1998년도 한국음향학회 학술발표대회 논문집 제 17 권 제 2(s) 호

멜켑스트럼의 성능 향상을 위한 critical band 필터의 최적화

현동훈, 이철희
연세대학교 전자공학과

Optimization of Critical Band Filter for Improving Performance of Mel-cepstrum

Donghoon Hyun and Chulhee Lee
Department of Electronic Engineering, Yonsei University
E-mail: chulhee@bubble.yonsei.ac.kr

요 약

현재 음성 인식에서 널리 사용되고 있는 피춰 중의 하나로 멜켑스트럼을 들 수 있다. 멜켑스트럼은 인간의 청각 특성을 적용한 critical band 필터를 사용하여 구하는데, 필터의 형태를 다양하게 적용하여 같은 음성에 대해서 여러 가지의 멜켑스트럼을 구할 수 있다. 본 논문에서는 critical band 필터의 형태, 즉 필터의 모양, 인접한 필터간의 중심 주파수 간격, 그리고 필터의 대역폭을 각각 변화시키면서 멜켑스트럼을 구하여 음성 인식 성능에 미치는 영향을 분석하였다. 또한 최적의 인식 성능을 나타내는 멜켑스트럼을 구하기 위하여 simplex 기법을 사용하여 필터를 최적화하는 방법을 제안한다. DTW (dynamic time warping)를 인식 알고리즘으로 사용하였고 한국어 숫자음을 사용하여 인식 실험을 수행한 결과, 제안된 방법으로 최적화된 필터를 사용하여 구한 멜켑스트럼은 기존의 critical band 필터를 사용하는 것보다 향상된 인식 성능을 나타내었다.

1. 서 론

일반적으로 음성의 피춰는 음성 신호를 수십 ms의 일정한 프레임으로 나누어 각 프레임에서 추출하게 되는데, 현재 널리 사용되고 있는 피춰로 켑스트럼이 있다. 켑스트럼은 음성 신호로부터 성도에 대한 정보를 추출한 것인데, LPC 켑스트럼과 멜켑스트럼 등이 있다. LPC 켑스트럼은 LPC 계수들로부터 얻을 수 있고, 멜켑스트럼은 그림 1과 같은 과정을 통해서 구하게 된다 [1].

멜켑스트럼은 critical band 필터를 사용하여 구하는데, Davis와 Mermelstein은 그림 2와 같은 critical band 필터를 사용하여 구한 멜켑스트럼이 다른 피춰들보다 우수한 성능을 나타내는 것을 보여주었다 [2]. 그림 2에서 인접한 필터간의 중심 간격과 대역폭은 각각 mel 단위의 일정한 값을 가지는데, 이 값들을 다르게 적용하여 멜켑스트럼을 구하고 이에 따른 인식 성능을 분석할 수 있다.

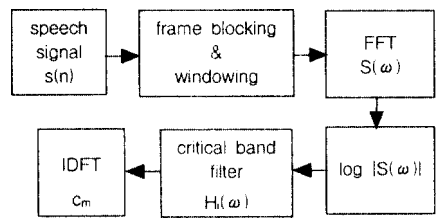


그림 1. 음성 신호로부터 멜켑스트럼을 구하는 과정.

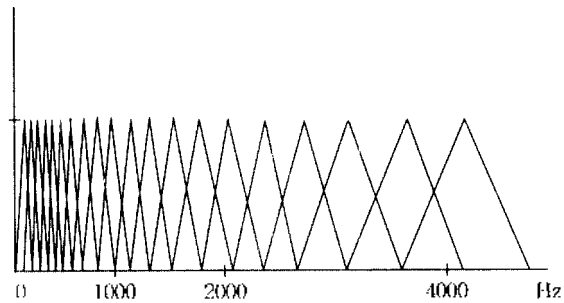


그림 2. 멜켑스트럼을 구하기 위한 critical band 필터들.

