

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka
SPECJALNOŚĆ: Technologie informacyjne w systemach automatyki (ART)

PRACA DYPLOMOWA
INŻYNIERSKA

Detekcja aktywności mówcy w systemach
automatycznego rozpoznawania mowy

Voice activity detection in automatic speech
recognition systems

AUTOR:
Paulina Szczerbak

PROWADZĄCY PRACĘ:
Prof. dr hab. inż. Ryszard Makowski

OCENA PRACY:

Spis treści

1 Wstęp	3
2 Generowanie sygnału mowy	5
2.1 Mowa w życiu człowieka	5
2.2 Biologiczny proces generowania mowy	5
2.3 Sygnał w dziedzinie czasu oraz w dziedzinie częstotliwości	7
3 Wybrane metody detekcji aktywności mówcy	9
3.1 Czym jest detekcja aktywności mówcy oraz gdzie się ją wykorzystuje	9
3.2 Metoda bazująca na energii sygnału z adaptacyjnym współczynnikiem skalującym	10
3.2.1 Początkowy próg detekcji	10
3.2.2 Próg detekcji zmieniający się dynamicznie w czasie	10
3.2.3 Wyznaczanie progu	10
3.2.4 Rozszerzenie algorytmu	11
3.3 Metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości	12
3.3.1 Obwiednie sygnału dla każdej częstotliwości	12
3.3.2 Ważone składowe obwiedni sygnału mowy	13
3.3.3 Logika podejmowania decyzji	15
3.4 Zmieniony algorytm Single Frequency Filtering (SFF2)	17
4 Implementacja	19
4.1 Metoda oparta o energię sygnału	19
4.2 Metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości	20
4.3 Zmieniona metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości	23
5 Wyniki badań	25
5.1 Sposób oceny	25
5.2 Wyniki dla pojedynczych słów	26
5.3 Wyniki dla ciągów słów	30
5.4 Powtórzenie badań przy większym poziomie zaszumienia	35
6 Podsumowanie	39
Bibliografia	39

Rozdział 1

Wstęp

Celem niniejszej pracy jest zaprezentowanie wybranych metod detekcji aktywności mówcy (VAD) w systemach automatycznego rozpoznawania mowy w oparciu o napisany program w języku C++. Kolejnym etapem jest porównanie zaimplementowanych metod pod względem dokładności detekcji w separowanych wyrazach oraz w dłuższych ciągach słów.

Rozdział 2. opisuje w uproszczony sposób proces wytwarzania mowy przez człowieka. Prezentuje, w jaki sposób działa aparat mowy oraz z jakich narządów się składa. Na koniec pokazany jest matematyczny model jaki można stworzyć wzorując się na naturalnym systemie generowania mowy.

Rozdział 3. zawiera wyjaśnienie na temat detekcji aktywności mówcy - czym jest oraz gdzie jest wykorzystywana. W tym rozdziale opisane są również wybrane algorytmy, które zostały zestawione w dalszej części pracy. Przedstawiona jest zasada ich działania oraz pokrótko wyjaśniona kwestia implementacyjna każdego z nich.

Rozdział 4. objaśnia niektóre aspekty implementacyjne programu.

Rozdział 5. przedstawia sposób oceny wyników. Zawiera wyniki detekcji dla pojedynczych słów oraz całych ciągów. Na końcu tego rozdziału znajduje się również ostateczne porównanie w działaniu wybranych algorytmów i ich ocena.

Rozdział 2

Generowanie sygnału mowy

2.1 Mowa w życiu człowieka

Mowa w życiu większości ludzi stanowi podstawę komunikacji interpersonalnej. Jest sygnałem akustycznym, czyli rozważany jest zakres częstotliwości słyszanych przez człowieka, to jest od 20Hz do 16kHz. Zatem mowa to nic innego jak system artykułowanych dźwięków, które układają się zgodnie z konwencją wybranego języka. Pełni ona funkcję nie tylko komunikacyjną (przekazywanie informacji drugiej osobie o tym, co doświadczyliśmy, czy czego się dowiedzieliśmy), ale również ekspresyjną (można w niej zawrzeć informacje o emocjach nadawcy) oraz regulacyjną (wydawanie i przyjmowanie dyspozycji).

2.2 Biologiczny proces generowania mowy

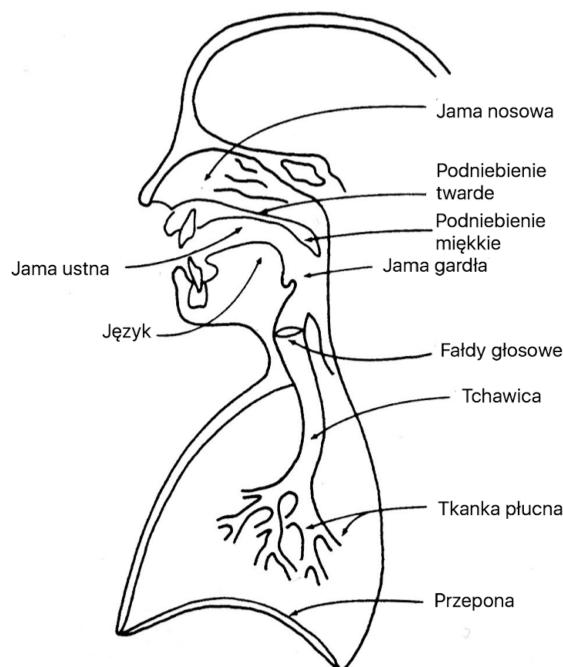
Wszelkie metody przetwarzania sygnału mowy muszą bazować na strukturze sygnału, a ta jest niewątpliwie uzależniona od sposobu, w jaki jest on wytwarzany. Niegdyś generowanie sygnałów mowy było domeną jedynie organizmu człowieka, czyli systemu naturalnego. W celu stworzenia systemu, który w jakiś sposób operuje na sygnałach mowy, czyli np. syntezatora mowy, systemu generującego sygnały mowopodobne, systemu automatycznego rozpoznawania mowy czy detekcji aktywności mówcy, należy mieć przynajmniej podstawową wiedzę na temat systemu naturalnego - tego, w jaki sposób działa aparat mowy człowieka.

Wytwarzanie mowy przez człowieka jest procesem niezwykle skomplikowanym, który ma swój początek w mózgu, gdzie następuje konstrukcja wypowiedzi. Później następuje sformułowanie fonetyki i artykulacja poprzez aparat mowy. Ponadto, w procesie generowania mowy można wyróżnić cztery pomniejsze etapy:

- proces psychologiczny - wymyślenie i skonstruowanie wypowiedzi,
- proces neurologiczny - pobudzenie przez układ nerwowy mięśni, które biorą udział w wytwarzaniu mowy,
- proces fizjologiczny - proces kształtowania dźwięków mowy ludzkiej,
- proces aerodynamiczny - drgania i przepływ powietrza przez aparat mowy.

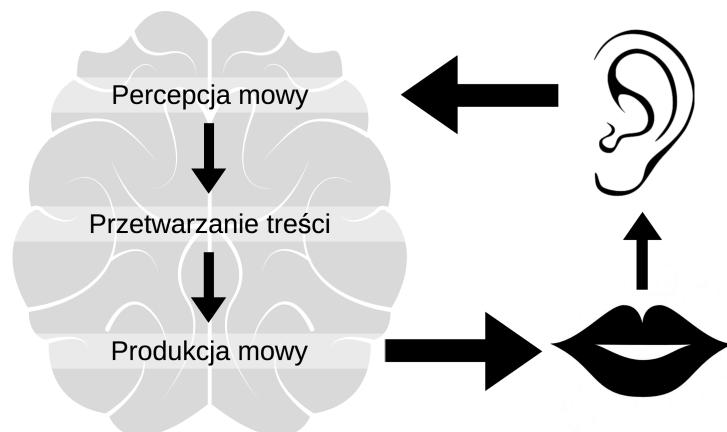
Pierwszym narządem wchodząącym w skład traktu głosowego człowieka są płuca - dostarczają one powietrze do procesu artykulacji, są źródłem zmian ciśnienia akustycznego. Organ mowy człowieka jest napędzany przez wydychane powietrze. Powietrze to, jest prowadzone przez oskrzela i tchawicę do krtani, a drgające w niej struny głosowe modyfikują ciśnienie i wytwarzają dźwięczne fragmenty mowy. Następnie, dzięki wnękom rezonansowym, tworzonym przez język, podniebienie, zęby oraz wargi, dźwięk ten jest modulowany.

Niezwykle ważną rolę przy formowaniu tych wnęk, odgrywają ruchy żuchwy i policzków. Podczas generowania głosek nosowych zamknięta jama ustna spełnia rolę bocznika akustycznego, a dzięki odpowiedniemu ustawieniu językczka podniebienia miękkiego, fala dźwiękowa jest emitowana przez jamę nosową i nozdrza. Struktura traktu głosowego jest przedstawiona schematycznie na rysunku 2.1.



Rysunek 2.1 Aparat mowy człowieka

Ponadto, sterowanie całym systemem generowania mowy jest bardzo złożone i w dużej mierze opiera się na licznych sprzężeniach zwrotnych. Główną rolę odgrywa tutaj sprzężenie zwrotne, które poddaje jakość wydawanych dźwięków bezpośrednio ocenie poprzez analizator słuchowy. Dzięki temu proces artykulacji jest odpowiednio sterowany. Istotę tego sprzężenia zwrotnego potwierdzają trudności z mową wśród ludzi głuchych oraz ludzi słyszących, którzy tymczasowo przebywają w trudnych warunkach środowiskowych, które uniemożliwiają słyszenie własnego głosu.



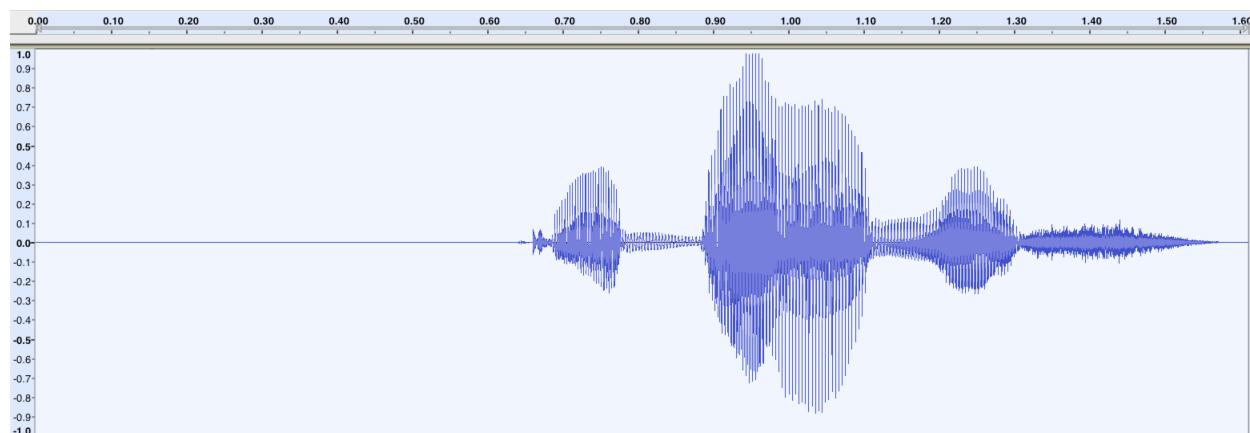
Rysunek 2.2 Sprzężenie zwrotne w wytwarzaniu mowy

