1 Реализованные алгоритмы

1.1 Алгоритм нахождения мел-частотных кепстральных коэффициентов

Запишем исходный речевой сигнал как

$$x_n, \;\; 0 \leq n < N \tag{1}$$

Получим спектр сигнала, используя ДПФ

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i}{N} kn}, \;\; 0 \leq k < N \tag{2}$$

Определим оконные функции. В данном случае использованы M треугольных окон, равномерно расположенные относительно мел-шкалы

$$H_m = \begin{cases} 0, & k < f_{m-1} \\ \frac{(k - f_{m-1})}{(f_m - f_{m-1})}, & f_{m-1} \leq k < f_m \\ \frac{(f_{m+1} - k)}{(f_{m+1} - f[m])}, & f_m \leq k \leq f_{m+1} \\ 0, & k > f_{m+1} \end{cases} \tag{3}$$

Граничные частоты f_m получены из равенства

$$f_m = (\frac{N}{F_s}) \cdot B^{-1}(B(f_1) + m\frac{B(f) - B(f_1)}{M+1}), \;\; 0 \leq m < M \tag{4}$$

где $B(f)$ — операция перевода значений частоты в мел-шкалу

$$B(f) = 1125 \ln(1 + f/700) \tag{5}$$

Соответственно, обратная операция

$$B^{-1}(b) = 700(exp(b/1125) - 1) \tag{6}$$

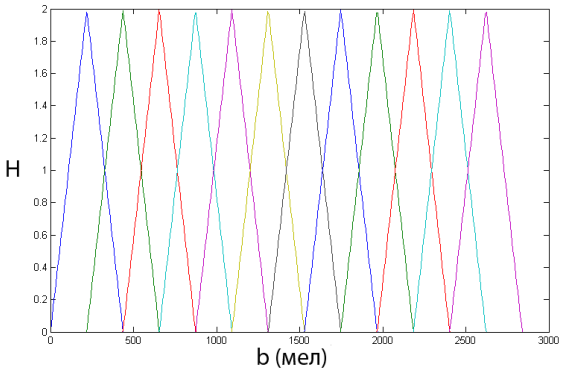

Рис. 1: Окна (мел-шкала)

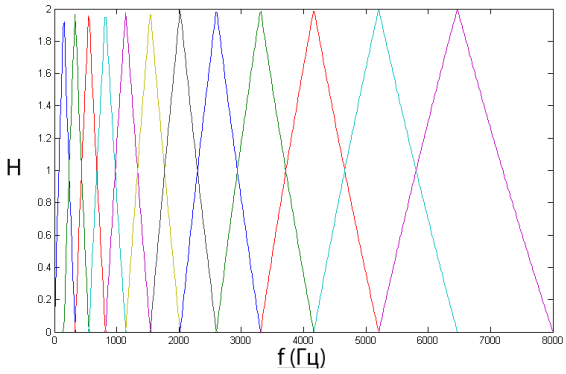

Рис. 2: Окна (частотная шкала)

Вычислим логарифм значения энергии сигнала для каждого окна анализа

$$S_m = ln(\sum_{k=0}^{N-1} |X_k|^2 H_{m,k}), \;\; 0 \leq m < M \tag{7}$$

Чтобы получить набор мел-частотных кепстральных коэффициентов, к полученным значениям применим ДКП

$$c_n = \sum_{m=0}^{M-1} S_m cos(\pi n(m + 1/2)/M), \;\; 0 \leq n < M \tag{8}$$

Полученные в результате значения, а также их изменения во времени, используются в дальнейшем как описание речевого сигнала.

1.2 Алгоритм сравнения речевых сигналов с применением динамического программирования (DTW)

Алгоритм динамического трансформирования времени (DTW) вычисляет оптимальную последовательность трансформации (деформации) времени между двумя временными рядами. Алгоритм вычисляет оба значения деформации между двумя рядами и расстоянием между ними.

Предположим, что есть две числовые последовательности $A = a_1, a_2, \ldots, a_I$ и $B = b_1, b_2, \ldots, b_J$. Длина двух последовательностей может быть различной.

Временные различия между A и B могут быть описаны с помощью некоторой последовательности $c = (i, j)$:

$$F = c(1), c(2) \ldots, c(k), \ldots, c(K) \tag{9}$$

где $c(k) = (i(k), j(k))$. Данная последовательность представляет собой функцию, которая позволяет отобразить временную ось A на временной оси B. Назовем ее функцией деформации. [3]