본 논문에서는 숫자음을 인식 대상으로 하고 DTW를 인식 알고리즘으로 사용하여 그림 2와 같은 critical band 필터의 중심 주파수, 대역폭, 모양 등을 변화시키며 구한 멜켑스트럼의 성능을 분석하여 최적화 기법을 제안한다. 본 논문의 구성은 다음과 같다.

2절에서 최적화 방법으로 사용한 simplex 방법에 대해 설명하고, 3절에서 simplex 방법을 이용한 멜켑스트럼의 최적화 알고리즘을 제시한다. 4절에서는 한국어 숫자음에 대한 인식 실험 결과를 보여주고 5절에서 결론을 제시한다.

2. 최적화 방법

주어진 함수의 최대값 또는 최소값을 수식적인 과정으로 구할 수 없는 경우에는 최적화 방법을 통해서 해결할 수 있다. 최적화 방법에는 여러 가지가 있으나, 함

수의 gradient를 이용하지 않는 경우에는 direct search 방법을 사용하며, 이러한 방법에는 Nelder와 Mead가 제안한 simplex 방법이 있다 [3]. 이때 simplex는 N차원 공간에서 $N+1$개의 점으로 구성되는 다면체를 의미한다.

최대값을 찾는 simplex 방법은 다음과 같이 설명할 수 있다. 입력이 N차원 벡터 $\boldsymbol{v}$이고 출력이 k인 함수 f를 생각할 때, 먼저 N차원 공간에 초기 simplex를 설정한다. 초기 simplex를 일련의 과정에 따라 변화, 이동시키고, 이러한 과정을 반복함으로써 함수 f의 최대값을 가지는 영역으로 이동하게 된다. 그림 3은 함수 f의 입력이 2차원 벡터 $\boldsymbol{v}=[x\ y]$인 경우 simplex의 변형, 이동을 나타낸다.

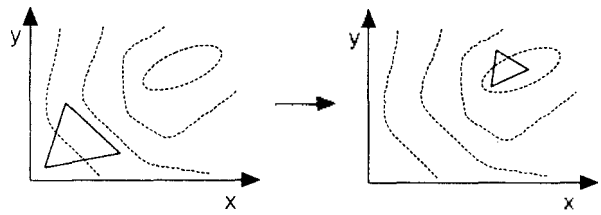


그림 3. $f(\boldsymbol{v})$의 최대값을 찾기 위한 simplex의 변형, 이동.

이때 N차원 공간에서 $N+1$개의 점으로 구성되는 초기 simplex는 식 (1)에 의해서 설정한다.

$$\boldsymbol{v}_i = \boldsymbol{v}_0 + \alpha_i \boldsymbol{e}_i, \quad i=1,\cdots,N \tag{1}$$

$$\textit{where} \begin{cases} \boldsymbol{e}_1=[1\ 0\ 0\ \cdots\ 0]^T \\ \boldsymbol{e}_2=[0\ 1\ 0\ \cdots\ 0]^T \\ \quad\vdots \\ \boldsymbol{e}_N=[0\ 0\ 0\ \cdots\ 1]^T \end{cases}$$

여기서 $\boldsymbol{v}_0$는 N차원에서 임의로 잡은 벡터, $\boldsymbol{e}_i$는 N차원의 단위 벡터, 그리고 α_i는 simplex의 각 모서리의 길이를 결정하는 상수이다.

이와 같이 초기 simplex를 설정한 뒤, $N+1$개의 점들에서 함수 f의 값을 계산하여 최대값과 최소값을 각각 f_G, f_S로 표시하고, 그에 해당되는 점을 각각 $\boldsymbol{G}$, $\boldsymbol{S}$라 하자. 이때 $\boldsymbol{S}$를 제외한 나머지 점들이 구성하는 다면체의 중심 $\boldsymbol{X}$를 다음 식에 의해 구할 수 있다.

$$\boldsymbol{X}=\frac{1}{N}\left(\sum_{i=0}^{N}\boldsymbol{v}_i - \boldsymbol{S}\right) \tag{2}$$

여기서 $\boldsymbol{X}$는 $\boldsymbol{S}$를 이동하기 위한 기준점으로 생각할 수 있다.