2.3 Sygnał w dziedzinie czasu oraz w dziedzinie częstotliwości

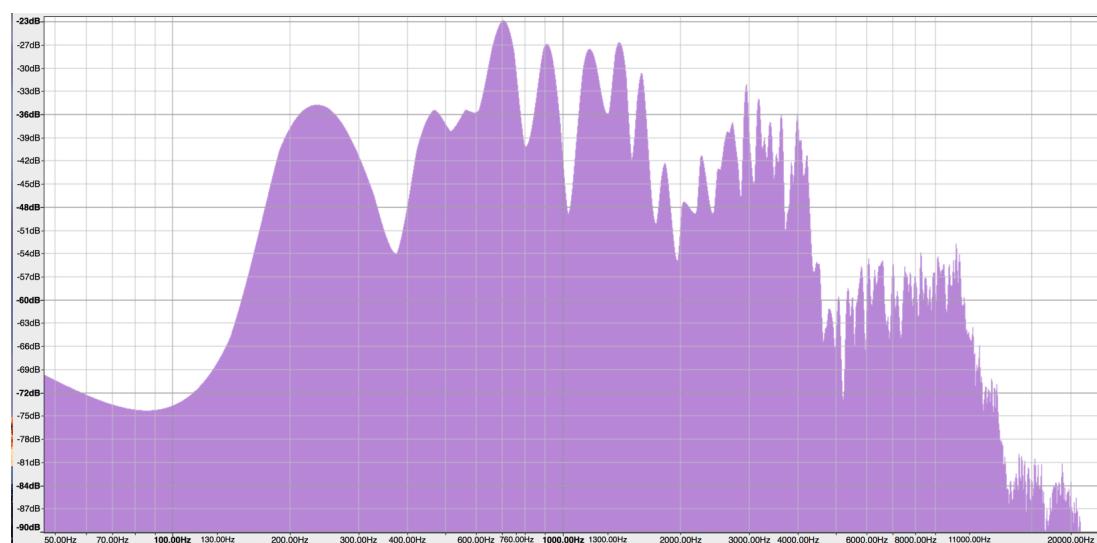
Powstawanie głosu jest bardzo złożonym zjawiskiem akustyczno - mechanicznym, a jego bazowym składnikiem jest dźwięk. W kategoriach fizycznych dźwięk tworzą drgania mechaniczne, a w organizmie człowieka takim generatorem drgań jest krtań. Charakter tego dźwięku, zwany też tonem krtaniowym (lub tonem podstawowym), zależy od właściwości fałdów głosowych - ich długości, napięcia, elastyczności i masy oraz od charakteru przepływu powietrza. Ton krtaniowy ma określoną wysokość i natężenie, ale sam w sobie jest słaby i bezbarwny. Nabiera odpowiedniej siły i barwy dopiero poprzez przejście przez wyższe partie traktu głosowego.

Mowę możemy podzielić na dźwięczną i bezdźwięczną. Mowa dźwięczna wyróżnia się okresowością - jest bowiem powieleniem tonu krtaniowego. Bezdźwięczna natomiast - ma charakter szumowy, w trakcie jej trwania struny głosowe pozostają nieruchome. Dzięki analizie sygnału w dziedzinie czasu - Rysunek 2.3 - można w prosty sposób wyróżnić dźwięczne fragmenty mowy dzięki regularnie powtarzającym się maksymach.

Najważniejszym jednak narzędziem do analizowania mowy, jest jego widmo częstotliwościowe - czyli przedstawienie sygnału w dziedzinie częstotliwości, które można otrzymać przy pomocy transformaty Fouriera. Przypada ono na zakres ok 100-8000Hz. Przykład został zaprezentowany na Rysunku 2.4.



Rysunek 2.3 Przebieg sygnału w dziedzinie czasu dla słowa *kabanos*



Rysunek 2.4 Widmo sygnału dla słowa *kabanos*

Rozdział 3

Wybrane metody detekcji aktywności mówcy

3.1 Czym jest detekcja aktywności mówcy oraz gdzie się ją wykorzystuje

Detekcja aktywności mówcy (Voice Activity Detection - VAD) jest powszechnie stosowana w systemach automatycznego rozpoznawania mowy. Podczas rejestrowania wypowiedzi do późniejszego przetwarzania jej przez system ARM, zostaje zarejestrowana cała wypowiedź mówcy, włącznie z częścią, która nie zawiera mowy. Jeżeli we fragmencie jest zawarty sygnał mowy, mówimy, że mówca jest aktywny. Aktywnością mówcy nazywa się emitowny przez niego dźwięk. Zawartość semantyczna wypowiedzi jest zawarta w głównej mierze we fragmentach, kiedy mówca jest aktywny. Analizowanie całego zarejestrowanego sygnału mowy, bez wykorzystania systemu VAD, jest oczywiście możliwe, aczkolwiek niepotrzebnie zwiększa czas obliczeń oraz istnieje prawdopodobieństwo, że fragment, gdy mówca nie jest aktywny, zostanie błędnie zaklasyfikowany jako jakiś konkretny fonem - zatem w dużej mierze może popsuć jakość rozpoznania. Detekcja aktywności mówcy w ogólnym przypadku zakłada, że sygnał może występować w dwóch stanach: tylko szum (brak sygnału mowy), szum + sygnał mowy. Korzystając z zagadnienia hipotez ze statystyki, możemy pierwszy stan oznać jako hipotezę H_0 , a drugi jako H_1 , dzięki czemu możemy przedstawić to w następujący sposób:

$$\begin{aligned} H_0 : f(n) &= x(n) \\ H_1 : f(n) &= v(n) + x(n) \end{aligned} \tag{3.1}$$

Przy takim rozumowaniu konieczne jest określenie statystyki $S(n)$ sygnału, dzięki czemu możliwe będzie dokonywanie detekcji, a w dalszej kolejności zastosowanie kryterium decyzyjnego. Kryterium decyzyjne zwykle polega na porównaniu wartości $S(n)$ z progiem detekcji, który w mniej skomplikowanych algorytmach przyjmuje stałą wartość. Natomiast w tych bardziej złożonych, może występować np. jako funkcja czasu. Wartość stałej wartości progu jest ustalana w wyniku teoretycznych rozważań lub empirycznie. Zatem detekcja $\gamma(n)$, w ogólnej postaci, będzie prezentować się następująco:

$$\begin{aligned} S(n) \geq \gamma(n) &\rightarrow H_1 \\ S(n) < \gamma(n) &\rightarrow H_0 \end{aligned} \tag{3.2}$$

3.2 Metoda bazująca na energii sygnału z adaptacyjnym współczynnikiem skalującym

Najbardziej powszechną metodą do obliczenia energii dla całego pasma w sygnale mowy jest:

$$E_j = \frac{1}{N} \sum_{i=(j-1)N+1}^{jN} x^2(i)$$

gdzie: E_j - energia j-tej ramki

3.2.1 Początkowy próg detekcji

Początkowa wartość progu jest ważna dla jego dalszego rozwoju - ponieważ będzie się zmieniała zgodnie ze średnim poziomem szumu w sygnale. Przyjęto, że początkowe 100 ms nagrania nie zawiera mowy. Jest to podyktowane tym, że mówca potrzebuje czasu na rekację, nabranie powietrza, aktywację strun głosowych. Te 100 ms są uznawane za przebieg pozabawiony sygnału mowy i ich średnia wartość obliczana jest zgodnie z powyższym wzorem.

3.2.2 Próg detekcji zmieniający się dynamicznie w czasie

Główną ideą tego algorytmu jest możliwość obliczenia progu detekcji bez potrzeby korzystania z obszarów niezawierających mowy. Wykorzystywana jest minimalna oraz maksymalna energia sygnału mowy.

Innym popularnym sposobem na obliczenie energii sygnału mowy jest pierwiastek ze średniokwadratowej wartości energii (root mean square energy - RMSE), dany jako:

$$E_j = \sqrt{\frac{1}{N} \sum_{i=(j-1)N+1}^{jN} x^2(i)}$$

'Dynamiczny' VAD jest oparty o obserwację, że estymata mocy sygnału mowy pokazuje wyraźne szczyty i doliny. Podczas gdy szczyty odpowiadają aktywności mowy, doliny mogą zostać wykorzystane do uzyskania estymaty mocy szumu. Ponadto, RMSE jest bardziej odpowiedni.

3.2.3 Wyznaczanie progu

Estymacja progu bazuje na poziomach energii E_{min} oraz E_{max} otrzymane z ciągu nadchodzących ramek. Te wartości są trzymane w pamięci i próg θ jest obliczany jako:

$$\theta = k_1 E_{max} + k_2 E_{min}$$

gdzie k_1 i k_2 to współczynniki wykorzystane do interpolacji wartości progu dla optymalnych wyników.

Jeżeli energia bieżącej ramki jest mniejsza niż wartość progu, ramka zostaje oznaczana jako niezawierająca mowy.

Jako, że mogą się pojawić pewne anomalie spowodowane zbyt niską energią, wprowadzono odpowiednią prewencję. Parametr E_{min} jest nieznacznie zwiększany dla każdej ramki, zdefiniowane jako:

$$E_{min}(j) = E_{min}(j - 1)\Delta(j)$$

Parametr Δ dla każdej ramki jest zdefiniowany jako:

$$\Delta(j) = \Delta(j - 1) \cdot 1.0001$$

3.2.4 Rozszerzenie algorytmu

Możliwe jest przedstawienie równania na wyznaczenie dynamicznie zmieniającego się programu przy pomocy jednego parametru λ (np. $\lambda = k_2$).

$$\theta = (1 - \lambda)E_{max} + \lambda E_{min}$$

gdzie λ to parametr skalujący , który kontroluje proces estymacji.

Detektor mowy działa wiarygodnie, gdy λ należy do przedziału $[0.950, \dots, 0.999]$. Jednakże, wartości dla różnych typów sygnałów mogą nie być takie same i informacja a priori wciąż wymaga poprawnego ustalenia wartości λ . Równanie poniżej pokazuje jak sprawić, żeby współczynnik skalujący λ był niezależny i odporny na zmieniające się warunki środowiska.

$$\lambda = \frac{E_{max} - E_{min}}{E_{max}}$$

3.3 Metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości

Do dokonania detekcji, należy wcześniej policzyć obwiednię dla sygnału podzielnego na 185 pasm - od 30Hz do 4000Hz, co 20Hz. Wybrany przedział częstotliwości pokrywa się z użytecznym pasmem, który jest wykorzystywany przez mowę. Poniżej zostały przedstawione kolejne kroki potrzebne do policzenia 185 obwiedni i odpowiedniego przekształcenia ich w funkcję czasu, na której będzie można dokonać detekcji.

Sygnal mowy ma zależności zarówno w dziedzinie czasu, jak i w dziedzinie częstotliwości. Skutkuje to tym, że stosunek sygnał-szum (SNR) jest funkcją czasu, jak i częstotliwości. Dla idealnego szumu o danej całkowitej mocy, moc jest równo rozdzielona na wszystkie częstotliwości, podcza

3.3.1 Obwiednie sygnału dla każdej częstotliwości

Sygnal mowy w zdyskretyzowanej dziedzinie czasu $s(n)$ jest różniczkowany i jest rozumiany jako $x(n) = s(n) - s(n - 1)$. Częstotliwość próbkowania to f_s . Sygnal $x(n)$ jest przemnażany przez zespoloną sinusoidę o danej znormalizowanej częstotliwości $\bar{\omega}_k$. Wynik tej operacji w dziedzinie czasu jest dany jako:

$$x_k(n) = x(n)e^{j\bar{\omega}_k n},$$

gdzie $\bar{\omega}_k = \frac{2\pi f_k}{f_s}$

Kiedy pomnożymy $x(n)$ przez $e^{j\bar{\omega}_k n}$, wynikowe widmo $x_k(n)$ będzie przesuniętym widmem $x(n)$. Czyli:

$$X_k(\omega) = X(\omega - \bar{\omega}_k),$$

gdzie $X_k(\omega)$ i $X(\omega)$ to odpowiednio widma $x_k(n)$ i $x(n)$.

