrów w banku [1].

## Tworzenie współczynników cepstrum

Parametry banku filtrów są w dużym stopniu powiązane ze sobą powodując pogorszenie skuteczności rozpoznawania, nawet jeśli się założy niezależność parametrów w wektorze obserwacji. Poprawienie jakości rozpoznawania można uzyskać poprzez zastosowanie przekształcenia cepstralnego parametrów banku filtrów, polegającego na wyznaczeniu współczynników cepstralnych w skali *mel* MFCC jako dyskretnych przekształceń kosinusowych logarytmów parametrów banku filtrów według zależności [1]:

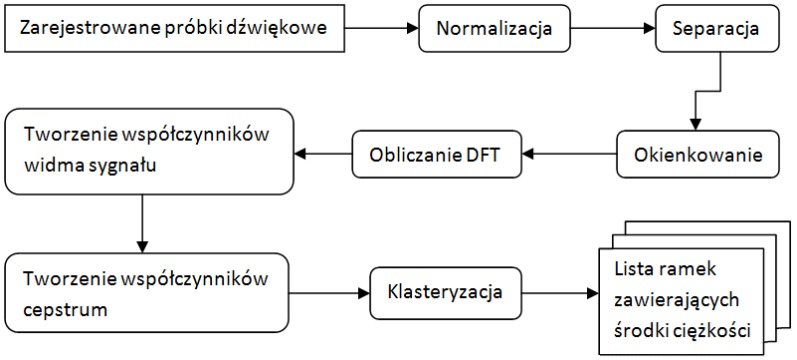
$$MFCC(n) = \sum_{k=0}^{K-1} \log(S(k)) \cos\left[n(k-0.5)\frac{\pi}{K+1}\right], \text{ dla } n = 0,1,2,\ldots,N \tag{2}$$

gdzie: $S(k)$ – współczynniki widma mocy; $K$ – liczba filtrów w banku filtrów, $N$ – liczba wymaganych współczynników cepstrum [1].

W pracy przyjęto liczbę wymaganych współczynników cepstrum $N$ równą 20. Dodatkową zaletą współczynników MFCC jest uniezależnienie sygnału mowy od wpływu kanału transmisji.

# 6 Proces tworzenia książki kodowej

W procesie analizy sygnału audio w pierwszym kroku należy przygotować książkę kodową, odzwierciedlającą przestrzeń akustyczną danego użytkownika. Proces tworzenia książki kodowej wykonuje się za każdym razem, gdy do systemu *CamDar* dodawany jest nowy użytkownik. Książka kodowa jest niezbędnym mechanizmem przy ekstrakcji wymaganych charakterystyk sygnału audio i tworzeniu wektorów obserwacji.




Rys. 8: Schemat blokowy procesu tworzenia książki kodowej.

W procesie tworzenia książki kodowej, analizie cepstralnej należy poddać nagranie zawierające zbiór wypowiedzi danego użytkownika, dobrany co do ilości i rodzaju słów w ten sposób, aby w całości pokryć przestrzeń akustyczną dla tego użytkownika. Zbiór taki powinien zawierać słowa obejmujące wszystkie fonemy języka polskiego wchodzące w skład przyjętego zagadnienia systemu. W przypadku sterowania systemem odpowiednimi komendami, można utworzyć zbiór wypowiedzi złożonych z wszystkich komend, jednak jeśli jest to tylko wykonalne, można ograniczyć ten zbiór pomijając komendy, których poszczególne fonemy występują już w innych, wziętych do zbioru komendach [1].

Podczas tworzenia książki kodowej podawany jest zestaw słów, które proces traktuje jako jedną wypowiedź. Nie stosuje się tutaj wydzielania właściwych słów, poprzez usunięcie ciszy, gdyż również dla ciszy przydzielany jest pewien symbol książki kodowej. Następnie podany zestaw słów poddawany jest procesowi obróbki dźwięku.

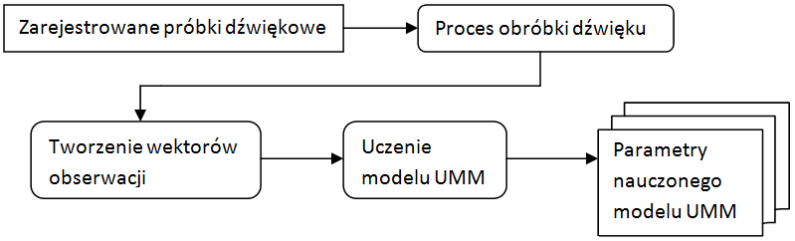
W systemie *CamDar* wszystkie poszczególne wypowiadane słowa są kodowane przy pomocy 37 symboli (20-wymiarowych wektorów) kodowych, odpowiadających liczbie fonemów języka polskiego. Wektory kodowe z książki reprezentują te wektory, które otrzymuje się z danych źródłowych. Każdemu wektorowi kodowemu przypisany jest liczbowy indeks.

Zbiór współczynników cepstrum wyznaczonych dla wszystkich ramek sygnału wejściowego tworzącego książkę kodową podlega podziałowi na przyjętą liczbę obszarów. Podział ten w systemie *CamDar* dokonuje się wykorzystując algorytm *k-means*. Każde-

mu wyróżnionemu obszarowi przypisana zostaje wartość symbolu z ograniczonego przyjętą liczbą obszarów zbioru symboli. W ten sposób wszystkie zakodowane przy pomocy współczynników cepstrum ramki zostają sklasyfikowane do jednego z wydzielonych obszarów zwanych klastrami. Klaster jest obszarem zawierającym elementy o podobnych cechach fonetycznych. Dla każdego klastra wyznaczany jest środek ciężkości (ang. *Centroid*) będący elementem leżącym "pośrodku" w grupie elementów tworzących klaster.