초기 simplex를 구성하는 각 점으로부터 함수값을 구한 뒤, 이 값들을 이용하여 simplex를 변형시키는데, 변형 방법에는 크게 두 가지가 있다.

첫 번째 방법은 simplex를 구성하는 $N+1$개의 점 중에서 1개의 점을 이동하는 것인데, 여기에는 reflection, expansion, contraction 등의 단계가 있다. 각각의 단계에 해당되는 점을 각각 $\boldsymbol{R}$, $\boldsymbol{E}$, $\boldsymbol{C}$라 하면 아래와 같은 식으로 구할 수 있다.

$$\begin{aligned} \boldsymbol{R} &= \boldsymbol{X} + (\boldsymbol{X}-\boldsymbol{S}), & f_R &= f(\boldsymbol{R}) \\ \boldsymbol{E} &= \boldsymbol{R} + (\boldsymbol{X}-\boldsymbol{S}), & f_E &= f(\boldsymbol{E}) \\ \boldsymbol{C} &= \boldsymbol{X} + \tfrac{1}{2}(\boldsymbol{S}-\boldsymbol{X}), & f_C &= f(\boldsymbol{C}) \end{aligned} \tag{3}$$

여기서 f_R, f_E, f_C는 각 점에 해당하는 함수 값이다.

이때 이동된 점에서의 함수값 f_R, f_E, f_C를 f_G, f_S와 비교하여 그림 4에 제시된 몇몇 조건들에 의하여 이동된 점으로 새로운 simplex를 형성한다.

다른 방법은 $\boldsymbol{G}$를 제외한 모든 점을 이동하는 것인데, 이를 multiple contraction이라 한다. 이것은 아래 식에서 구한 $N+1$개의 점으로 simplex를 변형시키는 것이다.

$$\boldsymbol{v}_i=\frac{\boldsymbol{v}_i+\boldsymbol{G}}{2}, \quad i=0,\cdots,N \tag{4}$$

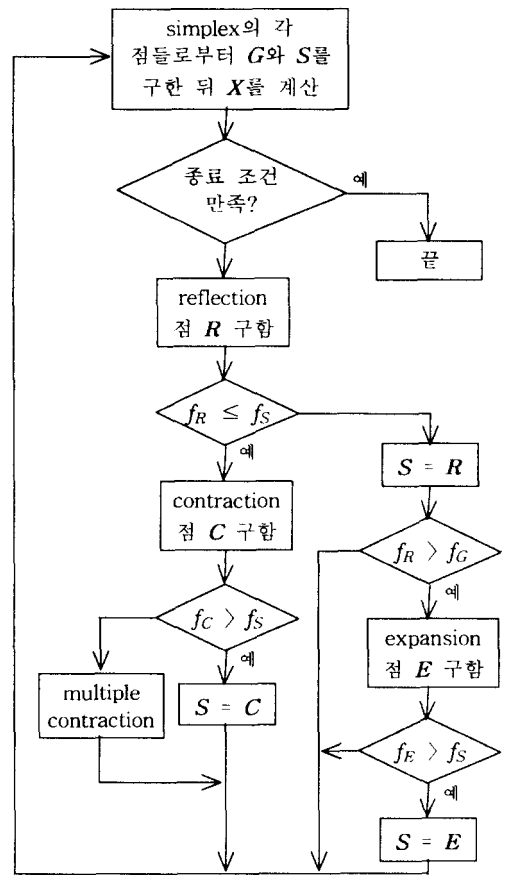


그림 4. 최대값을 찾기 위한 simplex 방법의 흐름도.