Sygnal $x_k(n)$ jest przepuszczany przez jednobiegunkowy filtr, którego transmitancja jest dana jako:

$$H(z) = \frac{1}{1 + rz^{-1}}$$

Jednobiegunkowy filtr ma biegun na osi liczb rzeczywistych w odległości r od początku układu współrzędnych. Lokalizacja pierwiastka jest w $z = -r$ na płaszczyźnie liczb zespolonych, co odpowiada połowie częstotliwości próbkowania, np. $f_s/2$. Wyjście filtra $y_k(n)$ jest dane jako:

$$y_k(n) = -ry_k(n - 1) + x_k(n)$$

Obwiednia sygnału $y_k(n)$ jest dana jako:

$$e_k(n) = \sqrt{y_{kr}^2(n) + y_{ki}^2(n)},$$

gdzie $y_{kr}(n)$ i $y_{ki}(n)$ są odpowiednio częścią rzeczywistą i urojoną $y_k(n)$.

Kiedy filtrowanie $x_k(n)$ będzie zrobione dla $\frac{f_s}{2}$, powyższa obwiednia $e_k(n)$ będzie odpowiadać obwiedni sygnału $x_k(n)$ przefiltrowanego w pożądanej częstotliwości

$$f_k = \frac{f_s}{2} - \bar{f}_k$$

Powyższa metoda estymowania obwiedni składowej dla częstotliwości f_k jest określana jako podejście filtracji pojedynczych częstotliwości (Single Frequency Filtering). Wybór filtra z biegunem w $z = -r$ do estymacji obwiedni przefiltrowanego sygnału wydaje się być bardziej odpowiedni, jako że obwiednie są obliczane w możliwie najwyższych częstotliwościach ($f_s/2$). Ponadto, wybór filtru w stałej częstotliwości dla jakiekolwiek pożądanej częstotliwości f_k zapobiega efektu przeskalowania w związku z różnymi wzmacnieniami filtrów w różnych częstotliwościach. Jeżeli biegun zostanie wybrany z obszaru na kole jednostkowym, np $z = r = -1$, może to skutkować niestabilnością wyjścia filtra. Stabilność filtra jest zapewniona dzięki przesunięciu bieguna nieco bardziej wewnątrz koła jednostkowego. Z tego powodu r zostało dobrane jako 0.99.

W tym badaniu obwiednia została obliczona dla każdych 20Hz w przedziale od 300Hz do 3000Hz jako funkcja w dziedzinie czasu. Wybrany został przedział częstotliwości 300-4000Hz, ponieważ pokrywa się z użytkowym pasmem wykorzystywanym przez mowę. Zatem mamy obwiednie dla 185 częstotliwości jako funkcja w dziedzinie czasu. Zasadniczo obwiednia może zostać obliczona dla każdej pożądanej częstotliwości.

3.3.2 Ważone składowe obwiedni sygnału mowy

Kiedy sygnał mowy ma bardzo dużą rozpiętość tonalną w dziedzinie częstotliwości, sygnał może mieć wysoką wartość mocy w niektórych częstotliwościach w każdej chwili czasowej. W tych częstotliwościach SNR będzie miał większą wartość, jako, że moc szumu będzie prawdopodobnie mniejsza w związku z większym rozkładem jednostajnym mocy. Nawet dla szumów z nierównomiernym rozkładem mocy, niższe korelacje próbek szumu skutkują w niższej rozpiętości tonalnej w rozpiętości mocy szumu przez częstotliwość, w porównaniu z sygnałem mowy. Zauważmy, że widmowa rozpiętość tonalna daje przejaw korelacji próbek w dziedzinie czasu.

Moc szumu tworzy funkcję cechy (podlogi) dla obwiedni dla każdej częstotliwości i poziom cechy zależy od rozkładu mocy szumu wobec częstotliwości. Podloga jest bardziej jednorodna wobec czasu, jeżeli szum jest niemalże stacjonarny. Nawet jeżeli szum jest niestacjonarny, jest względnie stacjonarny ponad większymi przerwami w czasie niż sygnał mowy. W takich przypadkach, poziom cechy może zostać obliczony ponad długimi przerwami w dziedzinie czasu dla każdej częstotliwości, jeżeli jest to potrzebne.

Żeby zrekompensować efekt szumu, wartość wagi dla każdej częstotliwości jest obliczana używając wartości funkcji cechy. Dla każdego wyrażenia, średnia (μ_k) z 20 % najmniejszych wartości, wartości obwiedni dla każdej częstotliwości f_k jest wykorzystywana do obliczenia znormalizowanej wagi wartości ω_k dla danej częstotliwości. Wybór akurat 20% wartości jest oparty o założenie, że jest przynajmniej 20% ciszy w każdym wyrażeniu mowy. Znormalizowana waga wartości w każdej częstotliwości jest dana jako:

$$\omega_k = \frac{\frac{1}{\mu_k}}{\sum_{t=1}^N \frac{1}{\mu_t}},$$

gdzie N to liczba kanałów.

Obwiednia $e_k(n)$ dla każdej częstotliwości f_k jest przemnażana przez wartość wagi w_k w celu zrekompensowania poziomu szumu w tej częstotliwości. Wynikowa obwiednia jest określana jako obwiednia z ważonymi komponentami. Zauważmy, że przez to ważenie, obwiednia dla każdej częstotliwości jest dzielona przez estymatę cechy szumu u_k .

Do wszystkich sygnałów została dodana mała ilość białego szumu, żeby mieć pewność, że wartość funkcji cechy nie jest zerem. Dla obliczeń w_k , wartości w dodanych obszarach ciszy nie będą rozważane.

W każdej chwili czasu średnia ($\mu(n)$) kwadratu ważonych obwiedni obliczonych wobec częstotliwości odpowiada w przybliżeniu energii sygnału w danej chwili (rys 2(c)) Oczekuje się, że $\mu(n)$ będzie wyższe dla mowy, niż dla szumu w obszarach, gdzie występuje sygnał mowy, ponieważ wartości szumu są o obniżonej wadze. W każdej chwili czasu, odchylenie standardowe ($\sigma(n)$) kwadratu ważonych obwiedni również będzie względnie wyższe dla mowy niż dla szumu w obszarach mowy - związane jest to ze strukturą formantu. Dlatego $\sigma(n) + \mu(n)$ jest na ogół wyższe w obszarach mowy i niższe w regionach pozabawionych mowy. Ponieważ oczekuje się, że rozpiętość szumu (po kompensacji) będzie niższa, zaobserwowano, że wartości $\sigma(n) - \mu(n)$ są zwykle niższe w obszarach pozabawionych mowy, w porównaniu do obszarów zawierających mowę (rys. 2(e)). Pomnożenie ($\sigma(n) + u(n)$) przez ($\sigma(n) + u(n)$) daje ($\sigma^2(n) - u^2(n)$), co podkreśla kontrast pomiędzy obszarami zawierającymi mowę i tymi, które mowy nie zawierają.

W związku z dużą rozpiętością tonalną wartości ($\sigma^2(n) - u^2(n)$), ciężko jest zaobserwować obszary mowy z małymi wartościami ($\sigma^2(n) - u^2(n)$). Aby podkreślić kontrast pomiędzy obszarami mowy i obszarami niezawierającymi mowy, rozpiętość tonalna jest redukowana poprzez obliczenie

$$\delta(n) = \sqrt[M]{|(\sigma^2(n) - \mu^2(n))|},$$

gdzie M zostało wybrane jako 64

*** tutaj rysunek Rys.3.1 z przykładami $\mu(n)$, $\sigma(n)$, $\sigma(n) - \mu(n)$, $\delta(n)$ ze znakiem, $\delta(n)$

Wartość M nie jest decydująca. Każda wartość M z przedziału 32-256 wydaje się być dobra, aby zapewnić dobry kontrast pomiędzy obszarami zawierającymi mowę, a tymi, które mowy nie zawierają na wykresie $\delta(n)$. W obliczeniach $\delta(n)$ brana jest po uwagę tylko wartość bezwzględna wartości chwilowej ($\sigma^2(n) - \mu^2(n)$). Jeżeli znak wyrażenia ($\sigma^2(n) - \mu^2(n)$) jest przypisany(?) do delta(n), wartości będą wałać się w okolicach zera w obszarach pozabawionych mowy dla większości typów szumów , ale krótki czas (20-40 msec) tymczasowych średnich wartości będzie mały i będzie się wałał, sprawiając, że cecha szumu będzie nierówna. To powoduje trudności w ustaleniu progu detekcji dla obszarów pozabawionych mowy. Wartości $\delta(n)$ będą miały wysoką średnią w obszarach pozabawionych mowy z małą średnią wariancją. Pomoże to w ustaleniu odpowiedniego progu do odizolowania obszarów pozabawionych mowy od tych, które mowę zawierają. Zakres $\delta(n)$ ze znakiem rys. 3.1(f) jest inny niż wartości $\delta(n)$ (rys. 3.1(g)). Mały tymczasowy obszar wartości $\delta(n)$ w obszarach niezawierających mowy i jego średnia wartość pomagają w dobraniu pasującego progu. Wartości $\delta(n)$ w obszarach niezawierających mowy są podykowane poziomem szumu. Zauważmy, że rozważając wartości $\delta(n)$ bez znaku, tracimy trochę zalet w rozróżnialności obszarów niezawierających mowy, które mają zarówno dodatnie, jak i ujemne wartości - natomiast obszary zawierające mowę mają w większości dodatnie wartości. Wartości $\delta(n)$ z $M = 64$ są wykorzystywane do dalszego przetwarzania do podejmowania decyzji. Warto zauważać zmiany w przeskalowaniu na rys 3.1(f) i 3.1(g), aby zrozumieć istotę używania wartości bezwzględnej, np $\delta(n)$ bez znaku.

3.3.3 Logika podejmowania decyzji

Logika podejmowanej decyzji opiera się o $\delta(n)$ dla każdego wyrażenia poprzez wyprowadzenie najpierw progu detekcji z przyjętego z założenia obszaru zawierającego szum, a później zastosować ten próg na tymczasowo wygładzonych wartościach $\delta(n)$. Rozmiar okna l_w wykorzystany do wygładzenia $\delta(n)$ jest zaadoptowany w oparciu o estymatę rozpiętości tonalnej (ρ) energii zaszumionego sygnału dla każdego wyrażenia, zakładając, że jest przynajmniej 20% obszarów zawierających ciszę w każdym wyrażeniu. Binarna decyzja odnośnie mowy i jej braku w każdej chwili czasowej, oznaczana odpowiednio jako 1 i 0, jest dalej wygładzana z wykorzystaniem okna adaptacyjnego, żeby dotrzeć do ostatecznej decyzji detekcji. Następujące 5 kroków opisuje implementację szczegółów w logice podejmowania decyzji:

1) Obliczenie progu (θ):

Obliczyć średnią (μ_θ) i wariancję (σ_θ) dla 20% najmniejszych wartości.

Próg $\theta = \mu_\theta + 3\sigma_\theta$ jest używany we wszystkich przypadkach. Wartość θ zależy od analizowanego wyrażenia. Zatem wartość progu, odpowiadająca wartości cechy z $\theta(n)$, jest adaptowana do konkretnego wyrażenia w zależności od charakterystyki sygnału i szumu w tym wyrażeniu.

