

РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПРОГРАММЫ ДЛЯ ЗВУКОВОЙ НАВИГАЦИИ ПО ВЕБ-САЙТУ

В. А. Столер, канд. техн. наук, доцент,
К. А. Гурин, магистрант

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Ключевые слова: обработка звука, дефекты произношения слов, интерфейс программы, алгоритмы обработки.

Аннотация. Предложен алгоритм построения пользовательского интерфейса программы для распознавания дефектов слов при их произношении, используемой в устройствах звукового управления, например веб-сайтом. Рассмотрены способы оптимизации программы и пути повышения точности распознавания речи.

К настоящему времени произошло заметное увеличение способов взаимодействия пользователя с интерфейсом программ и устройств. В частности, большую популярность набирает автоматическое распознавание речи (ASR), используемое в звуковом управлении устройствами. Известно множество различных программ распознавания речи, однако большинство из них являются коммерческими. Поскольку коммерческие распознаватели речи доступны для четко определенных приложений, таких как произношение или транскрипция, многие проблемы ASR, такие как распознавание в шумной среде, низкое качество записи и распознавание при дефектах речи, еще предстоит эффективно решить [1].

В данной статье предложен алгоритм построения программы для распознавания ключевых слов, используемых при голосовом управлении, содержащих дефекты произносимой речи. В терминах цифровой обработки сигналов процесс шумоочистки представляет собой преобразование входного сигнала, содержащего как полезный сигнал – речь, так и аддитивный сигнал-помеха – шум, в выходной сигнал, содержащий только речь. Поскольку создание систем, в точности удовлетворяющих данному условию, невозможно, задачу очистки сигнала от шума упрощают [2].

На первом этапе зададим погрешность для шума, оставшегося после обработки. Определим локальные максимумы графика. Интервал M_i , на котором график выходит за пределы заданного значения $\Delta_{ш}$, имеет локальный максимум и ограничен локальными минимумами, является буквой в слове (рисунок 1).

На втором этапе сопоставляем эталонный график и график записанной речи. Выберем количество точек N для сравнения на заданном интервале M_i и зададим погрешность для точек, в пределах сравниваемых интервалов графиков, $\Delta_{т}$. При этом эталонными считаются графики слов, записанные на профессиональном оборудовании и содержащие минимальное количество шумов. Данные условия необходимы для минимизации числа ошибок при сравнении слов.

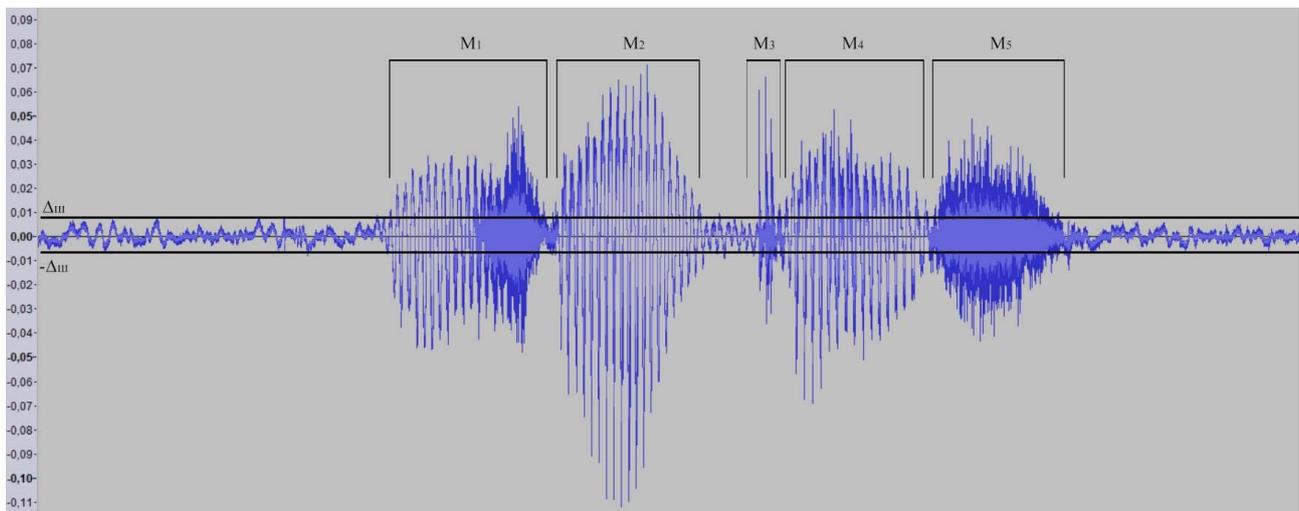


Рисунок 1 – Выбор интервалов M_i на графике записанной речи

При равенстве точек $[x_j; y_j]$ графика записанной речи точкам $[x_k; y_k]$ эталонного графика с учетом выбранной погрешности Δ_T можно говорить о равенстве произнесенных букв.

Из равенства всех интервалов M_i сравниваемых звуковых сигналов следует равенство произнесенных слов.

В результате, обобщенный алгоритм программы для распознавания ключевых слов в записанной речи можно представить следующим образом (рисунок 2).

Для оптимизации алгоритма необходимо нормализовать входной сигнал, т. е. задать верхнюю границу значения y . Таким образом, если разница между минимальным (максимальным) значением y первого графика и минимальным (максимальным) значением y второго графика больше заданной погрешности точки Δ_T , можно говорить о неравенстве символов, т.е. о неравенстве слов [3].

