RELATIVE PITCH TRACKING FOR SINGING VOICE AS APPLICATION IN QUERY BY HUMMING SYSTEMS

G. Ananthakrishnan[1], A.G. Ramakrishnan[2]
Dept. of Electrical Engineering,
Indian Institute of Science, Bangalore – 560012,
India
{ananthg[1], ramkiag[2]}@ee.iisc.ernet.in

ABSTRACT

Pitch extraction from singing voice has been viewed from the absolute pitch of the voice. However, for the query by humming applications, the absolute pitch is not of as much importance as that of relative pitch. This paper addresses the issue of relative pitch tracking for singing voice and attempts to improve the transcription accuracy. It also tries to bring robustness to transcription, in the several forms of querying by a user. The algorithm makes use of specially designed Bach filters which automatically associate a relative frequency with a corresponding musical semitone. This makes conversion of the transcribed pitch values to the MIDI format relatively easy. Preliminary results show up to 85% accuracy in the automated transcription when compared to manually transcription.

KEY WORDS

Musical Transcription, Singing, Pitch-tracking, Bach filters

1. Introduction

‘Musical transcription’ is referred to the process of converting a musical rendering or performance into a set of symbols or notations. The notations may contain information about the pitch, duration, stress and timbre of the musical rendering. But for the context of query by humming only pitch and duration information are considered. So the term musical transcription has been interchangeably used with pitch tracking.

There are several approaches to pitch estimation. [1], [2] and [3] have reviewed the methods in detail. However for singing voices, a reliable conversion of pitch estimates to a symbolic notation has proven to be a challenging problem. This is because a typical vocal rendition contains both in accuracies in pitch and timing. Besides, pitch estimates of voiced regions are only possible and to convert to a symbolic notation, it is necessary to assign suitable pitch values to even the unvoiced regions.

Another important factor to be considered while transcribing hummed queries is that the vocal rendition of a query may have a different absolute pitch as compared to the musical rendition of the original sound track. The query may or may not contain words, may be a whistle.

This paper tries to provide a pitch tracking algorithm, which obtains relative frequencies, and works reasonably well under all the conditions specified. It must be noted that the output of the automated transcription method is compared with manually transcribed MIDI files. i.e. Transcription is done for the actual sample of singing voice (or whistle). So the manually transcribed MIDI files contain many notations of ‘glide’ or ‘pitch-bends’ where the pitch changes from one semi-tone to the other, without it being referred to as a new note. This is a different approach to the usual trend of comparing the automated transcription result with any MIDI file of the same track. Thus the paper tries to mimic the human transcription process to some extent.

2. Theory

2.1 Bach filter-bank

The inspiration for the ‘Bach’ scale is obtained from music. In music, every octave contains 12 semi-tones. Each of the semitones is related to the next one by roughly a ratio of $2^{(1/12)}$. This ratio was discovered by the great musician of the 18th century, J.S. Bach [4, 5]. This number of $2^{(1/12)}$ holds true for almost all genres of music and relates to some natural perceptual phenomenon.

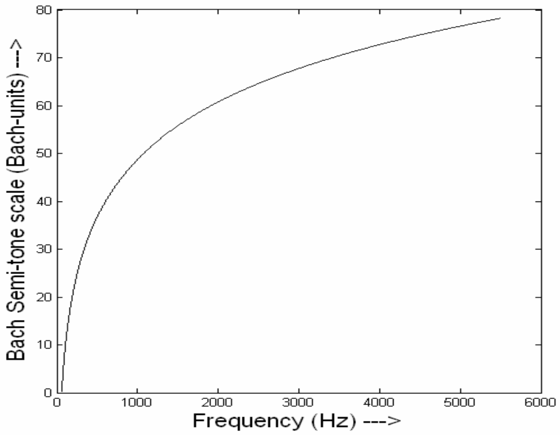

Fig 1: - The Bach scale

For ‘Bach’ scale

$$\mathrm{b(f)} = 12^*\log_2\left(\frac{\mathrm{f}}{\mathrm{base}}\right) \qquad (1)$$

The Bach scale is only a relative scale and depends on the ‘base’ frequency.