2) Wyznaczenie okna wygładzającego l_w :

Energia E_m sygnału $x(n)$ jest obliczana dla ramki 300msec z przesunięciem 10msec, gdzie m to numer ramki. Rozpiętość tonalna (ρ) sygnału jest obliczana jako:

$$\rho = 10 \log_{10} \frac{\max_m(E_m)}{\min_m(E_m)}.$$

Parametr opisujący długość okna l_w do wygładzenia sygnału jest uzyskiwany z rozpiętości tonalnej (ρ) sygnału. Wartości ρ różnią się dla różnych szumów przy tym samym SNR, ponieważ charakterystyki szumów się różnią. Wskaźnik SNR dla mowy z dystansu zależy od warunków środowiskowych oraz od odległości, z jakiej mówca mówi do mikrofonu. Zaobserwowano, że wartości ρ dla mowy z odległością są rozciągnięte w porównaniu z wartościami ρ dla różnych szumów. Jest to głównie spowodowane efektem echa. Rozkład wartości ρ zależy również od odległości mówcy od mikrofonu. Wartość ρ dla każdego wyrażenia jest wykorzystywana do określenia wartości niektórych parametrów do dalszego przetwarzania $\delta(n)$ i do otrzymania decyzji o klasyfikacji. W przypadkach, gdzie $\delta(n)$ reprezentuje charakterystyka dyskryminacyjna przedstawiająca zarówno mowę, jak i jej brak, odpowiadające wartości ρ są wysokie, jak zaobserwowano w przypadku szumów volvo, lamparta i karabinu maszynowego. W takich przypadkach używane są małe wartości parametru l_w okna wygładzającego. Następujące wartości l_w zostały wybrane na drodze przeprowadzonych doświadczeń z sygnałem mowy zaszumionym przez różne typy szumów z różnymi poziomami SNR:

$$l_w = 400\text{msec}, \quad \text{dla } \rho < 30$$

$$l_w = 300\text{msec}, \quad \text{dla } 30 \leq \rho < 40$$

$$l_w = 200\text{msec}, \quad \text{dla } \rho > 40.$$

3) Logika podejmowania decyzji w każdej chwili czasowej:

Wartości $\delta(n)$ są uśredniane przez okno o rozmiarze l_w , aby otrzymać uśrednione wartości $\bar{\delta}(n)$ w każdej próbce o indeksie n. Decyzja jest podejmowana według następujących zależności:

$$d(n) = 1, \quad \text{dla } \bar{\delta}(n) > \theta$$

$$d(n) = 0, \quad \text{dla } \bar{\delta}(n) \leq \theta.$$

4) Wygładzanie decyzji na poziomie próbek:

Decyzja $d(n)$ dla każdej próbki jest okienkowana o rozmiarach okienek 300msec, 400msec, 600msec, dla, odpowiednio, $\rho < 30$, $30 \leq \rho < 40$, $\rho > 40$. Założymy, że η jest progiem (w procentach) w zależności od wartości $d(n)$, które dają 1 w okienku. Jeżeli wartość procentowa wartości $d(n)$, które wynoszą 1 w okienku, jest wyższa niż wartość η , wtedy ostateczna decyzja $d_f(n)$ jest ustawiana na 1 w chwili czasowej n , w przeciwnym wypadku - 0. Wartość przypisana dla η to 60%.

5) Decyzja na poziomie ramek:

Decyzja w metodzie AMR jest podejmowana dla każdej ramki, co 10msec. W celu porównania zaproponowanej metody z metodą AMR, decyzja $d_f(n)$ jest konwertowana na 10msec ramkę w oparciu o podjętą decyzję. Dla każdej 10msec, nienakładającej się ramki, jeżeli przeważająca ilość decyzji $d_f(n)$ wynosi 1, to cała ramka jest oznaczana jako zawierająca mowę, w przeciwnym wypadku jest oznaczana jako niezawierająca mowy. Informacje dotyczące sygnałów mowy pozyskane z empirycznie, są również otrzymywane z każdej 10msec ramki.

3.4 Zmieniony algorytm Single Frequency Filtering (SFF2)

W dalszej części pracy, jako, że ten algorytm bazuje na założeniach SFF, będzie nazywany algorytmem SFF2. W celu zmniejszenia złożoności obliczeniowej oraz zwiększenia dokładności detekcji, zostały zaproponowane pewne zmiany w algorytmie SFF:

- 1) Zmiana w sposobie obliczania obwiedni sygnału.
- 2) Metoda SFF zakłada obliczenie 185 obwiedni począwszy od 300Hz do 4000Hz co 20Hz. Doświadczalnie sprawdzono, że wystarczające jest obliczenie co czwartej obwiedni zaczynając od 300Hz, oraz zaczynając od 340Hz, co 20Hz każda. Zatem, zamiast 185 obwiedni, liczonych jest jedynie 92.
- 3) Zmiana w sposobie liczenia progu detekcji. Liczone jest 15% z najmniejszych wartości. Tworzony jest również histogram dla $\delta(n)$, na podstawie którego zostaje wyliczonych 15% najmniejszych wartości. Znacząco zwiększa to szybkość wykonywanych obliczeń.

Rozdział 4

Implementacja

Do zaimplementowania algorytmów skorzystano z biblioteki Aquila, która wspiera wykonywanie operacji na sygnałach oraz ze standardowych bibliotek C++, takich jak cmath, STL.

4.1 Metoda oparta o energię sygnału

Implementacja algorytmu bazującego na energii sygnału była dość prosta i jego główna idea, czyli dynamicznie zmieniający się próg, jest zawarta w następujących liniach kodu:

```
//DELTA = 1.00
void EnergyBased :: calculateThresholdEminEmax( Aquila :: WaveFile wav ,
size_t currentFrameNumber ) {

    double singleFrameEnergy=frameEn . countSingleFrameEnergy ( wav , currentFrameNumber ) ;

    //if it ' s first frame
    if ( currentFrameNumber == 0 ) {
        initialValue=singleFrameEnergy ;
        if ( initialValue == 0 ){
            initialValue=1;
            Emax = initialValue ;
            Emin = initialValue ;
        } else {
            Emax = initialValue ;
            Emin = initialValue ;
        }
    }
    if ( currentFrameNumber > 0 ){
        delta=delta * 1.0001;
        Emin=Emin*delta ;
    }
    //if frame energy is higher than Emax
    if ( singleFrameEnergy > Emax ){
        Emax=singleFrameEnergy ;
    }
    //if frame energy is lower than Emin
    else if ( singleFrameEnergy < Emin ) {
        if ( singleFrameEnergy == 0 ){
            Emin=initialValue ;
            //resetDelta
            delta=DELTA;
        }
        else{
            Emin=singleFrameEnergy ;
            //resetDelta
            delta=DELTA;
        }
    }
    else{
        //resetDelta
        delta=DELTA;
    }
}
```

```

scalingFactor=(Emax-Emin)/Emax;
threshold=((1-scalingFactor)*Emax)+(scalingFactor*Emin);

}

```

4.2 Metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości

Tutaj cały proces jest nieco bardziej złożony. Na początek warto przedstawić sposób liczenia obwiedni sygnału, ponieważ jest to jedna z rzeczy, które różnią SFF od SFF2.

```

/*
 * counting Single Frequency Filtering envelope for specific frequency
 * @param wav - signal to process
 * @param normalizedFrequency - specific frequency which will be used to complex sine
 * @return signal's SFF envelope
 */
vector<SampleType> SFF::countSFFEvelope(SignalSource &wav, int normalizedFrequency) {
    //difference the signal
    SignalSource wavDifferenced = differenceSignal(wav);

    //multiply signal x(n) by a complex sinusoid of a given normalized frequency wk
    vector<complex<SampleType>> wavMultipliedByComplexSinusoid = MultiplyByComplexSinusoid(wavDifferenced, n);

    //count output yk(n) = -r*yk(n-1) +xk(n)
    vector<complex<double>> wavFilterOutput = countFilterOutput(wavMultipliedByComplexSinusoid);

    //count envelope
    vector<SampleType> wavEnvelope = countEnvelope(wavFilterOutput);

    return wavEnvelope;
}

/**
 * multiplies signal by complex sinusoid
 * @param source - wav signal
 * @param frequency - signal sampling frequency
 * @return vector with complex values
 */
vector<complex<SampleType>> SFF::MultiplyByComplexSinusoid(const SignalSource& wav, int normalizedFrequency) {
    vector<SampleType> wavVector(wav.begin(), wav.end());
    vector<complex<SampleType>> result;
    //complex sinusoid
    const double pi = acos(-1);
    const complex<double> j(0, 1);

    double omega_k = (2*pi*normalizedFrequency)/wav.getSampleFrequency();
    for (vector<SampleType>::iterator it=wavVector.begin(); it!=wavVector.end(); ++it){
        long index = std::distance(wavVector.begin(), it);
        //exp(j*sr_omega_k*n)
        result.push_back(*it*exp(j*omega_k*(double)index));
    }

    return result;
}

/**
 * Single-pole filter has a pole on the real axis at a distance of r from the origin.
 * the location of the root is at z=r in the z-plane, which corresponds to half the sampling
 * frequency fs/2
 * The output of the filter is given by
 * yk(n) = -r*yk(n-1) +xk(n)
 * @param wav - vector to filter
 * @return filter's output
 */


```

```

*/
vector<complex<double>> SFF::countFilterOutput (vector<complex<SampleType>> &wav) {
const double root = 0.99; //article

vector<complex<double>> output;
//y(0)=0
output.push_back((0.0,0.0));

//iterating starts at index=1, because we use output[index-1]
for (vector<complex<SampleType>>::iterator signalSample=wav.begin() + 1; signalSample!=wav.end(); ++signalSample)
    unsigned long currentIndex = (unsigned long)distance(wav.begin(), signalSample);
    output.push_back(output.at(currentIndex - 1) *(-root) + *signalSample);
}

return output;
}

/**
* Function counting values for signal's envelope
* @param complexSignal – signal with complex samples
* @return signal's envelope
*/
vector<SampleType> Detector::countEnvelope (vector<complex<double>> &complexSignal) {
vector<SampleType> envelope;
for (vector<complex<SampleType>>::iterator signalSample=complexSignal.begin(); signalSample!=complexSignal.end();
envelope.push_back(sqrt((*signalSample).real()*(*signalSample).real() + (*signalSample).imag()*
}

return envelope;
}

```

Podczas obliczania progu, aby wybrać 20% najmniejszych wartości energii $\delta(n)$, wykorzystano bardzo mało wydajną metodę, która zakłada w pierwszej kolejności posortowanie próbek rosnąco i wybranie z nich kolejnych 20%. Miało to na celu zapewnienie, że wybierane są właściwe próbki do obliczenia progu. Głównym celem niniejszej pracy było sprawdzenie jakości detekcji algorytmów, a nie ich wydajności. Przy obliczaniu progu pojawia się problem związany z wcześniejszym wygładzeniem $\delta(n)$. Autor algorytmu podaje sposób, w jaki należy dobrać rozmiar okna wygładzającego oraz jak je zastosować, ale nie wspomina, jak powinny zostać wygładzone skrajne przebiegi $\delta(n)$ (po pół okna z każdej strony pozostaje nieśrednione). Pozostawienie tych brzegów niewygładzonych skutkuje dobraniem bardzo małego progu detekcji, przez który cały przebieg sygnału zaliczany jest do aktywności mówcy, natomiast wygładzanie tych fragmentów z wykorzystaniem mniejszego okna nie dawało wystarczającego efektu wygładzenia. Zostało to zaimplementowane w następujący sposób - najpierw wygładzana jest część główna sygnału, a następnie początek oraz koniec. Główna część sygnału jest wygładzana poprzez policzenie średniej arytmetycznej z próbek z okna i przypisać jej wartość do próbki położonej w środku okna (dlatego okno powinno mieć nieparzystą liczbę próbek). Następnie obliczane są średnie dla początkowych próbek - dla tych położonych poniżej połowy długości okna, ale w taki sposób, by wykorzystywane próbki nie wykraczały poza rozmiar wektora, rozmiar okna pozostaje bez zmian. Poniższy kod prezentuje wygładzanie sygnału.

```

vector<SampleType> SFF::averageVector (vector<SampleType>& vectorToAverage, double windowHeight) {
    vector<SampleType> averaged (vectorToAverage.size(), 0);

    int samplesPerWindow = (int) (_samplingFrequency * windowHeight);
    if(samplesPerWindow%2 == 0){
        samplesPerWindow += 1;
    }

    int halfOfSamplesPerWindow = (samplesPerWindow - 1)/2;

    //main smoothing
    for (vector<SampleType>::iterator it=vectorToAverage.begin() + halfOfSamplesPerWindow;