위와 같은 simplex의 변형 단계들을 반복함으로써 함수 f의 최대값을 가지는 N차원 영역으로 simplex가 이동하게 된다. Simplex의 변형을 중단하는 종료 조건으로 아래와 같은 식을 사용하여 simplex의 각 모서리 길이가 미리 정해 놓은 값 ε보다 작을 때까지 최적화 과정을 반복한다.

$$\max \left| \boldsymbol{v}_i - \boldsymbol{v}_j \right| < \varepsilon \qquad i, j = 1, \cdots, N \qquad (5)$$

지금까지 설명한 simplex 방법을 간략한 흐름도로 나타내면 그림 4와 같다.

3. 멜캡스트럼의 최적화

본 논문에서는 함수 f가 critical band 필터의 중심 주파수와 대역폭을 입력으로 하여 멜캡스트럼을 구하고 인식을 수행하여 최종적으로 인식률을 함수값으로 가진다고 가정한다. 이때 N_f개의 critical band 필터가 존재하는 경우에 $2N_f$개의 입력을 다음과 같이 $2N_f$차원 벡터 $\boldsymbol{v}$의 원소들로 생각하여 2절에서 설명한 simplex 방법을 적용할 수 있다.

$$\boldsymbol{v} = [\, c_1 \ \cdots \ c_{Nf} \ b_1 \ \cdots \ b_{Nf} \,]^T \qquad (6)$$

여기서 c_i, b_i는 각각 i번째 필터의 중심 주파수와 대역폭을 나타낸다.

Simplex 방법을 적용할 때, $2N_f$차원 공간에 설정한 초기 simplex가 최대 인식률을 나타내는 영역에 근접해 있으면 빠르고 정확하게 최대 인식률을 찾을 수 있으므로 초기 simplex의 위치를 설정하는 것에 주의해야 한다.

본 논문에서는 최대 인식률을 나타내는 영역에 근접하는 초기 simplex를 설정하기 위하여 그림 5와 같이 필터의 대역폭(B)과 인접한 필터의 중심 간격(C)과 모양을 변화시킨 여러 개의 critical band 필터를 사용하여 멜캡스트럼을 구하고 이에 따른 인식 결과를 분석하여 우수한 인식 성능을 나타내는 critical band 필터를 선택하였다. 선택된 필터의 중심 주파수와 대역폭을 초기 입력 $\boldsymbol{v}_0$로 설정하여 식 (1)에 의해서 초기 simplex를 구성하였다. 여기서 각 필터의 대역폭과 인접한 필터의 중심 간격은 각각 mel 단위의 일정한 값을 가지며 식 (7)과 같다. 또한 mel 단위와 hz 단위의 대응 관계는 식 (8)과 같다 [4].

$$c_i = c_{i-1} + C, \quad b_i = B \qquad (C, B: \ mel \ unit) \qquad (7)$$

$$F_{mel} = 2595 \log_{10}\left(1 + \frac{F_{Hz}}{700}\right) \qquad (8)$$

4. 실험 결과

본 논문에서는 남자 10명과 여자 10명의 화자가 숫자음 0에서 9까지 각각 10번씩 발음한 것을 인식 대상으로 하였다. 샘플링 주파수를 11.025kHz로 하여 PC에서 음성을 녹음하였고 음성의 시작점과 끝점은 수동으로 검출하였다. 또한 인식 알고리즘으로 DTW를 사용하였다.

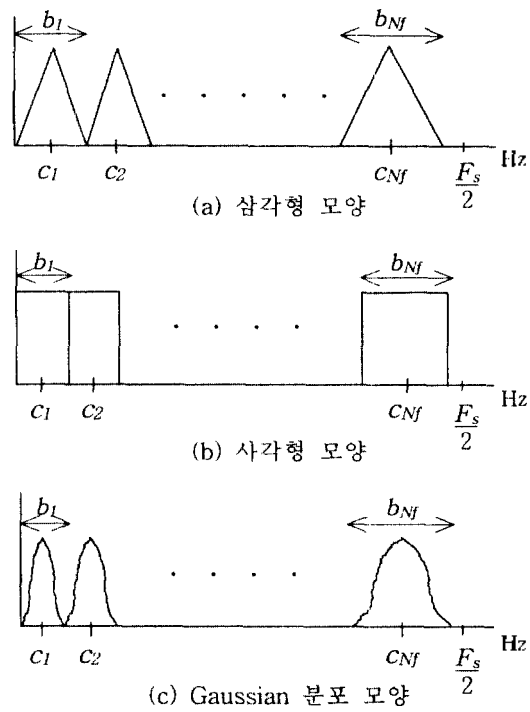


그림 5. 여러 가지 critical band 필터들.