Assuming 12 filters per octave corresponding to 12 semitones in the Bach scale, the maximum number of filters ‘M’ is calculated by.

$$M \ = \ 12\log_2\left(\frac{F_s}{2*base}\right) \qquad (2)$$

where ‘F_s’ is the sampling frequency. The band-width formulation is given by using equation (1) for b(f).

$$B_{bach}(f) \ = \ base*\left(2^{(b(f)+1)/12} \ - \ 2^{(b(f)-1)/12}\right) \qquad (3)$$

The number of filter coefficients used to generate the ‘nth’ filter is determined by

$$N(n) = 2*ceil(1/\ f_b(n)) \qquad (4)$$

We thus see that the time resolution is poor for lower frequencies but better for higher frequencies. So we get the paradoxical ability to get better time resolution for higher frequencies and better frequency resolution for lower frequencies.

The filters designed are lag-windows obtained by the standard Blackman-Tukey spectral estimation method [6]. The set of filter coefficients obtained, is the eigenvector associated with the maximum Eigen value of the matrix with elements

$$\gamma_{m,n} = \beta * signum((m-n)*\beta*\Pi) \qquad (5)$$

where 2*β is the band-width in radians/sec and

$$signum(x) = \sin(x)/x \qquad \{x\,!=0\} \qquad (6)$$

$$=1 \qquad \{x=0\}$$

The filter coefficients are real, symmetric and finite, so the phase responses are linear.

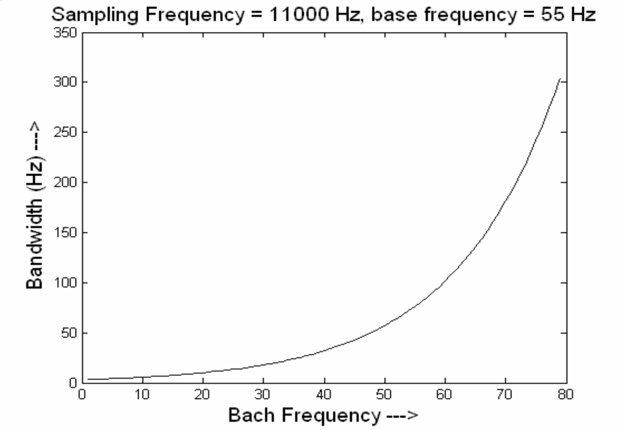

Fig. 2 – Bandwidth of the windows as against the centre frequency in Bach

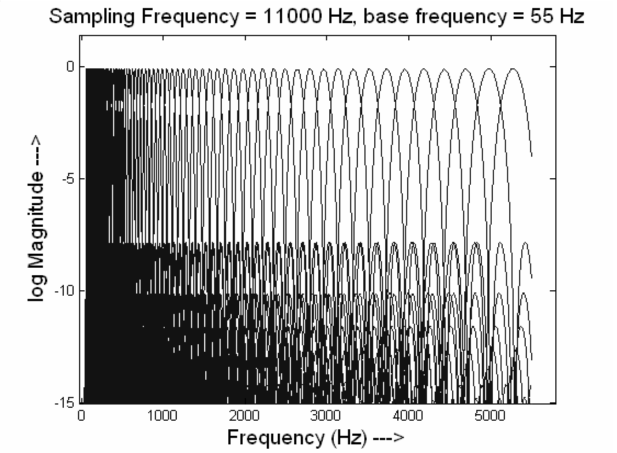

Fig. 3 – The Bach scale Filter-bank

2.2. Time varying representation

The most common method of analyzing the time-varying voice signal has been by treating it as short-time stationary. However, this correspondence considers the signal as time varying. The voice signal is filtered by a bank of ‘M’ band-pass filters each shifted in frequency by a fixed factor. So we have ‘M’ filtered versions of the same voice signal. Consider the ‘nth’ such version of the signal. The energy around the ‘nth’ frequency component of the signal around a time instant ‘k’ will be equal to the ‘kth’ output energy of the ‘nth’ filter-bank.

$$F_k(n) = F_n(k) = 20*\log_{10}\left(abs\left(h_n(k)\otimes s(k)\right)\right)$$

$$\text{(db SPL)} \qquad (7)$$

where s(k) is the input voice signal, $h_n(k)$ is the band-pass filter designed around centre frequency ‘n’. The $\otimes$ symbol represents linear filtering. The feature vectors $|F_k(n)|_{k=1:T}$ or $|F_n(k)|_{n=1:M}$ are the two ways of the 2-D representation of the signal *s(k)*. ‘*T*’ is the total number of samples in the signal.

The first filter is centered around a ‘base’ frequency (any base freq between 50 to 80 Hz results in a good performance). The filter-bank is only an analysis filter-bank and not a perfect reconstruction one.