```

```

it!=vectorToAverage.end()-halfOfSamplesPerWindow;
it++){  

    double average = accumulate( it-halfOfSamplesPerWindow ,
        it+halfOfSamplesPerWindow+1,
        0.0)/samplesPerWindow;
    long index = distance(vectorToAverage.begin(), it );
    averaged[index]=average;
}  

//      smoothing beginning
int counter(0);
for (vector<SampleType>:: iterator it=vectorToAverage.begin();
it!=vectorToAverage.begin()+halfOfSamplesPerWindow;
it++){  

    double average = accumulate( it-counter ,
        it+samplesPerWindow-counter ,
        0.0)/samplesPerWindow;
    long index = distance(vectorToAverage.begin(), it );
    averaged[index]=average;
    counter++;
}  

//smoothing end
counter=0;
for (vector<SampleType>:: iterator it=vectorToAverage.end()-halfOfSamplesPerWindow;
it!=vectorToAverage.end();
it++){  

    double average = accumulate( it-halfOfSamplesPerWindow-counter ,
        it+halfOfSamplesPerWindow-counter ,
        0.0)/samplesPerWindow;
    long index = distance(vectorToAverage.begin(), it );
    averaged[index]=average;
    counter++;
}
return averaged;
}

```

warto również pokazać, w jaki sposób liczony jest próg detekcji θ . Najpierw z wyznaczonej wcześniej funkcji $\delta(n)$ wybieramy 20% najmniejszych wartości próbek i liczymy dla nich średnią (μ_θ) oraz wariancję (σ_θ). Próg wyznaczany jest zgodnie ze wzorem:

$$\theta = \mu_\theta + 3\sigma_\theta$$

```

/***
* Compute the mean (mi_theta) and the variance (sigma_theta) of the lower 20% of the values of delta(n)
* A threshold of theta = mi_theta + 3*sigma_theta is used in all cases.
* The theta value depends on each utterance
*/
double SFF::countThresholdTheta(vector<SampleType> delta , double smoothingWindowSize ) {
    ///get 20% first samples
    int amountOfNotSmoothedSamples = (int)(_samplingFrequency * smoothingWindowSize);
    int amountOf20PercentOfSamples = (int)(delta.size()*0.2);

    ///sort delta and get 20% first values
    sort(delta.begin(), delta.end());

    vector<SampleType> splitedDelta(delta.begin(), delta.begin() + amountOf20PercentOfSamples);

    ///count mean and variance for splitedDelta

    double mean = accumulate(splitedDelta.begin(),
        splitedDelta.end(),
        0.0)/splitedDelta.size();

    ///variance: ((1/(n-1) * sum(i=1, n, (xi-x_sr)^2)
    double variance(0.0);
    for(auto& x : splitedDelta){
        variance += (x-mean)*(x-mean);
    }
    variance = variance/(splitedDelta.size() - 1);
}

```

```

    ///threshold theta
    double threshold = mean + 3.0 * variance;
    return threshold;
}

```

4.3 Zmieniona metoda bazująca na obwiedni sygnału podzielonego na pasma z filtracją pojedynczych częstotliwości

W tym przypadku wyznaczenie obwiedni sygnału wygląda trochę inaczej, co przedstawia poniższy kod:

```

vector<SampleType> wavEnvelope;
const double pi = acos(-1);

double omega = 2*pi*normalizedFrequency / _samplingFrequency;
double module = 0.97;

///filter factors
double a1 = module * cos(omega);
double a2 = module * sin(omega);

///filter initialization
vector<double> XReal(wavDifferenced.getSamplesCount(), 0);
vector<double> XIImaginary(wavDifferenced.getSamplesCount(), 0);

///recursive quadrature filter
for (int i = 1; i < wavDifferenced.getSamplesCount(); i++) {
    XReal[i] = wavDifferenced.sample(i) + a1*XReal.at(i-1) - a2*XIImaginary.at(i-1);
    XIImaginary[i] = a1*XIImaginary.at(i-1) + a2*XReal.at(i-1);
}

///envelope
for (int j = 0; j < wavDifferenced.getSamplesCount(); j++) {
    wavEnvelope.push_back(sqrt(XReal.at(j)*XReal.at(j)*XIImaginary.at(j)*XIImaginary.at(j)));
}

```

Liczenie progu detekcji β polega na wygenerowaniu histogramu, w którym można wyróznić dwa maksima lokalne - pierwsze z nich to rozkład wartości szumu w sygnale, a drugie to rozkład wartości mowy. Wartość progu wyznaczona zostaje przy pomocy wartości minimum lokalnego zlokalizowanego pomiędzy wyznaczonymi wcześniej maksimami.

```

/**
 * Compute the threshold beta using histogram
 */
double SFFChanged::countThresholdBeta(vector<SampleType> delta) {
    double maxDelta(0);

    ///density
    maxDelta = *max_element(delta.begin(), delta.end());
    //histogram
    //AMOUNT_OF_DENSITY_POINTS = 801
    vector<double> density = countDensityForPositiveValues(delta, AMOUNT_OF_DENSITY_POINTS, maxDelta);

    ///smoothing
    int amountOfLoops(20);
    density = smoothSignal(density, amountOfLoops);

    ///left max - in range(200, 600) - based on experience
    double dM(0.0);
    int sM(0);
    for (int i = 200; i < 600; i++) {

```

```
    if (density.at(i)>dM){  
        dM = density.at(i);  
        sM = i;  
    }  
}  
int leftMax = sM;  
///right max - in range(601, 800)  
dM = 0.0;  
sM = 0;  
for (int i = 601; i < 800; i++) {  
    if (density.at(i)>dM){  
        dM = density.at(i);  
        sM = i;  
    }  
}  
int rightMax = sM;  
  
dM = 10;  
sM = 0;  
for (int i = leftMax; i < rightMax; i++) {  
    if (density.at(i)<dM){  
        dM = density.at(i);  
        sM = i;  
    }  
}  
double dist = maxDelta / 800;  
double threshold = sM*dist;  
  
return threshold;  
}
```

Rozdział 5

Wyniki badań

5.1 Sposób oceny

Do przeprowadzenia badań zostały wybrane 2 sygnały (głos damski oraz głos męski) zawierające pojedyncze słowa oraz 2 sygnały zawierające ciągi słów. Pierwszy etap badań założył manualne wybranie próbek zawierających początki oraz końce wypowiedzi, rozważony został przypadek, gdy mowa rozpoczyna się w momencie generowania słyszalnych dźwięków przez mówcę. Następnie te same sygnały zostały zbadane przez 3 algorytmy detekcji, które również wybrały numery próbek, które miały reprezentować początki i końce mowy. Dla każdego pomiaru wykonano 10 prób i ostateczne wartości poddane badaniom są średnią wartością z tych prób. Jako główne kryterium oceny przyjęto odległość zdetectowanego rozpoczęcia lub końca aktywności od rozpoczęcia lub końca aktywności ze wzorca (w próbkach).

Następnie została policzona wartość bezwzględna próbki wybranej manualnie oraz próbki wybranej przez algorytm. Często pojawia się sytuacja, że któryś z algorytmów podjął decyzję o większej lub mniejszej ilości początków lub końców mowy. W związku z tym dodano nowe kryterium, jakim jest ilość detekcji. Za każdy nadprogramowy początek lub koniec doliczana jest pewna wartość, którą można potraktować jako karę za błąd. Przedstawione w tabelach wartości *Wynik* zostały obliczone zgodnie ze wzorem

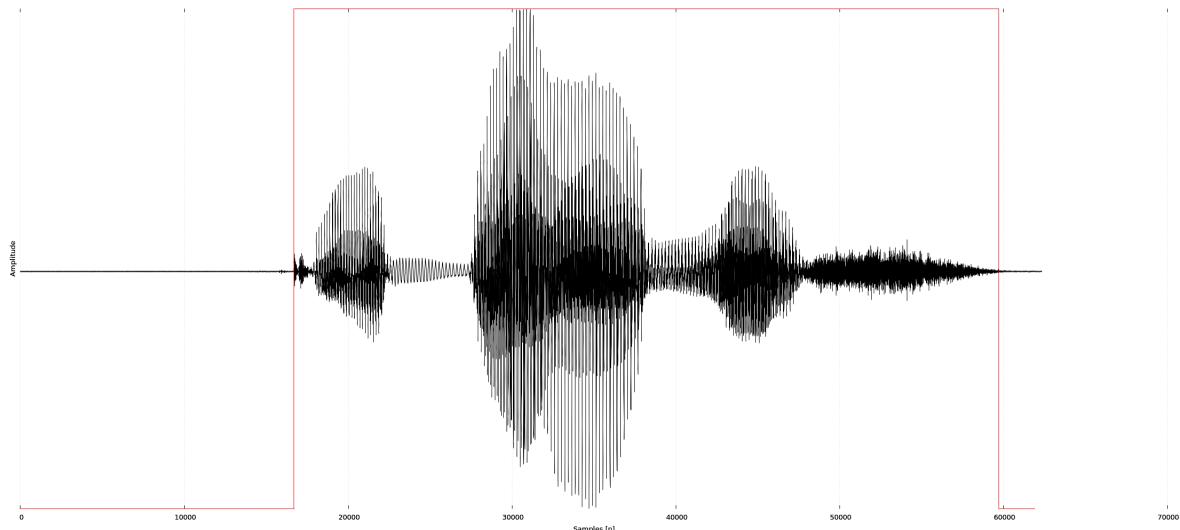
$$Wynik = |d_{wzór} - d_{alg}| * kara + \sum_{i=1}^j |x_{wzór}(i) - x_{alg}(i)|$$

