

音モーフィング技術を用いた楽音の接続と伸縮

小坂直敏

日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
〒243-0198 神奈川県厚木市森の里若宮 3-1
Osaka@brl.ntt.co.jp

あらまし 音モーフィングはコンピュータ音楽のための音合成技術の中でも近年発展してきた新しい合成技術であり、その実現方法が議論されている。筆者は、これまで楽音を対象として正弦波モデルに基づいたモーフィングの方式を検討してきた。本報告では、まずこの方式で採用している対応する部分音の算出方法に関して筆者らが提案した手法の改良法について詳細に述べる。また、この物理的特徴を明らかにする。これにより、適用する時間領域を100msec程度に短くすることにより音の接続に利用できること、さらに同一音に本方式を適用して音の伸縮に利用できることを示す。また、持続楽器音では一部演奏表情の制御にも利用できることを示す。

キーワード 音合成, 音モーフィング, 接続, 伸縮

Concatenation and Stretch/Squeeze of Musical Instrumental Sound using Sound Morphing

Naotoshi Osaka

NTT Communication Science Laboratories
3-1 Wakamiya, Morinosato, Atsugi-shi, Kanagawa, 243-0198
Osaka@brl.ntt.co.jp

Abstract Sound morphing is one of the successful synthesis technology 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, physical characteristics of correspondent partials matching algorithm is discussed. Then wider applications of morphing algorithm are introduced. The algorithm can be used as concatenation of two sounds if time region when morphing is applied is as small as 100 msec. It is also used as stretching and squeezing of a sound, if the same samples or two samples of similar timbre are chosen. Moreover, some examples of attaching a musical expression to a violin sound are introduced.

Key words Sound morphing, sound synthesis, concatenation, stretch/ squeeze

1. はじめに

音のモーフィングは、二つの音色を連続的に補間する合成技術で、コンピュータ音楽などのコンテンツ創作技術として、あるいはまた、音声研究

の枠組みとしても検討されており、近年急速に発展している。筆者らは信号モデル(信号のスペクトルに基づくモデル)と、物理モデル(音の発音機構を力学的に表現するモデル)の二つの手法につい

て検討を重ねてきた[1] [2] [3] [4] .信号モデルでは、正弦波モデルに基づいて基本的な方式を97年に発表し、Windows上で動作する音合成システム「おっくんしゃい」(音と声と一緒にするシステム、*O'kinshi; Qto to koega isshoni naru shisutemu*) [5]に実装したり、モーフィング音を実コンピュータ音楽作品への主題として取りこむなどの応用によりその意義を訴えてきた。本報告では、信号モデルによるモーフィング技術に横たわる問題を整理し、本方式の特徴について示す。また、応用について、モーフィングを適用する時間領域を調整することにより音接続に応用できること、また、同一サンプル音二つに適用することにより音の継続時間長さを伸長あるいは圧縮できることを示す。また、接続時の時間領域を増減することにより、演奏表情をも変えられることを示す。

2. 音モーフィング技術

楽音を対象としたモーフィングは筆者らの報告[1]~[4]のほか、Fits, Hakenら[6]の報告などがある。これらは正弦波モデルを用いて、二つの原音を部分音の和として分解し、これらを個々に接続する方法が一般的である。

一方音声研究でもいくつか報告されている。阿部はFFTから得られるスペクトル上で低域と高域を補間率に応じた割合で区切り、二つの原音を低域と高域それぞれに割り当てて新たにスペクトルを構成し、これを逆変換することによりモーフィング音を合成している[7]。このほか坂野ら[8]、河原ら[9]などにより別の手法が検討されている。なお、楽音とは逆に、音声研究では正弦波モデルを用いてのモーフィングは一般的ではない。むしろソースフィルタモデル、ボコーダなどの音声の基本的な表現の枠組みの中で発想している。

3. 部分音の対応探索アルゴリズム

画像のモーフィングでは二つの画像の対応点を見つけることが処理の第一歩である。一般に、既存の商用画像ソフトでは対応点の導出はユーザが手動で行っている。同様に、音モーフィングも各方式で表現されたパラメータ上で二つの音の対応点を見つけ、対応するもの同士の補間を行うことになる。音を少ないパラメータで表現ができない場合、手動で対応点を見つける方法は実際の的でなく、自動的に算出する必要がある。正弦波モデルを基本的音表現として用いる場合、モーフィングを行うために両原音の部分音の対応を見つけるア

ルゴリズムが必要となる。

Fitsらの方法[6]では、まず有ピッチ区間をF0で正規化し、周波数帯を分割する。これをチャンネルと呼ぶ。次に、対応するチャンネル内での対応する正弦波を見出す処理を行う。通常一つのチャンネルには調波が一つ、あるいは非調波音があるとしても、高々2個のエネルギーの高い正弦波が存在する。従って、対応するチャンネル間で強度の大きいもの同士を対応させる。

一方、河原の手法[9]ではソナグラム上で数点の数少ない対応点を手動で指定することにより全体の補間に対する指針を得ている。この方法は手動ではあるが、簡易であるため、今後とも用いられていく可能性が高い。