2.3. Frequency masking

The Frequency energy vector at every time instant contains the relative energies around the particular time instant. However in the presence of a dominant frequency, the other frequencies at that time instant are masked. This masking is associated with critical bandwidth, so the ‘Bach’ frequencies need to be converted to the Equivalent Critical Bandwidth frequencies (‘Bark’ scale)

$$\mathrm{f_{CB}} = 13\tan^{-1}\left(0.76*\mathrm{base}*2^{\frac{\mathrm{f_{bach}}}{12}}\right) + 3.5\tan^{-1}\left(\frac{\mathrm{base}*2^{\frac{\mathrm{f_{bach}}}{12}}}{7.5}\right)^2 \qquad (8)$$

The approximation for conversion from linear frequency to Critical Bandwidth is obtained from [7].

The presence of a tone is detected by the following criteria.

$$F_n(k) = \max(F_n((k+K):(k-K))) \text{ and}$$
$$F_n(k) = 7db + \min(F_n((k+K):(k-K))) \qquad (9)$$

Where,

$$K = 2, \qquad \log_2 \frac{0.17}{base} \le k \le \log_2 \frac{5.5*10^3}{base}$$
$$K = 3, \qquad \log_2 \frac{5.5*10^3}{base} \le k \le \log_2 \frac{11*10^3}{base} \qquad (10)$$
$$K = 6, \qquad \log_2 \frac{11*10^3}{base} \le k \le \log_2 \frac{20*10^3}{base}$$

The spreading function (SF) [8] depends on the maskee location i, the masker location j, the power spectrum F_n at j, and the difference between the masker and maskee locations in Barks (dz=f_{CB}(i)-f_{CB}(j)) is given by (11)

$$SF(i,j) = \begin{cases} 17dz - 0.4*F_n(j)+11 & \text{For } -3 <= dz < -1 \\ (0.4*F_n(j)+6)dz & \text{For } -1 <= dz < 0 \\ -17dz & \text{For } 0 <= dz < 1 \\ (0.15*F_n(j)-17)*dz - 0.15*F_n(j) & \text{For } 1 <= dz < 8 \end{cases} \qquad (11)$$

The tone masking noise [8]

T_{MN}(i,j) = F_n(j) - 0.175*f_{CB}(j) + SF(i,j) - 2.025 (dB SPL)

For tone masking tone [8]

T_{MT}(i,j) = F_n(j) - 0.275*f_{CB}(j) + SF(i,j) - 6.025 (dB SPL) (12)

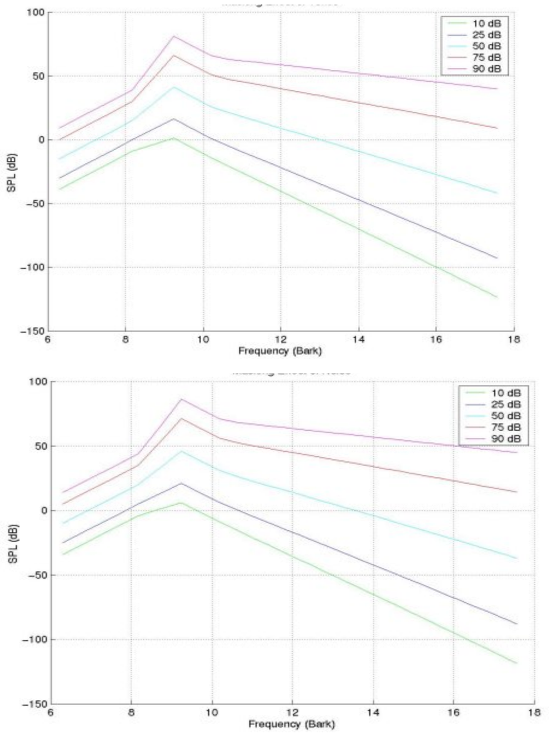

Fig 4: - (a) Tone masking Tone, (b)Tone masking noise

3. Algorithm

The algorithm contains the basic blocks as shown in figure 5.