,

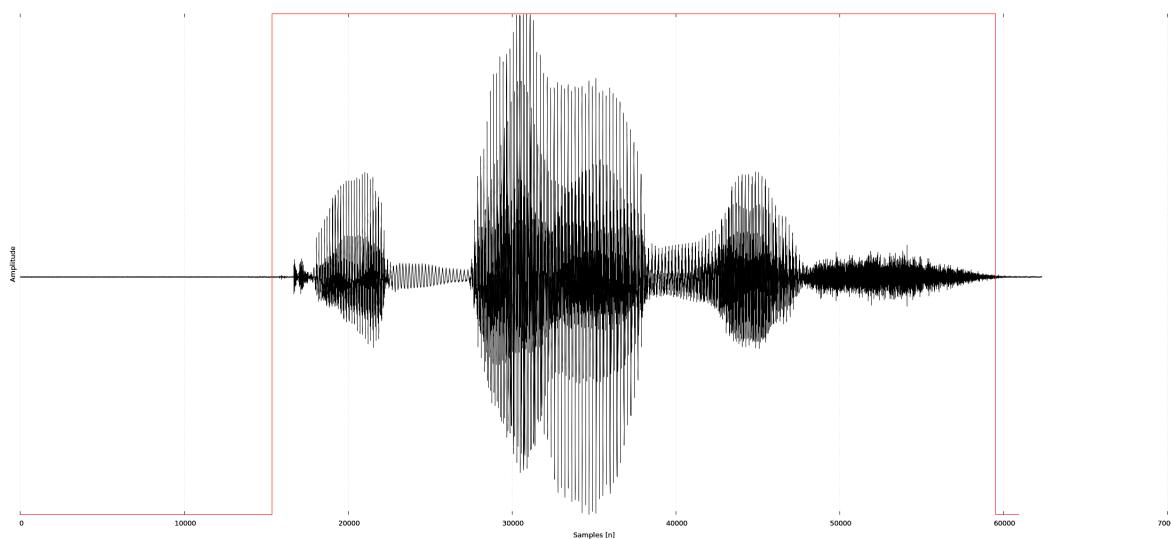
gdzie $d_{wzór}$ to sumaryczna ilość początków i końców detekcji we wzorze, d_{alg} to sumaryczna ilość początków i końców detekcji wybranego algorytmu, $kara$ to stała wartość, w tym przypadku ustawiona na 5000, i to numer kolejnej detekcji, j to ilość wszystkich detekcji, $x_{wzór}(i)$ to nr próbki wybranej manualnie, a $x_{alg}(i)$, to numer próbki wybranej przez algorytm. Wyniki dla poszczególnych algorytmów, w dwóch rozważanych przypadkach zostały przedstawione w tabeli poniżej. Oznaczenia algorytmów: En - algorytm bazujący na energii sygnału, SFF - algorytm bazujący na obwiedni sygnału, SFF2 - zmieniony algorytm SFF.

5.2 Wyniki dla pojedynczych słów

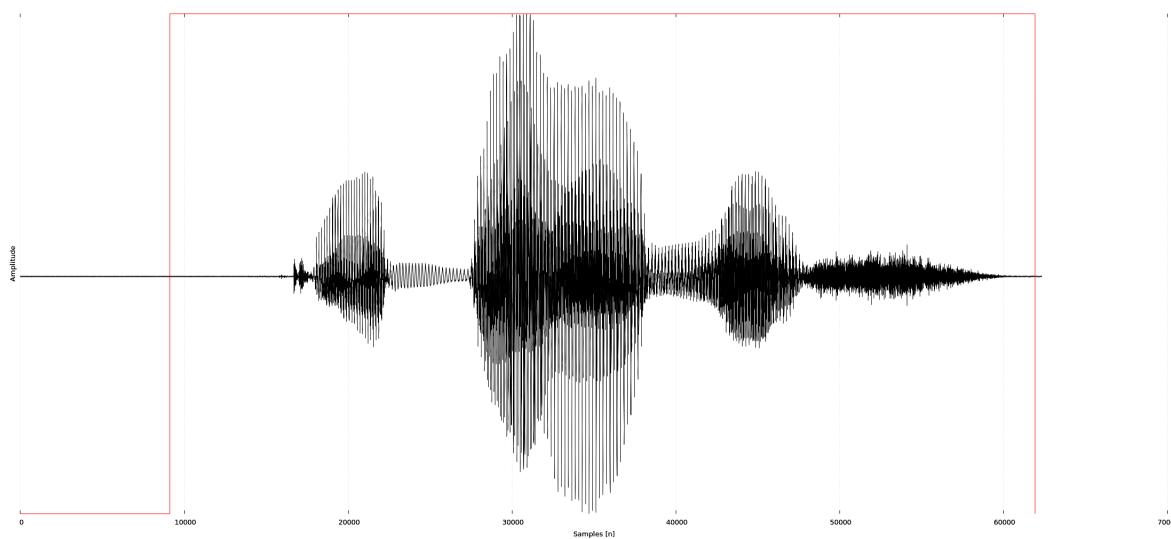
Na rysunku 5.1 przedstawiono przebieg czasowy wyrazu *kabanos*, głos damski, a na rysunku 5.5 przebieg czasowy wyrazu *zapamiętaj*, głos męski, które zostaną poddane detekcji przez algorytmy, które zostały zaprezentowane w poprzednich rozdziałach. Na rysunkach 5.1 oraz 5.5 detekcja została ustawiona manualnie, jako punkt odniesienia dla innych algorytmów. Kolejne rysunki zawierają wyniki detekcji.



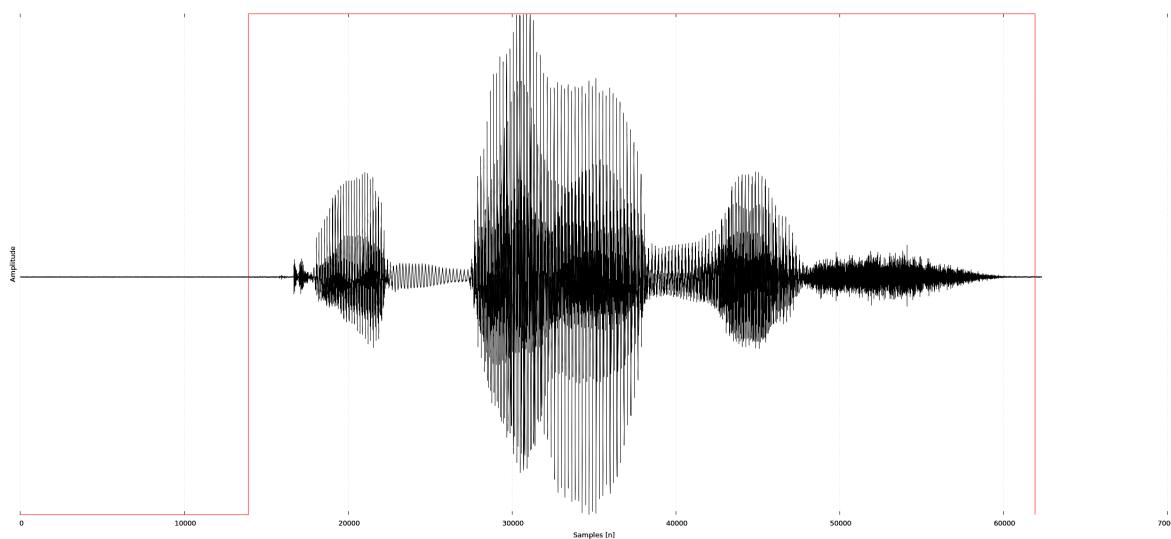
Rysunek 5.1 Przebieg czasowy słowa *kabanos* i jego wzorcowa detekcja



Rysunek 5.2 Wynik detekcji algorytmu bazującego na energii sygnału



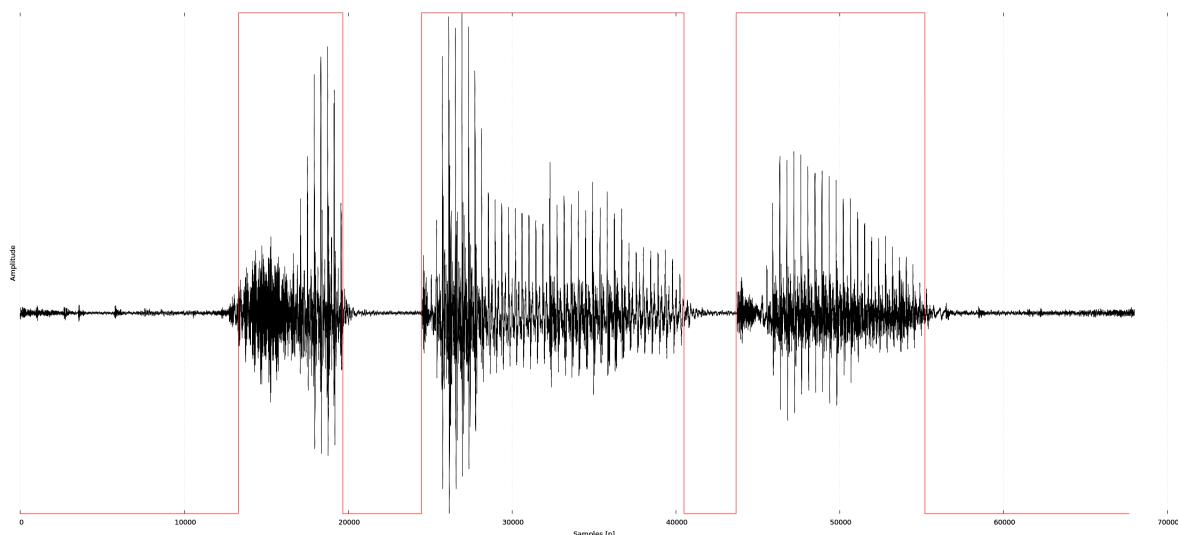
Rysunek 5.3 Wynik detekcji algorytmu SFF



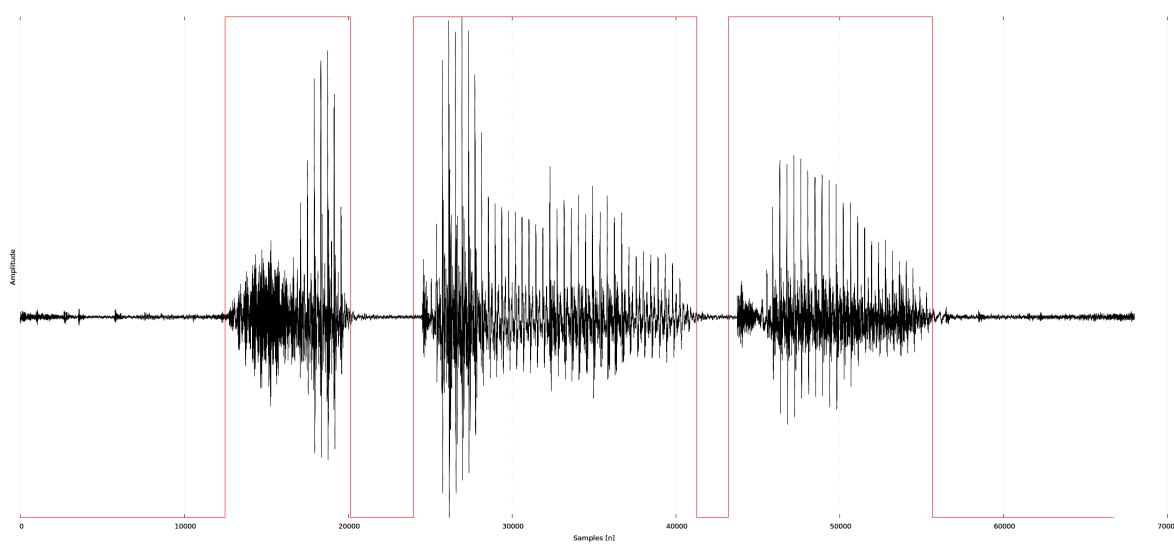
Rysunek 5.4 Wynik detekcji algorytmu SFF2

	Wzorzec	En[nr probki]	SFF[nr probki]	SFF2[nr probki]
Początek	6705	15360	9121	13921
Koniec	59712	60000	61920	61920
SUMA RÓŻNIC	0	1633	9792	4992