Для корректного определения M_i необходимо учесть, что значение y локального максимума символа много больше заданного значения погрешности шума $\Delta_{ш}$.

Изменение значений погрешности шума $\Delta_{ш}$ и погрешности точки Δ_T позволяет задать необходимую точность сравнения.

При разработке алгоритма необходимо учесть следующие возможные ситуации: количество символов M в записанном слове больше, чем у эталонного; интервал M в записанном слове вмещает в себя большее количество точек N , чем эталон.

Большее число символов M в записанном слове может означать не только более длинное слово, но и дефект произношения (например, «Привет»). Для обработки данной ситуации необходимо предусмотреть проверку, в которой у записанного слова будут попарно сравниваться соседние интервалы M_i и M_{i+1} . Если соседние интервалы равны между собой, значит M_i необходимо исключить из сравнения.

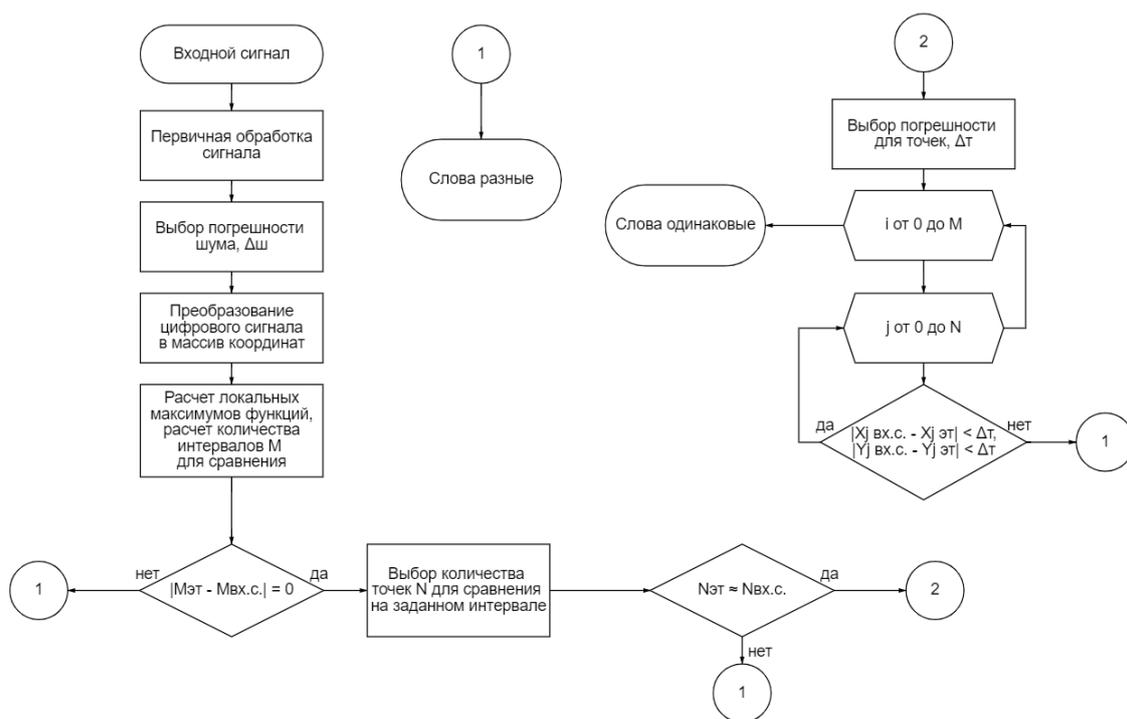


Рисунок 2 – Алгоритм программы для распознавания ключевых слов

В заключении необходимо обратить внимание на такую ситуацию, при которой длина интервала M_i в записанном слове больше длины соответствующего интервала M_i в эталонном слове, т.е. количество вмещаемых интервалом точек N различно. Такое различие может возникнуть не только при неравенстве символов, но и при более длинном произношении буквы (например, «Ммама»). При систематическом повторении значений y в пределах рассматриваемого интервала, можно говорить о более длинном произношении символа и исключить часть интервала из сравнения для уравнивания значений N .

В результате исследований был выполнен анализ основных способов взаимодействия пользователя с интерфейсами, используемых в программах распознавания речи. Предложен алгоритм обработки звука, учитывающий дефекты произношения слов, для построения компьютерной программы распознавания речи, используемой при звуковом управлении электронными устройствами. Рассмотрены способы оптимизации программы и повышения точности распознавания речи.

Список литературы

1. **Benesty, J.** Springer Handbook of Speech Processing / Jacob Benesty, M. Mohan Sondhi, Yiteng Arden Huang. – Springer-Verlag Berlin Heidelberg, 2008. – 1161 p.
2. **Vishnyakov, I. E.** Methods and algorithms for real time voice noise cleaning / I. E. Vishnyakov, M. M. Masyagin, O. A. Odintsov, V. V. Sliusar // Proc. Univ. Electronics, 2021. – vol. 26. – no. 2. – pp. 184–196.
3. **Padmanabhan, J.** Machine Learning in Automatic Speech Recognition: A Survey / Jayashree Padmanabhan, Melvin Jose, Johnson Premkumar // Proc. IETE Technical Review, 2015. – vol. 32. – no. 5. – pp. 240–251.