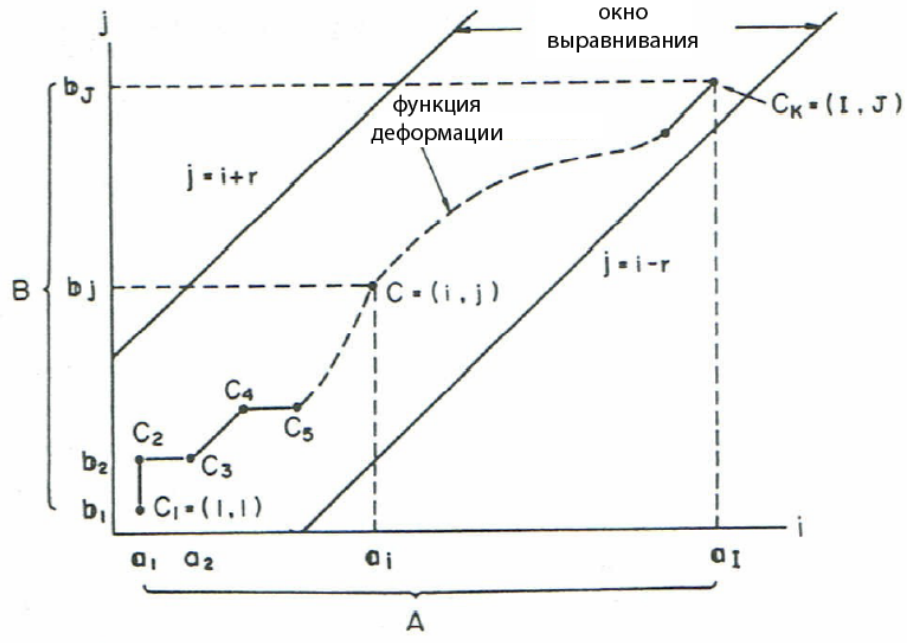

Рис. 3: Функция деформации и окно выравнивания

Алгоритм начинается с расчета локальных расстояний между элементами

двух последовательностей. Самый распространенный способ для вычисления расстояний является метод, рассчитывающий модуль разности между значениями двух элементов (Евклидова метрика). В результате получаем матрицу расстояний, имеющую I строк и J столбцов общих членов:

$$d(c) = d(i, j) = |a_i - b_j|, i = 1..I, j = 1..J \tag{10}$$

Взвешенная сумма значений метрик в точках, принадлежащих функции деформации F

$$E(F) = \sum_{k=1}^{K} d(c(k)) \cdot w(k) \tag{11}$$

(где $w(k)$ - неотрицательный весовой коэффициент) является мерой доброкачественности функции F. Она принимает минимальное значение, когда функция F оптимально выравнивает временные различия между A и B. Минимальное остаточное расстояние между A и B, которое остается после устранения временных различий, может служить мерой различия речевых последовательностей A и B

$$D(A, B) = min_F \left[\frac{\sum_{k=1}^{K} d(c(k)) \cdot w(k)}{\sum_{k=1}^{K} w(k)} \right] \tag{12}$$

Существует три условия, налагаемых на DTW алгоритм для обеспечения быстрой конвергенции:

1. Монотонность – путь никогда не возвращается, то есть: оба индекса, i и j, которые используются в последовательности, никогда не уменьшаются.

2. Непрерывность – последовательность продвигается постепенно: за один шаг индексы i и j, увеличиваются не более чем на 1.

3. Предельность – последовательность начинается в (1,1) и заканчивается в (I,J).

Практическая реализация данного алгоритма представляет собой нахождение значения нормированного расстояния

$$D(A,B) = \frac{1}{N} g_K(c(K)) \tag{13}$$

где $g_k(c(k))$ можно найти из уравнения:

$$g_k(c(k)) = \min_{c(k-1)} [g_{k-1}(c(k-1)) + d(c(k)) \cdot w(k)] \tag{14}$$

Начальное условие:

$$g_1(c(1)) = d(c(1)) \cdot w(1) \tag{15}$$

Из ограничений следует, что $c(1) = (1,1)$,

$$g(i,j) = \min \begin{bmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{bmatrix}. \tag{16}$$

В данном случае нормированное расстояние можно записать как

$$D(A,B) = \frac{1}{K} g(I,J), \text{где K - размерность вектора } c. \tag{17}$$

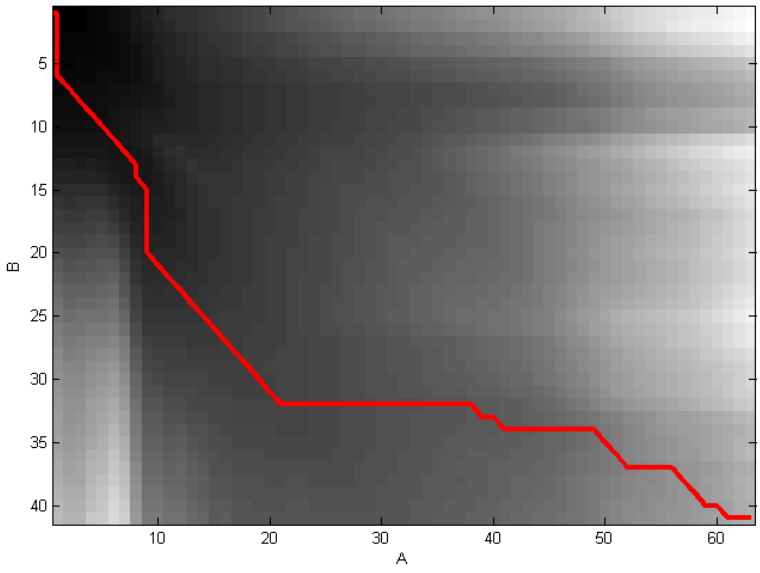

Рис. 4: Пример нахождения функции деформации

Эксперименты проводились для словаря из 10 слов (цифры от 0 до 9). Одним диктором было записано 100 повторений для сравнения (по 10 для каждого в словаре).