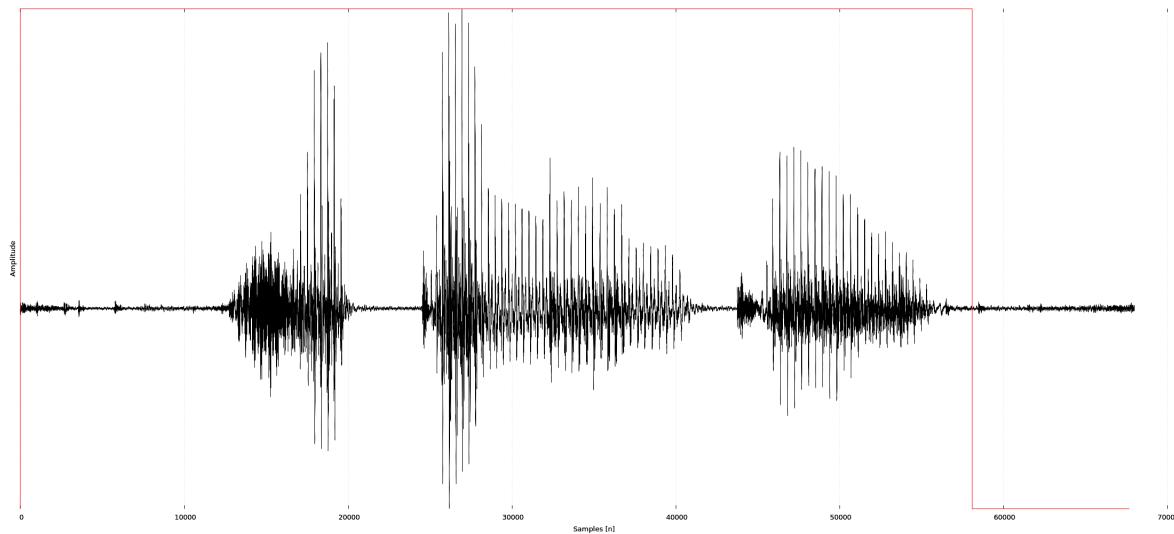
Rysunek 5.5 Wyniki działania algorytmów na słowie *kabanos*



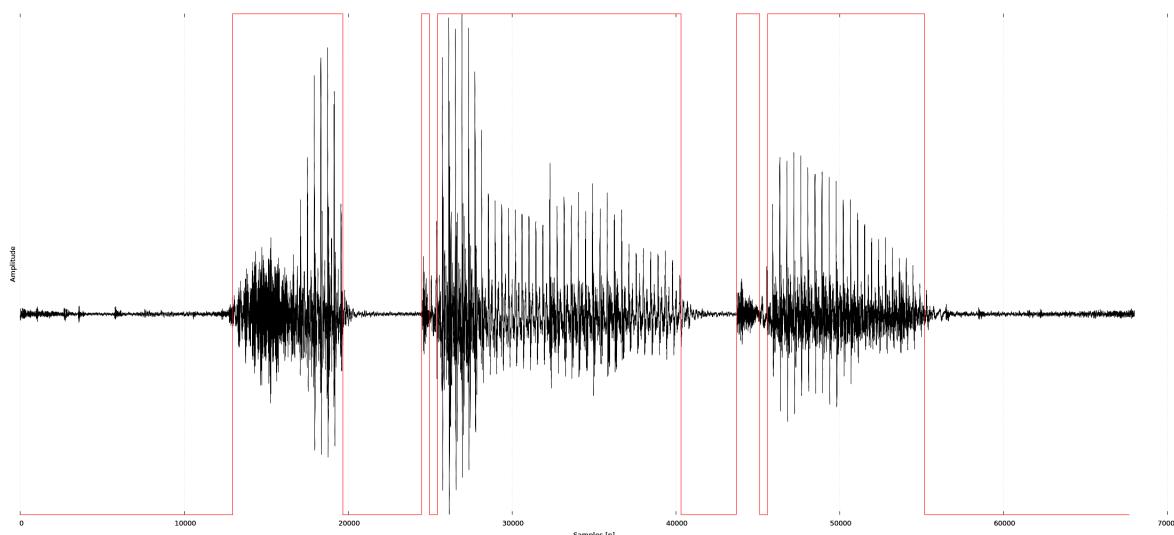
Rysunek 5.6 Przebieg czasowy słowa *zapamiętaj* i jego wzorcowa detekcja



Rysunek 5.7 Wynik detekcji algorytmu bazującego na energii sygnału



Rysunek 5.8 Wynik detekcji algorytmu SFF



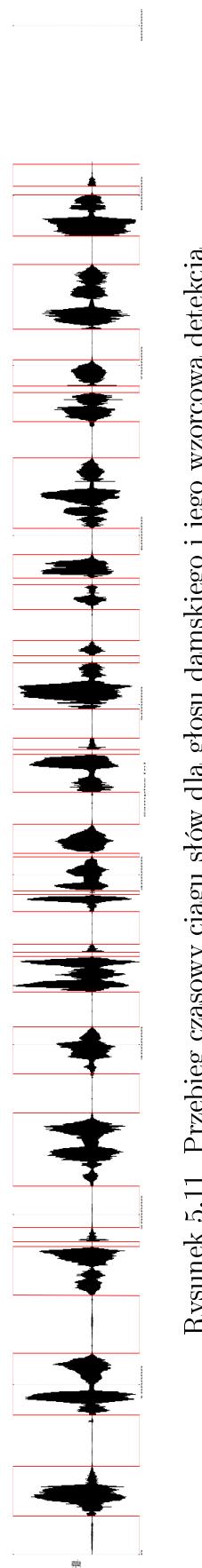
Rysunek 5.9 Wynik detekcji algorytmu SFF2

	<i>Wzorzec</i>	<i>En[nrprobki]</i>	<i>SFF[nrprobki]</i>	<i>SFF2[nrprobki]</i>
<i>Pocztek1</i>	13296	12480	2	12961
<i>Koniec1</i>	19680	20160		19681
<i>Pocztek2</i>	24480	24000		24481
<i>Koniec2</i>	40512	41280		40321
<i>Pocztek3</i>	43680	43200		43681
<i>Koniec3</i>	55200	55680	58081	55201
SUMA RÓŻNIC	0	3504	144527	530
Ilość detekcji	6	6	2	10
Wynik	0	3504	36175	20530

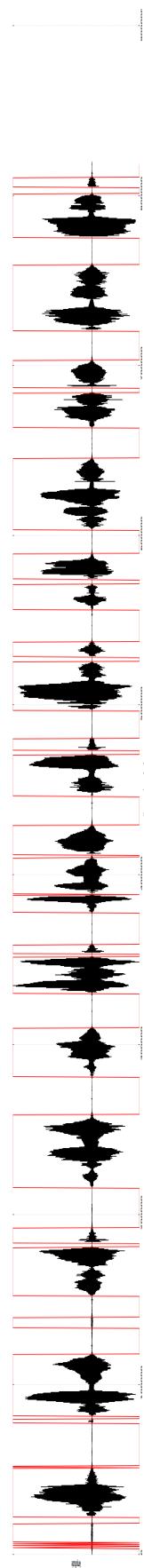
Rysunek 5.10 Wyniki działania algorytmów na słowie *zapamiętaj*

5.3 Wyniki dla ciągów słów

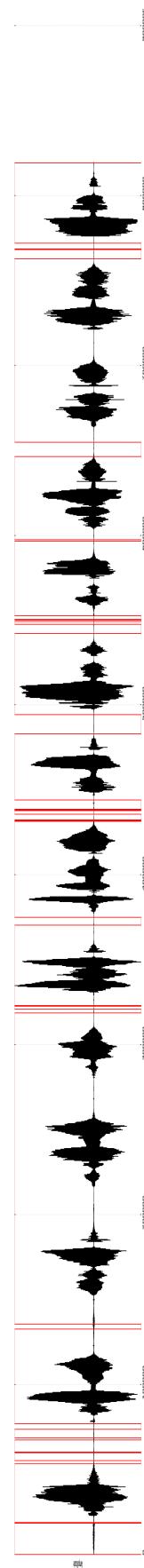
Podobnie jak w poprzednim przypadku, na rysunku 5.11 przedstawiono przebieg czasowy kolejnych słów: *grzech, poprzedni, mózdzek, widzenie, dzwoń, proszek, zapamiętaj, dziwak, radzić, fiska, szeregi, gorzki, kiedyś, ogród, głos damska, natomiast na rysunku 5.16 pokazano wzorcową detekcję przebiegu czasowego kolejnych słów: poczta, buczeek, turniej, radża, wyczyść, sek, sześć, leżeć, dzień, fajka, wzrok, łazik, zlepek, poziomo, głos męski. Zaznaczono na nich oczekiwana (wzorcowa) detekcję algorytmów.* Kolejne rysunki zawierają wyniki detekcji algorytmów.



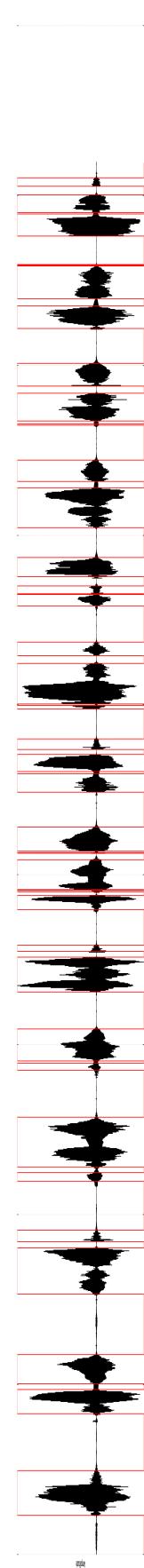
Rysunek 5.11 Przebieg czasowy ciągu słów dla głosu damskego i jego wzorcowa detekcja



Rysunek 5.12 Wynik detekcji algorytmu bazującego na energii sygnału



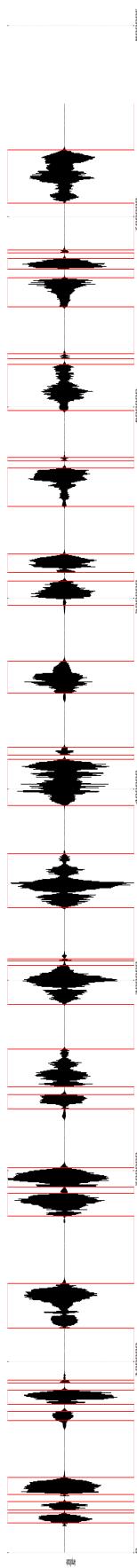
Rysunek 5.13 Wynik detekcji algorytmu SFF



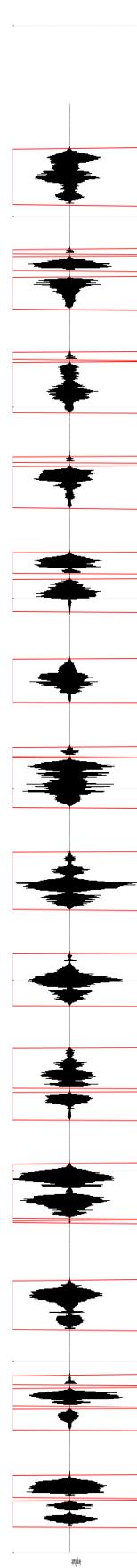
Rysunek 5.14 Wynik detekcji algorytmu SFF2

