

# CONCATENATION AND STRETCH/SQUEEZE OF MUSICAL INSTRUMENTAL SOUND USING SOUND MORPHING

Naotoshi Osaka

NTT Communication Science Laboratories\*  
3-1 Morinosato Wakamiya, Atsughi-shi Kanagawa, 241-0193, Japan

## ABSTRACT

Sound morphing is one of the successful synthesis technologies for recent computer music, and several studies have been reported on this subject. We have proposed a sound morphing algorithm based on a sinusoidal model. In this paper, a correspondence search algorithm is improved, which is core technology of morphing based on a sinusoidal model. Then wider applications of the algorithm are introduced. It can be used as concatenation of two sounds if morphing interval is as short as 100 msec. It is also used as stretch and squeeze of a sound, if the algorithm is applied to two copies from the same sample. Moreover, adding a musical expression to an instrumental sound are discussed.

## 1. INTRODUCTION

Sound morphing is a synthesis technology that continuously interpolates two timbres and becomes one of the important study items in not only computer music but also in speech research. The author has been developing the technology based on both sinusoidal model [1] and physical model [2]. This paper discusses an improved algorithm and wider application of the technology to various sound modifications along with time alignment, such as stretch/squeeze of a sound, and addition of music expression.

In image morphing, the first step is to find the corresponding points between the two images. In commercial software, this correspondence is done manually. Similarly in sound morphing, there are many cases when correspondence of represented parameters for two sounds should be found and interpolated. Since so many numbers of parameters are needed in a sinusoidal model, it is not practical to find correspondence manually. Therefore, certain algorithm to find the correspondence is indispensable. However, most of the reported technology avoids to solve the problem. Peak to peak match algorithm proposed by [3] works but is an ad hoc algorithm. Fits and Haken also reported an algorithm based on a sinusoidal representation [4]. Their method divides the frequency bands after normalizing by  $f_0$ . Sinusoidal match is done only in the correspondent band, and matching is not done across the different normalized frequency band, which limits the combination of sinusoids.

Our algorithm faces the correspondence problem, and tries to find an optimal solution. We call it an "optimum partner search problem." In the previous study, there were cases of not being able to find the correspondence when sinusoids were placed in complicated manner between facing frames to be interpolated. The new version can find the correspondence in any kinds of sinusoidal placement.

## 2. SEARCH ALGORITHM OF CORRESPONDENCE USING DYNAMIC PROGRAMMING

The algorithm is modified from the original one [1] so that it finds appropriate correspondence for various patterns of peak displacements among two successive frames.

Let  $x_i$ , ( $i=0,1,\dots,I-1$ ) and  $y_j$ , ( $j=0,1,\dots,J-1$ ) denote vectors of two groups.  $I$  and  $J$  are the numbers of vectors, respectively, and usually  $I$  is not equal to  $J$ . The problem is to decide the combination of close resemblance for  $x_i$ - $y_j$  pairs so that combination is an optimum as a whole. It is solved by adopting a criterion that minimizes total cost among pairs, using dynamic programming. Total cost is denoted as follows:

$$T_0(I) = \min \sum_{i=0}^{I-1} C(x_i, y_{w_i(k)}) \quad (1)$$

where  $w_i(k)$  is a window function for  $x_i$  to reduce the search range of  $y$  vectors less than distance  $d_{th}$ , and is defined as specific members in  $y$  group. Figure 1 shows a window function and cumulative cost function. Vectors of each group are lined up in certain order, practically frequency order.  $w_i(k)$  is defined as  $k$ th candidates of  $y$  group within the window, which  $i$ th vector of  $x$  takes into consideration, and is denoted as an absolute number of  $y$  in the order defined above.  $n_i$  is a total number of  $y$  vectors in the window  $w_i$ .

$$w_i(0) = 0, \quad i = 0,1,\dots,I-1 \quad (2)$$

$$\left\{ \begin{array}{l} \text{Case } D(x_i, y_{w_i(k)}) \leq d_{th} \\ \quad w_i(k) \in j, \quad 1 \leq k \leq n_i, \quad j = 0,1,\dots,J-1 \\ \text{Otherwise} \\ \quad w_i(k) = \phi \end{array} \right. \quad (3)$$

\* Currently, with School of Engineering, Tokyo Denki University. E-mail: osaka@im.dendai.ac.jp



Case  $n_{i-1} = 0$

$$(l^*, k^*) = F'(w, l, k, n, l+1) \quad (19)$$

While  $x_i$  points the  $k$ th candidate of  $y$  vector in  $w_i$ ;  $w_i(k)$ ,  $k^*$  is defined as an index when preceding  $x$ ,  $x_{i-1}$ , points the same  $y$  vector.  $k^*$  remains the same definition. Fig.2 compares both  $F$  and  $F'$ .  $d_{null}$  is defined appropriately by a user in the same dimension as a distance. Different values of  $d_{null}$  give the different optimum combination. Fig. 3 gives three examples of such a case.