С помощью данного алгоритма производится сравнение анализируемого сигнала с сохраненными в памяти компьютера эталонами. В результате выбирается пара с минимальной дистанцией и делается вывод о соответствии сигнала слову из словаря. Результат сравнения слова «четыре» со словарем можно видеть на рис. 5 (Меньшее значение дистанции означает большее сходство).

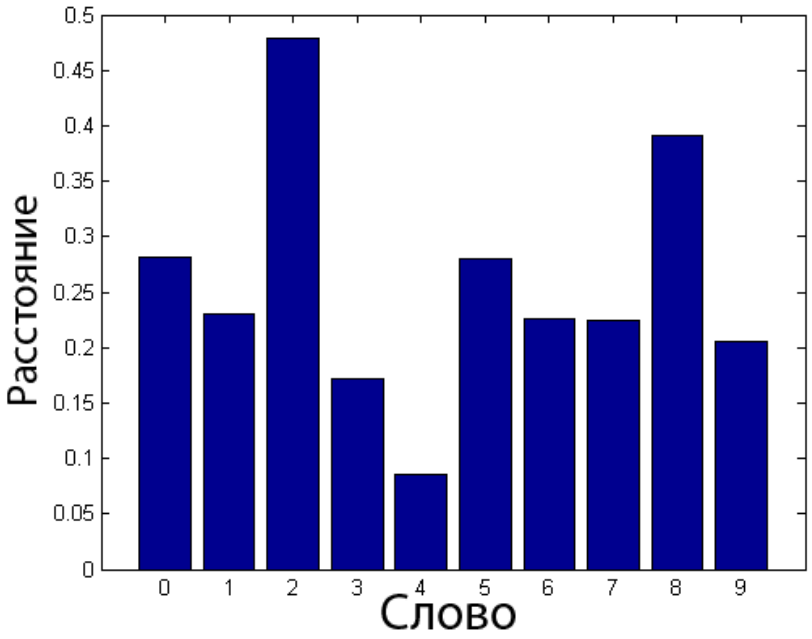

Рис. 5: Диаграмма расстояний для слова «четыре»

В результате, на 100 повторений было обнаружено 2 ошибки распознавания, что позволяет говорить о достаточной точности выбранного алгоритма. На рис. 6 показана диаграмма расстояний для ошибочно распознанного слова «восемь»

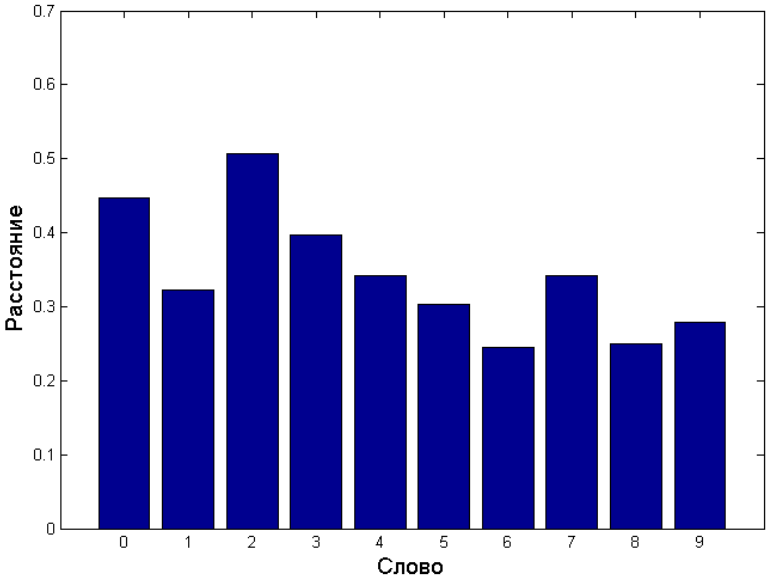

Рис. 6: Диаграмма расстояний для ошибочно распознанного слова «восемь»

Список использованных источников

[1] Винцюк Т.К. Анализ, распространение и интерпретация речевых сигналов. Киев: Наукова думка, 1987.

[2] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, Spoken Language Processing: A Guide to Theory, Algorithm, and System Development, Prentice Hall, 2001, ISBN:0130226165

[3] H. Sakoe and S. Chiba, «Dynamic programming optimization for spoken word recognition», IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-26, No. 1, Feb. 1978

[4] Мазуренко И.Л. Компьютерные системы распознавания речи. Интеллектуальные системы, Москва, 1998 г.

[5] П. Линдсей, Д. Норман Переработка информации у человека — М: Мир, 1974