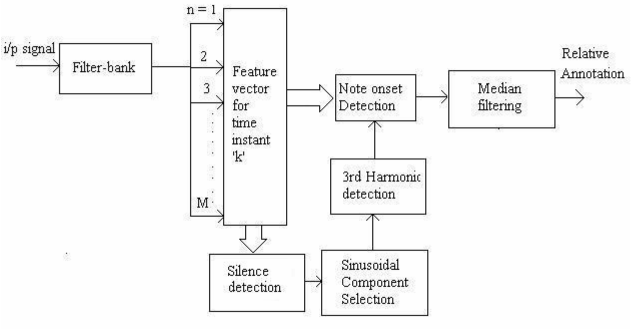

Fig 5- Block diagram of the algorithm

As seen from section 2, the filter-bank designed has a higher frequency resolution for lower frequencies and higher time resolution for higher frequencies. The fundamental frequency is usually in the range of 100 to 300 Hz, especially for males. A band-pass filter designed to track the fundamental frequency needs a high number of filter coefficients and is sluggish in nature (poor time resolution). A filter designed in the range of the 3rd harmonic usually has the optimum compromise between time and frequency resolution.

The 3rd harmonic, though not always, is usually one of the significant, harmonic peaks, which makes it another good reason to track the same.

3.1. Silence detection

It is assumed that the background noise is of non-sinusoidal nature. So the power spectrum will not be spiky and thus will have a low variance. This aspect is used to roughly determine the difference between voiced and unvoiced regions. So if the variance is below a certain threshold, then it is considered as silence. This threshold is related to the Signal to noise ration (SNR) of the signal.

$$\sigma^2(k) = \frac{1}{M-1}\sum_{n=1}^{M}(F_n(k) - \overline{F}(k))^2$$
$$\overline{F}(k) = \sum_{n=1}^{M}\frac{F_n(k)}{M} \qquad (13)$$

If $20\log_{10}(\max(\sigma^2)) - 20\log_{10}(\sigma^2(k)) > -SNR$, then s(k) is considered a region of silence. The voiced part of the sample may be interspersed with several unvoiced parts. But due to the sluggish nature of the lower frequency filters, the variance does not drop below the threshold. So this method is fairly successful.

3.2. Sinusoidal component detection

The threshold for hearing a sinusoid is defined from (12). The thresholds are calculated using (11) and (12) for frequencies in the surrounding. Sinusoidal component is searched for, using (9). If a particular frequency is sinusoidal, then masking threshold is recalculated. If greater than the previous threshold, then is taken as the new threshold. For time instant 'k' the initial masker *I(k)* is taken as the frequency with the highest amplitude $F_I(k)$.

$$I(k) = \underset{n=1:M}{\arg\max}\left(F_n(k)\right)$$
$$F_I(k) = \max_{n=1:M}\left(F_n(k)\right) \qquad (14)$$

```
$ind = 1$   and   $j = I(k)$
for $i = j-1:-1:1$ and $j+1:M$
    $T_{MN}(i,j) = F_j(k) - 0.175 f_{CB}(j) + SF(i,j) - 2.025$
    $T_{MT}(i,j) = F_j(k) - 0.275 f_{CB}(j) + SF(i,j) - 6.025$
    if $F_i(k) > T_{MT}(i,j)$
      $S_k(ind) = i$      $FS_k(ind) = F_i(k)$
      $ind{+}{+}$                                        (15)
      if $F_i(k) > T_{MN}(i,j)$
          $j = i$
```

Where S_k is the list of significant sinusoids at time instant 'k' and FS_k are their corresponding amplitudes.

3.3. Harmonically related sinusoidal set

The significant sinusoids detected have to be grouped according to their possible harmonics. We are taking into consideration 7 significant harmonics, which are usually the significant ones in case of human voice. Due to the non-linear nature of the Bach frequency scale, the harmonic values are related to the fundamental frequency 'f_0'in the following manner.

2nd harmonic	1st overtone	f_0+12
3rd harmonic	2nd overtone	f_0+19
4th harmonic	3rd overtone	f_0+24
5th harmonic	4th overtone	f_0+27
6th harmonic	5th overtone	f_0+29
7th harmonic	6th overtone	f_0+31

Thus H={0,12,19,24,27,29,31} is the set of harmonic differences. For a group of significant sinusoids, all potential harmonically related groups are identified.

e.g. For a group of S_k = {7, 12, 19, 22, 26, 31, 34, 38, 41} and corresponding FS_k = {0.02, 0.03, 0.05, 0.02, 0.07, 0.06, 0.04, 0.02, 0.03}, the ordering would be

$$G_k = \begin{pmatrix} 7 & 19 & 26 & 31 & 34 & 0 & 0 \\ 12 & 0 & 31 & 0 & 0 & 41 & 0 \\ 19 & 31 & 38 & 0 & 0 & 0 & 0 \\ 22 & 0 & 41 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 38 & 41 & 0 & 0 \end{pmatrix}$$