# 7 Proces uczenia

Proces uczenia polega na zakodowaniu każdej ramki sygnału zawierającego komendę uczącą opisywaną przez 20 współczynników MFCC przy pomocy symboli książki kodowej. Symbole książki kodowej są to numery kolejnych środków ciężkości zawartych w książce. Takie kodowanie to tzw. kwantyzacja wektorowa, gdzie zbiór wektorów współczynników cepstrum zostaje zamieniony na ciąg symboli, w pracy nazwany wektorem obserwacji. Tworzenie wektora obserwacji dla każdej ramki polega na obliczeniu odle-



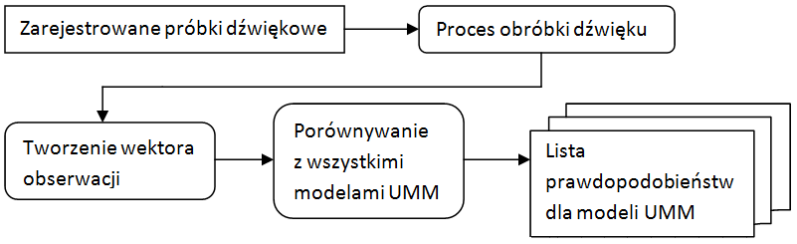

Rys. 9: Schemat blokowy procesu uczenia.

głości wektora współczynników cepstrum, od każdego z obszarów wydzielonych w książce kodowej. Symbol obszaru, dla którego ta odległość jest najmniejsza zostaje przypisany badanej ramce. W pracy do rozpoznawania mowy wykorzystano modele UMM (Ukryte Modele Markowa), w których podstawą rozpoznawania są jednostki fonetyczne, dla których tworzone są oddzielne modele akustyczne. Do tworzenia i zarządzania modelami UMM wykorzystano bibliotekę *Voice_HMM* [2]. W systemie rozpoznawania słów zastosowano dla każdego słowa oddzielny model UMM. Wszystkie modele w systemie są tych samych rozmiarów. Liczba modeli odpowiada liczbie słów zawartych w słowniku systemu. W systemie *CamDar* każdy użytkownik posiada własny zbiór modeli UMM.

# 8 Proces rozpoznawania polecenia

W procesie rozpoznawania cały przebieg budowy wektorów obserwacji komend głosowych jest taki sam, jak w procesie uczenia. W tym przypadku jednak na wejście podawane są wektory obserwacji tylko jednej wypowiedzi, bez jej powtórzeń. Rozpoznawane słowo reprezentowane przez wektor obserwacji porównywane jest z wszystkimi modelami UMM w systemie przyporządkowanych danemu użytkownikowi.

Największe prawdopodobieństwo określa model, który był uczony na danych najbardziej zbliżonych do rozpoznawanego słowa. Jeśli na wejście poda się słowo, dla którego




Rys. 10: Schemat blokowy procesu rozpoznawania komendy głosowej.

system nie został nauczony, wówczas wynikiem rozpoznania będzie słowo o podobnym brzmieniu, na którym system był uczony. Jest to jednak błędne rozpoznanie. W tym celu wprowadzono wartość progową określającą, kiedy prawdopodobieństwo wygenerowania sekwencji obserwacji można uznać za właściwe. Próg prawdopodobieństwa określa wartość jaką musi przekroczyć wartość bezwzględna różnicy prawdopodobieństw obliczana między dwoma najbardziej prawdopodobnymi modelami dla rozpoznawanej komendy. Jeśli warunek progu zostanie spełniony to na podstawie zapisanego w bazie danych znaczenia zwycięskiego modelu UMM podejmowanie jest określone działanie systemu.

# 9 Podsumowanie

System *CamDar* jest w pełni działający oraz skuteczny. Polecenia są w większości przypadków rozpoznawane poprawnie przy zastosowaniu różnych ustawień systemu wpływających na jakość rejestrowanego dźwięku oraz na proces jego obróbki. System poprawnie współpracuje z zastosowaną kamerą oraz z urządzeniem digitalizującym obraz z niej przechwytywany. Cały system można z pewnością uznać za sprawny i nadający się do użytkowania.

# Literatura

[1] dr inż. Mariusz Kubanek. *Metoda rozpoznawania audio-wideo mowy polskiej w oparciu o ukryte modele Markowa.* Częstochowa, 2005.

[2] Adam Mikuta. *Analiza metod stochastycznego modelowania ciągów czasowych w oparciu o komponent VOICE_HMM.* Częstochowa, 2008.

[3] http://opencv.willowgarage.com/wiki

[4] http://www.fftw.org/index.html

[5] http://www.commfront.com/RS232_Examples/CCTV/-Pelco_D_Pelco_P_Examples_Tutorial.HTM#3

[6] http://www.commfront.com/RS232_Examples/CCTV/-Pelco_D_Pelco_P_Examples_Tutorial2.HTM#6

[7] dict.comm.pl/wst_g/Okna%20czasowe.doc