Three characteristics of the algorithm are;

- 1) A distance function to sort vector members in order to use a window function and cost function to find correspondence of  $x$  and  $y$  vectors do not have to be the same distance function.
- 2) Total cost from  $x$  to  $y$  and  $y$  to  $x$  are not the same and does not satisfy reflexive law, or reflexive law is not proved.
- 3) When sorting is done using a distance function defined in a window function, optimum combination is derived so as not to cross over the correspondent member, that is, combination of, say, 3-7 and 4-5 do not happen.

Fig. 4 depicts applied results of the algorithm to peak to peak match problem. For a) through c), distance function to regulate the window function is defined in one dimension using frequency, as well as the function used to line up peaks (a) and b)). On the other hand, in c), two dimensional distance function in terms of frequency and amplitude is used in calculating distance of  $x$  and  $i$ th candidate of a partner, that is,  $D(x_i, y_{wi(k)})$ . In each of a) to c), matching is done to the near candidate, having nothing to do with the number of members. In c), we can tell that matching is done, taking amplitude into consideration.

### 3. APPLICATION TO THE CONCATENATION OF SOUND FRAGMENTS AND EXPRESSION ADDITION TO THE SUSTAINED INSTRUMENTAL SOUNDS

The algorithm stated in the previous chapter is used for both peak to peak match of the two succeeding frames in sinusoidal model, and morphing of two original sounds. Other applications introduced here are the special cases of morphing when transferring time from one sound to another is very short; concatenation. Three functions are introduced here:

1. Concatenation of two original sounds with different timbres.
2. Concatenation of two original sounds with the same timbre.
3. Stretch/Squeeze of a single sound.

For 1 and 2, if morphing is done in as short as about 100 msec, it is no longer morphing, but becomes concatenation perceptually. Fig. 5 shows these functions.

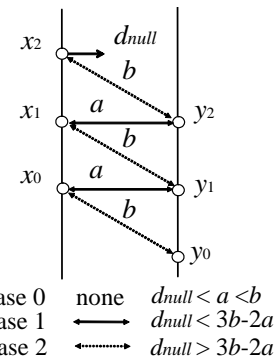


Fig.3 Different combination results for various values of  $d_{null}$

(1) depicts a concatenation of two different timbres. This can be applied to speech synthesis.

#### 3.1. Addition of music expression by controlling concatenation interval

(2)-1 of fig. 5 depicts a concatenation of two samples with the same timbre. Concatenation of different expression for the same instrument is possible. In string instruments, concatenation timing and interval can control the expression of slurring and bowing.

(2)-2 of fig.5 shows another example of a concatenation for different pitch sounds. This can be interpreted as an implementation of portament.

#### 3.2. Stretch and squeeze of sound

(3)-1 and 2 of fig. 5 depict the stretch and squeeze of a sound, respectively. Stretch requires two copies of a sample. Concatenation is done for the overlapped interval, and the sound is stretched as long as the time delay of the latter copy. On the other hand, squeezing is done for the truncated copies of an original sound. The former uses the head part of the copy, while the latter uses the tail part of the copy. These functions are useful in music composition when sound resources are limited.

So far these technologies have been applied to a real computer music pieces: "Nubatama II", limited sound resources for a tiny baby are used, and sound interval are controlled appropriately by stretch and squeeze. All the functions introduced from (1) through (3), including intrinsic morphing function.

## 4. CONCLUSION

Improved version of a correspondence search algorithm, using dynamic programming, is proposed, which is a core technology of sound morphing based on a sinusoidal model. The algorithm is rather complicated, but works for complicated combinations of sinusoidal patterns for real sounds. Variety of applications besides morphing are introduced, starting peak to peak match of succeeding two frame of a sample, and including music expression addition and sound duration control. Finding other music applications using this algorithm is a topic for the further study.