	Wzorzec[nr próbki]	En[nr próbki]	SFF[nr próbki]	SFF2[nr próbki]
	22506	22080	18721	23041
	51738	50880	53281	49441
	82035	81600	76801	82561
	118179	118080	132481	117601
	152457	152160	135361	153121
	181022	180960		180481
	183927	183360		183841
	192255	192480		190561
	216882	216000		219841
	259958	259200		257281
	282772	281280		285121
	310413	310080	318721	309121
	330914	330240	323041	331201
	351839	351840	370561	351361
	354455	353760		354721
	359172	359040		358561
	378238	377760	374881	379681
	388346	388320		387841
	390280	388800		390241
	410592	410400		408961
	412623	411840		412801
	429792	429600	431521	428161
	448556	446400	444001	448801
	470851	470880		470881
	473898	473280		473761
	480523	480480	482881	480001
	497692	496800	494401	497761
	525016	525600		524641
	528933	528480		528961
	537880	537600	541921	537121
	556209	556320	552481	558241
	570960	571200		570241
	574780	574080		575521
	588322	589440	596641	587041
	603943	603360	597121	604321
	645333	645600	646081	644161
	666836	663360	654721	664801
	683855	684000		683521
	687597	686880		687841
	703231	703200		701281
	721172	720480		721441
	759463	759360	762721	759361
	775968	775200	771841	776161
	800470	801120		800161
	805494	804960		805441
	818400	810720	819360	810241
SUMA RÓŻNIC	0	2177	222982	5141
Ilość detekcji	46	60	44	72
Wynik	0	72177	232982	135141

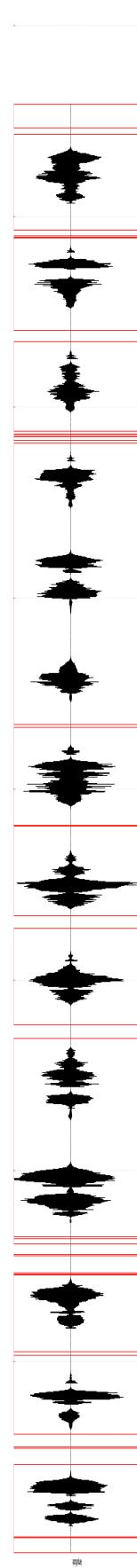
Rysunek 5.15 Wyniki detekcji algorytmów dla ciągu słów, głos damski



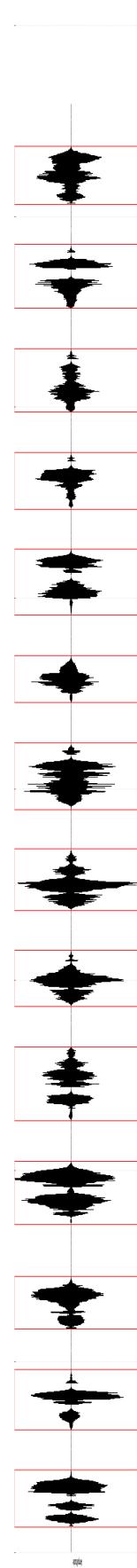
Rysunek 5.16 Przebieg czasowy ciągu słów dla głosu damskiego i jego wzorcowa detekcja



Rysunek 5.17 Wynik detekcji algorytmu bazującego na energii sygnału



Rysunek 5.18 Wynik detekcji algorytmu SFF



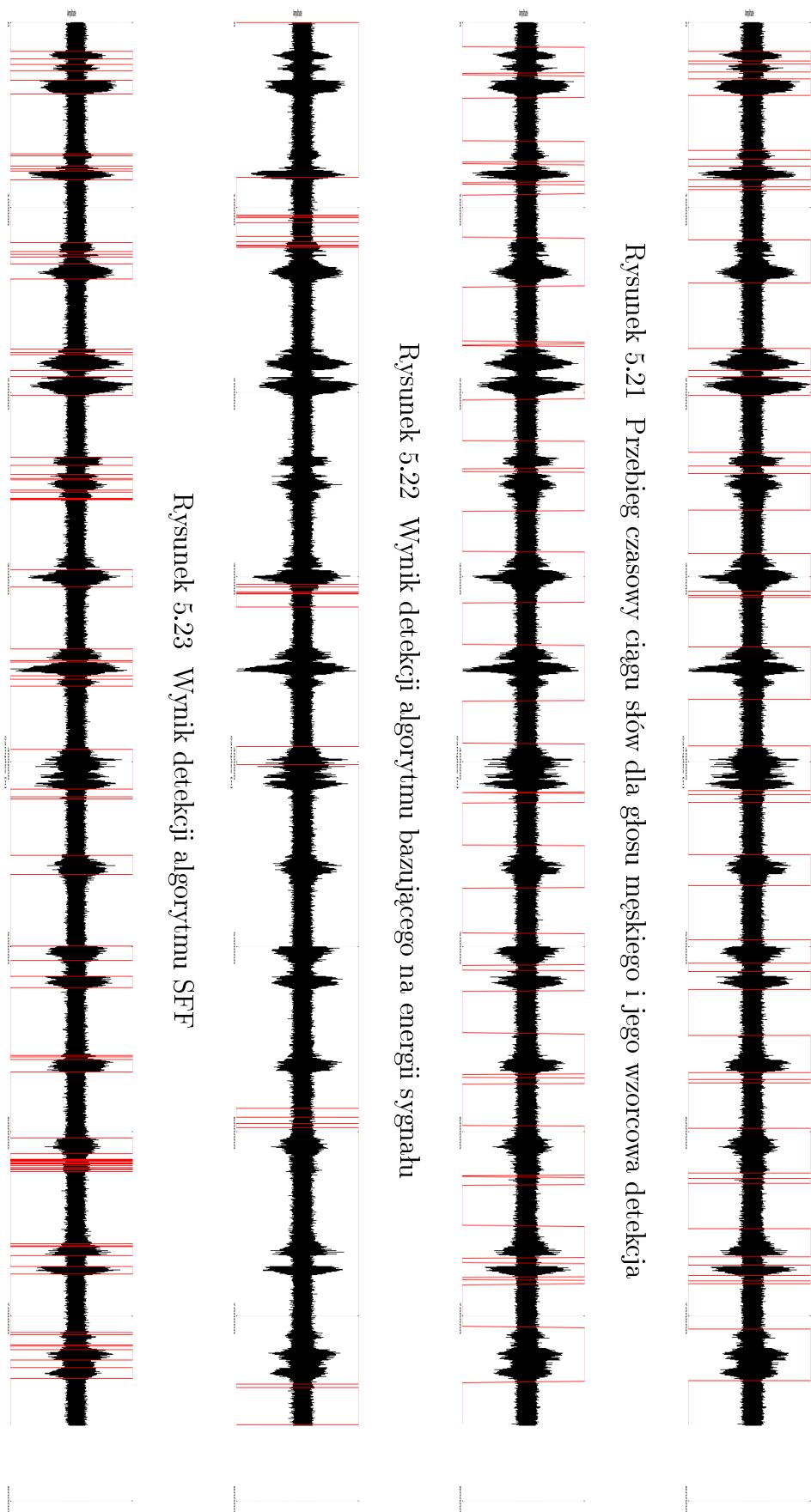
Rysunek 5.19 Wynik detekcji algorytmu SFF2

	Wzorzec	En[nr probki]	SFF[nr probki]	SFF2[nr probki]
	15306	13440	8161	13441
	20545			
	22161			
	26800	27360		
	30362	28800		
	39144	40800	46081	43201
	68926	64320	61921	64321
	74062	75360		
	77542	76800		
	85080	86400		
	88725	87840		
	90341	93120	103201	96001
	117389	116640	105121	117121
	140751	143040	145441	144481
	176084	175200	165601	171841
	188096			
	191534			
	201600	204000		204961
	232500	226560		226081
	239956	241440		
	244140	243360		
	263732	264480	269281	264961
	287260	286560	276481	286081
	307634			
	309728			
	311125	313920	326881	315361
	337591	336960	333601	336481
	366025	367200	380641	368641
	391540	390240	381121	389281
	415468	416640		
	417943	417120		
	421814	422400	432481	424321
	450122	445440	433921	445441
	467068	468480		469921
	496200	492960		491041
	509084	510240		
	513591	513120		
	522984	524160		525601
	548054	547200		546721
	568174	569280		
	571918	571200		
	573696	574560	581281	576481
	598131	597120	587521	597601
	622503	624000		
	625486	624960		
	627898	629280	634561	630721
	652714	651360	640321	651841
	668010	668640		
	672390	671520		
	678102	679200		
	681085	680640		
	682291	683040	688801	685441
	706853	706080	692641	706081
	734970	735840	743041	736801
SUMA RÓŻNIC	0	8764	281165	261005
Ilość detekcji	54	50	40	28
Wynik	0	28764	351165	391005

Rysunek 5.20 Wyniki detekcji algorytmów dla ciągu słów, głos męski

5.4 Powtórzenie badań przy większym poziomie zaszu-mienia

W wyniku osiągnięcia zadowalających efektów działania algorytmów na sygnałach z minimalnym poziomem zaszumienia, postanowiono sprawdzić, jakie wyniki dzadzą zastosowane na sygnałach o dużo wyższym poziomie zaszumienia. Zastosowano do tego szum Gaussowski o średniej $\mu = 0.0$ oraz o odchyleniu standardowym $\sigma = 1000$



Rysunek 5.21 Przebieg czasowy ciągu słów dla głosu męskiego i jego wzorcowa detekcja

Rysunek 5.22 Wynik detekcji algorytmu bazującego na energii sygnału

Rysunek 5.23 Wynik detekcji algorytmu SFF

Rysunek 5.24 Wynik detekcji algorytmu SFF2

Rysunek 5.25 Wyniki detekcji algorytmów dla ciągu słów, głos męski

Rozdział 6

Podsumowanie

Bibliografia

Spis rysunków

2.1	Aparat mowy człowieka	6
2.2	Sprzężenie zwrotne w wytwarzaniu mowy	6
2.3	Przebieg sygnału w dziedzinie czasu dla słowa <i>kabanos</i>	7
2.4	Widmo sygnału dla słowa <i>kabanos</i>	7
5.1	Przebieg czasowy słowa <i>kabanos</i> i jego wzorcowa detekcja	26
5.2	Wynik detekcji algorytmu bazującego na energii sygnału	27
5.3	Wynik detekcji algorytmu SFF	27
5.4	Wynik detekcji algorytmu SFF2	27
5.5	Wyniki działania algorytmów na słowie <i>kabanos</i>	28
5.6	Przebieg czasowy słowa <i>zapamiętaj</i> i jego wzorcowa detekcja	28
5.7	Wynik detekcji algorytmu bazującego na energii sygnału	29
5.8	Wynik detekcji algorytmu SFF	29
5.9	Wynik detekcji algorytmu SFF2	29
5.10	Wyniki działania algorytmów na słowie <i>zapamiętaj</i>	30
5.11	Przebieg czasowy ciągu słów dla głosu damskego i jego wzorcowa detekcja	31
5.12	Wynik detekcji algorytmu bazującego na energii sygnału	31
5.13	Wynik detekcji algorytmu SFF	31
5.14	Wynik detekcji algorytmu SFF2	31
5.15	Wyniki detekcji algorytmów dla ciągu słów, głos damski	32
5.16	Przebieg czasowy ciągu słów dla głosu damskego i jego wzorcowa detekcja	33
5.17	Wynik detekcji algorytmu bazującego na energii sygnału	33
5.18	Wynik detekcji algorytmu SFF	33
5.19	Wynik detekcji algorytmu SFF2	33
5.20	Wyniki detekcji algorytmów dla ciągu słów, głos męski	34
5.21	Przebieg czasowy ciągu słów dla głosu męskiego i jego wzorcowa detekcja .	36
5.22	Wynik detekcji algorytmu bazującego na energii sygnału	36
5.23	Wynik detekcji algorytmu SFF	36
5.24	Wynik detekcji algorytmu SFF2	36
5.25	Wyniki detekcji algorytmów dla ciągu słów, głos męski	37

Spis tabel