'L' is the total number of harmonically related groups. In this case it is 5. Here $\mu(k)$ = 26 and $F_\mu(k)$ = 0.07. There are two ways to find the most likely 3rd harmonic

- **Most significant peak:**

  if $G_k(p,q) = \mu(k)$ for $p = 1$ to L, $q = 1$ to 7

  then $\lambda_{auto}(k) = G_k(p,q) - H(q) + 19$

  In this case $\lambda_{auto}(k)$ = 26
- **The harmonically richest set:**

  $$HF_k(p) = \sum_{q=1}^{7} F_k(G_k(p,q))$$
  $$HS_k = \underset{p=1:L}{\arg\max}(HF_k(p)) \qquad (16)$$
  $$\lambda_{auto}(k) = G_k(HS_k, 3)$$

  In this case, too $\lambda_{auto}(k)$ = 26

Thus $\lambda_{auto}(k)$ is the pitch contour estimate found by both methods and usually the two estimates turn out to the same. In case they are different, the estimate closest to $\lambda_{auto}(k\text{-}1)$ is taken as $\lambda_{auto}(k)$.

The $\lambda_{auto}(k)$ function is then median filtered in order to avoid some spikes created during the transitions of the pitch.

3.4 Note onset detection

This is a non-trivial problem, and is usually the source of errors. There are three possible types of note onsets. The first one, associated with a change in frequency and the second associated with a change in energy and the third associated with a change in syllable being pronounced. A change in note may not however always be associated with a note onset, because of the presence of glides in singing voice. A three step approach is used to detect the note onsets.

- **Change in frequency:**

  $\vartheta(k) = 1$ if $\lambda(k) \mathrel{!}= \lambda(k\text{-}1)$ Else $\vartheta(k) = 0$

Thus $\vartheta(k)$ =1 denotes the estimate for note onsets based on change in frequency.

- **Change in spectral variance:**

  $\sigma^2(k)$ is obtained from (13)

  $$\text{if} \quad \sigma^2(k) = \min_{p=k-R:k+R}(\sigma^2(p))$$
  $$\text{then} \quad \xi(k) = 1 \qquad (17)$$
  $$\text{else} \quad \xi(k) = 0$$

Thus ξ(k) =1 denotes the estimate for note onsets based on variations in spectral energy. 'R' is the number of samples equivalent to the fastest note transition. 'R' can be roughly taken as 0.05*F_s

- **Change in normalized spectral variance**

  Normalized Spectral variance (*NSV*) is defined as

  $$NSV(k) = \frac{1}{M-1}\sum_{n=1}^{M}\left(\frac{F_n(k) - \overline{F}(k)}{\overline{F}(k)}\right)^2 \qquad (18)$$

NSV, normalizes the spectral variation with respect to variations in amplitude with respect to time. Thus the plot of NSV as shown in fig 7 is independent of the energy associated with the spectrum, and just depends on the spread of the spectrum. The NSV changes wherever there is a change in spectral content. The first derivative (or difference) is called the *DNSV*(k) = *NSV(k)-NSV(k-1).*

The note onset estimate ζ(k) is obtained from the *DNSV* as follows

$$\text{if} \quad DNVR(k) = \min_{p=k-R:k+R} (DNVR(p))$$

then (19)

$$\text{if } DNVR(k) < \Theta \text{ then } \zeta(k) = 1$$

$$\text{else } \zeta(k) = 0$$

The threshold ‘Θ’ must be calculated empirically and is related to the SNR of the signal. A good approximation would be.

$$\Theta = \frac{\min\limits_{k=1:T} DNVR(k)}{10^{\frac{-SNR}{20}}} \qquad (20)$$

The final decision whether a region ‘Ψ’ around sample ‘k’ contains a note onset is decided by the Boolean logic.

$$p = k - \Psi : k + \Psi$$

$$N_o(k) \; = \; (\overset{\frown}{\vartheta(p)} \,\&\&\, \overset{\frown}{\zeta(p)}) \;\|\; (\overset{\frown}{\zeta(p)} \,\&\&\, \overset{\frown}{\xi(p)}) \qquad (21)$$