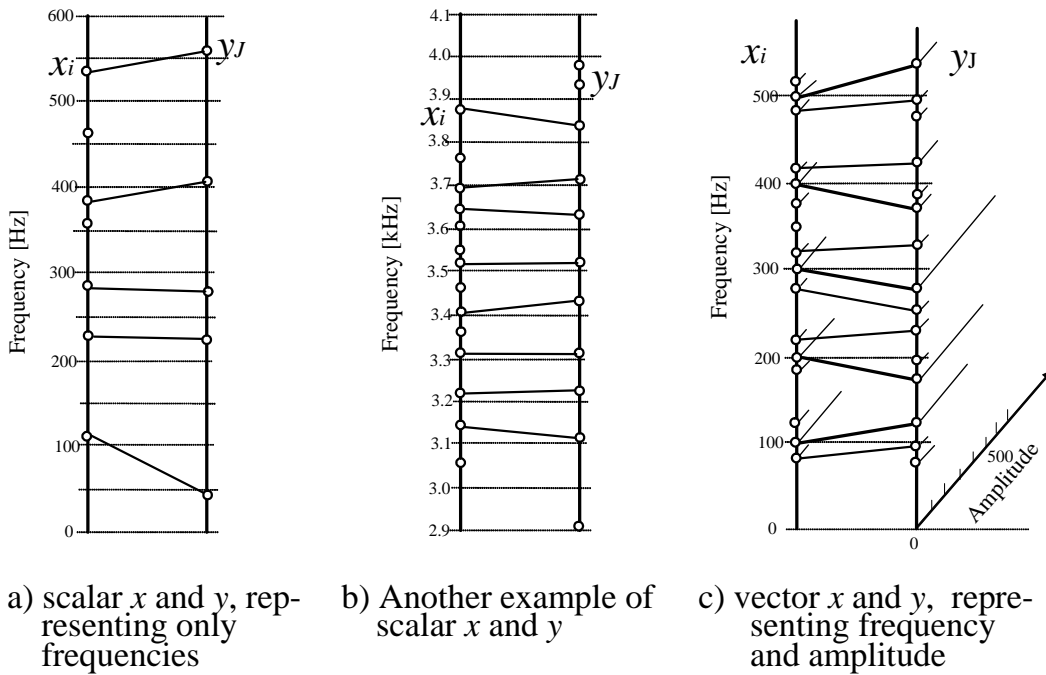
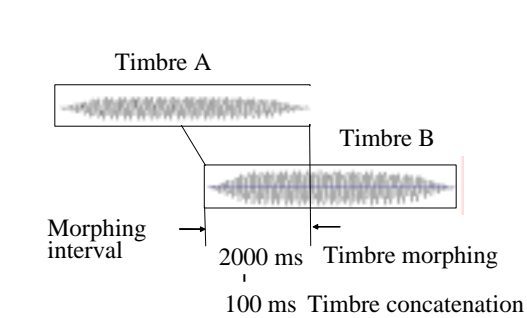
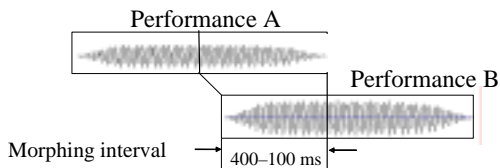


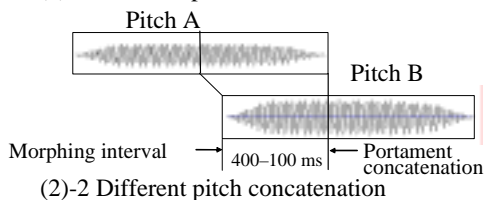
Fig. 4 Examples of peak to peak match



(1) Musically meaningful timber morphing



(2-1) Different performance concatenation



(2-2) Different pitch concatenation

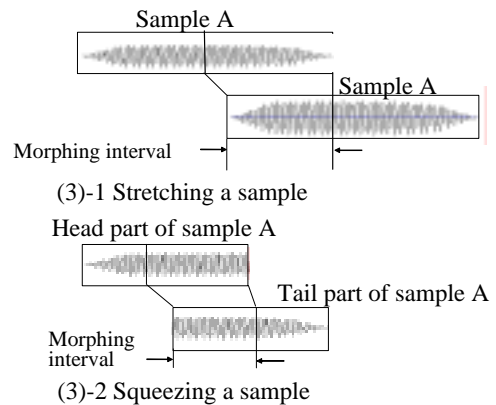


Fig. 5 Various applications of sound morphing

## 5. REFERENCES

- [1] Osaka, N. "Timbre interpolation of sounds using a sinusoidal model," *Proc. of ICMC '95*, pp. 408-411, Banff, Canada, 1995.
- [2] Hikichi, T. and Osaka, N. "Sound timbre interpolation based on physical modeling," *Acoustical Science and Technology*, pp. 101-111, Feb. 2001.
- [3] McAulay, R. and Quatieri, T., "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on ASSP*, vol. ASSP-34, No4. pp. 744-754, Aug. 1986.
- [4] Fits, K., Haken, L., Lefvert, and S., O'Donnell, M. "Sound morphing using Loris and the reassigned bandwidth enhanced additive sound model: Practice and applications," *Proc. Of ICMC '02*, pp. 393-400, Gothenburg, 2002.