첫 번째 실험은 화자 2명(남자 1명, 여자 1명)이 한번씩 발음한 숫자음을 기준 음성으로 사용하여 최적화를 수행하는 것이다.

3절에서 설명한 바와 같이 초기의 simplex를 설정하기 위해서 mel 단위의 (C, B)를 다음과 같은 10가지 경우로 설정하여 초기 실험을 수행하였다.

(75, 75), (100, 100), (125, 125), (150, 150), (175, 175)
(75, 150), (100, 200), (125, 250), (150, 300), (175, 350)

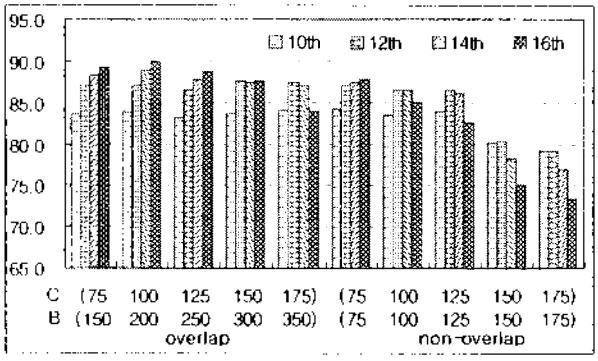
인접한 필터간의 중심 간격과 대역폭을 위의 (C, B)로 사용하고, 필터의 모양을 삼각형, 사각형, Gaussian 분포 모양으로 하였을 때 인식 결과를 그림 6(a), 6(b), 6(c)에 각각 나타내었다. 기준 음성으로 남자 1명, 여자 1명이 한번씩 발음한 음성을 임의로 한 개 선택하였고 10, 12, 14, 16차 멜캡스트럼에 대한 인식률을 구하였다.

그림 6을 보면 사각형 모양의 필터에서 (C, B)가 (75, 75), (100, 100), (125, 125)일 때 인식률이 양호하였다. 따라서 초기의 simplex를 구성하기 위하여 사각형 모양의 필터에서 (C, B)를 (100, 100)으로 하여 초기의 입력 $\boldsymbol{v}_0$를 설정하였다. 샘플링 주파수를 고려하면 24개의 필터가 $F_s/2$ 범위 안에 존재하므로 초기의 입력 $\boldsymbol{v}_0$는 다음과 같이 나타난다.

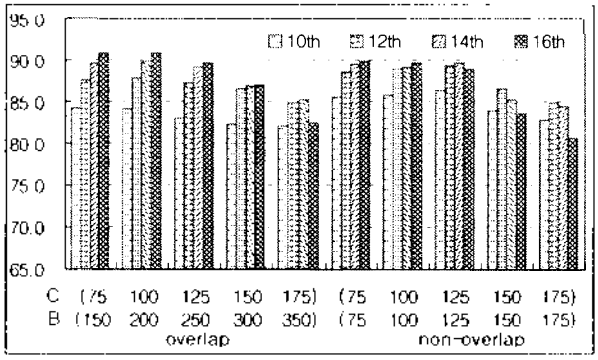
$$\boldsymbol{v}_0 = [\,100 \ \ 200 \ \cdots \ 2400 \ \ 100 \ \ 100 \ \cdots \ \ 100\,]^T$$

이것은 초기 simplex를 구성하는 한 점이 되고 나머지 점들은 아래의 식으로 구한다.