Where for vector $|A|_{1:p}$

$$\overset{\frown}{A} = 1, \qquad \text{if } \sum_{i=1}^{p} A(i) > 1 \quad \text{else } \overset{\frown}{A} = 0$$

The region ‘Ψ’ is selected to be the allowed tolerance for note onset timing. In this case we take it to be 30 ms.Thus Ψ can be calculated to be 0.03 *F_S

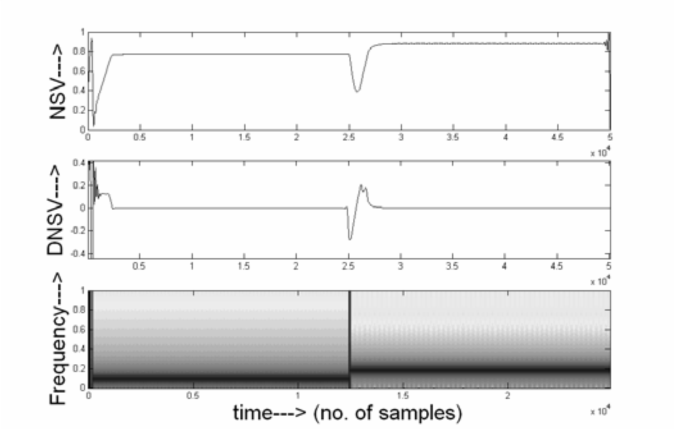

Fig 6- The plot of (a) NSV, (b) DNSV and (c) The spectrogram of an artificial sample, having two sinusoids
(F_s=11000Hz)

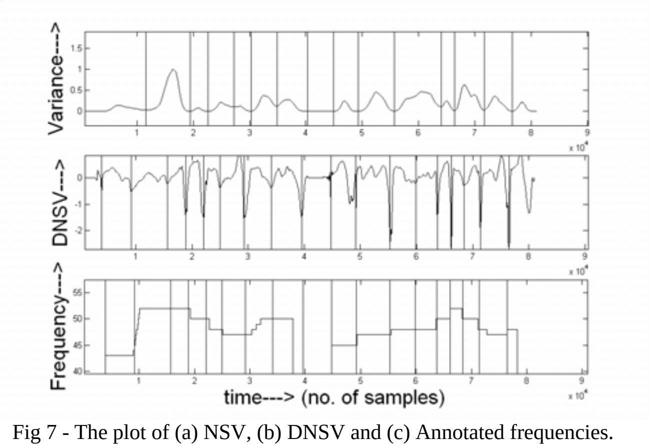

Fig 7 - The plot of (a) NSV, (b) DNSV and (c) Annotated frequencies. The vertical lines indicate the segmentation of the samples
(F_s=11000Hz)

Between two consecutive note onsets, $N_o(k_1)$ and $N_O(k_2)$ if for at least one $k_1 < k < k_2$, ϑ *(k)* =1, then, the region is considered to be a glide or a pitch-bend.

Thus the Pitch contour is converted into a set of 3 variables: $\Gamma_{auto}(i) = \{\tau_{start}, \lambda_{start}, \lambda_{end}\}$ where τ_{start} represents the note onset time (sec) λ_{start} represents the starting relative frequency of a note, and λ_{end} represents the ending relative frequency of the note. In case of glides λ_{start} and λ_{end} will be different, and in case of regular notes, λ_{start} and λ_{end} will be the same.

The comparison between two relative pitch contours can only be done by shifting the transcription by ‘*S*’ number of semitones before comparing it. The automated transcription is referred to as Γ_{auto} and the original manual transcription to MIDI format is represented as Γ_{midi}

$$\mu_{auto} = \frac{1}{I_{auto}} \sum_i \Gamma_{auto}(i,2)$$

$$\mu_{midi} = \frac{1}{I_{midi}} \sum_i \Gamma_{midi}(i,2) \qquad (22)$$

where I is the total number of notes in the sample.

Then for i = 1 to I_{auto} (23)

$$\Gamma_{auto}(i,2) = \Gamma_{auto}(i,2) + \mu_{midi} - \mu_{auto}$$

$$\Gamma_{auto}(i,3) = \Gamma_{auto}(i,3) + \mu_{midi} - \mu_{auto}$$

Thus $S = \mu_{midi} - \mu_{auto}$

4. Results and Analysis

The algorithm was tested for 20 samples of both males and female voices. The samples included singing with words, humming with /m/ and /n/ (nasal) phones, and whistling. The recording was done with a SNR of -25 dB and sampling rate (F_s) of 11 KHz. The transcription into MIDI was done by trained musicians for the samples.

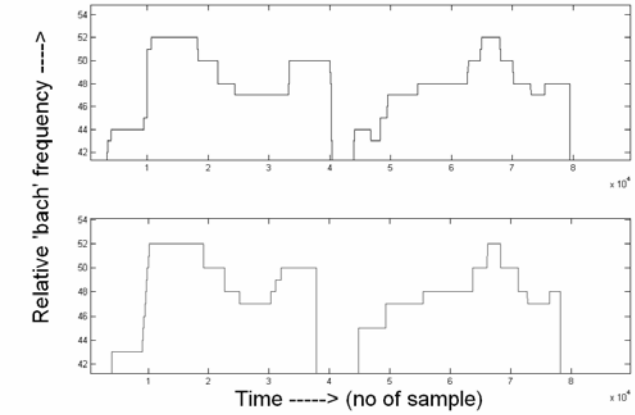

Fig 8 - (a) The pitch contour generated by the algorithm
(b) The pitch contour generated by converting MIDI transcription to relative ‘bach’ frequencies
(F_s=11000Hz)

There are two ways of analyzing the error of the algorithm. The first method does not bother about the onset timings, and finds the fidelity with which the pitch is tracked.

$$\%E = \frac{\sum_{k=1}^{T} abs(\lambda_{auto}(k) + S - \lambda_{midi}(k))}{\sum_{k=1}^{T} abs(\lambda_{auto}(k) - \mu_{midi}(k))} \qquad (24)$$

Where λ_{midi} represents the relative 'bach' frequency contour extracted from the MIDI transcription

The second method employs the use of the onset timings. If a unique automated note onset is detected within 30ms (Ψ) of a manual note onset, then it is considered as a correct detection of onset (%O). If there is no automated onset within 100ms of a manual note onset, it is considered as a Deleted onset(%D). If there is no manual note onset within 100 ms of an automated note onset, then it is considered as an insertion (%I). If the value of the automated relative 'bach' frequency for a correctly onset is equal to that of the manually transcribed one, then it is considered as correct transcription (%T). The results of the 20 files tested are shown in table 1.

Table 1 – Error rates for Normal singing voice (male and female) (10 files)

%E	%O	%D	%I	%T
10%	89%	5%	21%	81%

Table 2 – Error rates for Normal singing voice (male and female) (5files)

%E	%O	%D	%I	%T
12%	84%	9%	12%	79%

Table 3 – Error rates whistling (5 files)

%E	%O	%D	%I	%T
16%	83%	17%	5%	78%

Though extensive testing hasn't been done for the algorithm at this stage, the results are encouraging. Audio reconstruction of the transcribed data was played back to the singers, who approved the transcription quality. It also shows that the algorithm is quite robust for the various forms of queries likely to be made by the users of a query-by-humming system.

5. Conclusions and future work

Thus an algorithm for transcribing human singing voice, into relative pitch contours has been suggested. It is shown to be robust for most forms of human voice queries. However, in order to establish beyond doubt the robustness of the algorithm, further testing has to be done on a larger data set. It may be worthwhile to extend this algorithm to include polyphonic samples predominantly containing the human singing voice.

References

[1] W. J. Hess, "Pitch and voicing determination," in *Advances in Speech signal processing* (S. Furui, M. M. Sondhi, eds.), pp. 3–48, Marcel Dekker, Inc., New York, 1991.

[2] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in S*peech Coding and Synthesis* (W. B. Kleijn and K. K. Paliwal,eds.), ch. 14, pp. 495–518, Elsevier Science, 1995.

[3] A. de Cheveign´e and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, pp. 1917–1930, April 2002.

[4] N. Slonimsky, *Thesaurus of Scales and Melodic Patterns* (NY: Schribner, 1947);

[5] C. Sachs, *The Wellsprings of Music* (New York: McGraw Hill, 1965).

[6] Petre Stoica and Randolph L. Moses, *Introduction to Spectral Analysis* (Prentice Hall publications, New Jersey, pp 46-48)

[7] Davis Pan**,** A Tutorial on MPEG/Audio Compression, *IEEE Multimedia*, *Vol. 2, No. 2*, 1995, pp. 60-74

[8] X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing - A Guide to Theory, Algorithm, and System Development* (New Jersey: Prentice Hall, 2001)