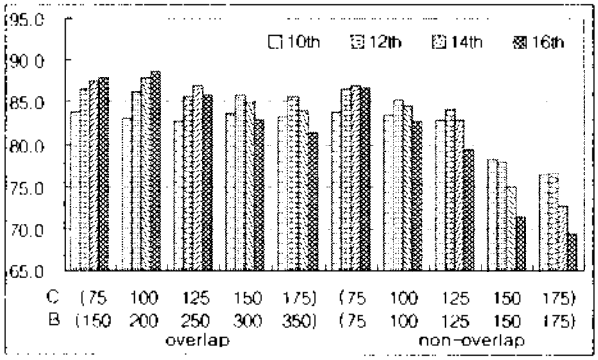
$$\begin{cases} v_i = v_0 + 60e_i, & i = 1, \cdots, N_f \\ v_i = v_0 + 150e_i, & i = N_f + 1, \cdots, 2N_f \end{cases}$$



(a) 삼각형 모양의 필터.



(b) 사각형 모양의 필터.



(c) Gaussin 분포 모양의 필터

그림 6. 여러 가지 (C, B)에 의한 멜캡스트럼의 인식률

이와 같이 초기 simplex를 설정하였고, 종료 조건으로 식 (5)에서 ε을 80으로 정하고 10, 12, 14, 16차의 멜캡스트럼에 대해서 각각 최적화를 수행하였다. (C, B)가 (100, 100)일 때의 인식률과 최적화된 인식률을 비교하면 그림 7과 같다. 그림에서 볼 수 있듯이 4~6%의 성능 향상이 관찰되었다.

두 번째 실험은 첫 번째 실험과 다른 기준 음성에 대해서 최적화된 필터들을 적용할 때, 인식 성능이 어느 정도 향상되는지 분석하는 것이다. 기준화자 2명을 임의로 35회 선택하여 실험을 수행하였고, 인식률의 평균을 그림 8에 나타내었다. 평균적으로 볼 때 3~4%의 성능 향상을 관찰할 수 있었다.

5. 결 론

본 논문에서는 simplex 알고리즘에 의하여 critical band 필터를 최적화하여 멜캡스트럼의 성능 향상을 모색하였다. 사각형 모양의 필터를 사용하여 구한 멜캡스트럼이 우수한 인식 성능을 나타내었으며, 필터들의 중심 주파수와 대역폭을 변화시키면서 멜캡스트럼의 성능 향상이 가능함을 보여 주었다. 기존의 필터와 비교해볼 때, 최적화된 필터는 저주파 성분의 경우 대역폭이 대부분 증가하였고, 고주파 성분의 경우 대역폭이 대부분 감소하는 것을 관찰할 수 있었다.

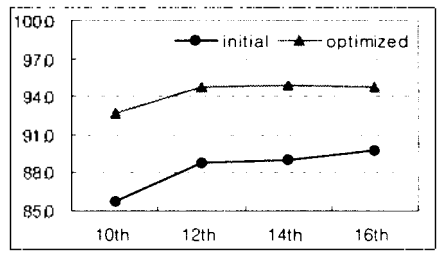


그림 7. (C,B)가 (100, 100)일 때와 최적화된 인식률의 비교. (기준화자 2명).

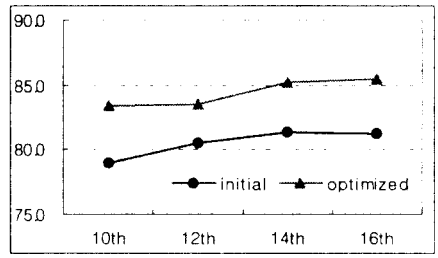


그림 8. (C,B)가 (100, 100)일 때와 최적화된 인식률의 비교. (기준화자 2명을 임의로 35회 선택).

6. 참고 문헌

[1] J. R. Deller J. R. J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Prentice Hall, 1987.

[2] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 357-366, Aug. 1980.

[3] J. L. Buchanan. P. R. Turner, *Numerical Methods and Analysis*, McGraw-Hill, INC., 1992.

[4] D. O'Shaughnessy, *Speech Communication*, Addison-Wesley Publishing Company, 1987.