これらの手法は、次元数の異なる部分音、あるいは、パラメータのすべての対応点をとる処理そのものは回避している。この問題を回避することにより、以下の問題が生ずる。1)パワーの大きな調波は対応点が見つからなくてもあらゆる部分音に対処していないため、モーフィング音質が劣化する。2)各帯域(チャンネル)に区切り、その範囲内での対応点を探索するため、実際には距離が近くともチャンネルが異なると対応されない、などの欠点がある。

3.1 部分音同士の対応

先に報告した方式[2]は、ピッチを有すハーモニクス構造が明確な音同士には機能したが、違いに異なる複雑な部分音の配置となる場合では対応点がうまく探索できない場合もあった。そこで複雑なパターンでも対応点が探索できるよう、アルゴリズムを改良した。

4. DPによる対応関係の探索アルゴリズム

大枠は文献[2]に示したとおりだが、さまざまな部分音配置パターンに対処できるよう、以下のように改良した。

いま、 x_i ($i=0, \dots, I-1$)と y_j ($j=0, \dots, J-1$)を二つのグループのベクトルを表すとする。

I と J はベクトルの数で通常 $I > J$ である。問題は I 個のベクトルと J 個のベクトルをできるだけ近い(似通った)相手を見つけ、全体として最適となるようベクトルの組み合わせを決定することである。これは、DPを用いて考慮するペアの中の総合的な距離を最小とするような基準で解くことができる。総合評価は以下のように表される。

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

ここに、 $w_i(k)$ は x_i に対する y グループの探索範囲

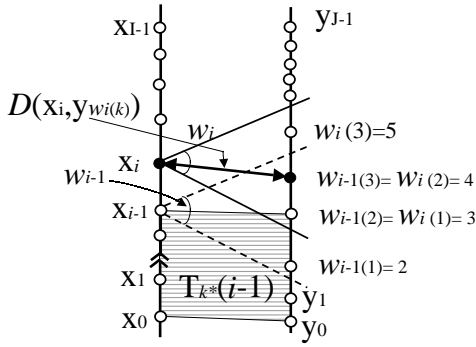


Fig.1 Cumulative cost $T_{k^*}(i-1)$ used in the algorithm

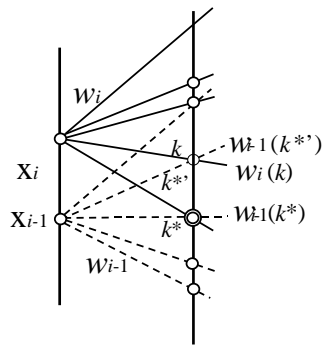


Fig. 2 Comparison of k^* and $k^{*'}$ acquired from functions F and F'

を減少させるための窓関数で、 y グループの具体的なメンバーを表す。図1は窓関数および累積コストの説明図である。各グループのベクトルをDPで用いる距離関数と同様な関数で線分上に小さい順に並べる、そのとき、 x 側 i 番目のベクトルが扱う k 番目(1始まり)の y 側のメンバーを番号で表すものである。

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

$$W_i(k) = j, \quad 1 \leq k \leq n_i, \quad j=0, \dots, J-1 \quad (3)$$

ここに、

$T_k(i)$ は x_i から $y_{w_i(k)}$ までの累積コスト関数、 C はコスト関数で、対応するベクトルに対しては距離関数 $D(x_i, y_{w_i(k)})$ で表され、対応する相手がない場合は d_{null} で表される。

$T_0(i)$ は x_i が対応するベクトルがないとき、前段の x_{i-1} が $y_{w_i(k)}$ までを対象にしたときの累積コストである。 T_0 は相手が見つからない場合に限ってさらに累積コスト関数 T の次元が一つ増える必要がある(3次元)。漸化式は以下で与えられる。

初期条件 ($i=0$ のとき)

$$T_0(0) = d_{null} \quad (4)$$

for $k = 1, \dots, n_0$

$$T_k(0) = \min (D(x_0, y_{w_0(k)}), T_{k-1}(0)) \quad (5)$$

$i=0$ のとき

for $k=1, \dots, n_1$

$$T_k(i) = \min \begin{cases} k-1 & T_{k-1}(i), \quad k=1 \\ D(x_i, y_{w_i(k)}) + T_{k^*}(i-j^*) \\ D(x_i, y_{w_i(k)}) + T_0(0) \end{cases} \quad (6)$$

$i>1$ のとき

for $k=1, \dots, n_i$

$$T_k(i) = \min \begin{cases} k-1 & T_{k-1}(i), \quad k=1 \\ D(x_i, y_{w_i(k)}) + T_{k^*}(i-j^*) + d_{all} \\ D(x_i, y_{w_i(k)}) + T_0(i-j^*) + d_{all} \end{cases} \quad (7)$$

ここに

$$d_{all} = d_{null} \times (j^*-1) \quad (8)$$

終了条件

$$T_0(I) = \min \begin{cases} T_{n-1}(I-1) \\ T_0(I-1)_{n-1} \end{cases} \quad (9)$$

なお、ここに k^*, j^* は $y_{w_i(k)}$ を指し示すための1段下の k の値の意味で、以下に再帰的に定義される。

$$(j^*, k^*) = F(w, I, k, n, j) \quad (10)$$

初期値

$j=1$

$$i-j < 0 \text{ のとき } k^*=0, \quad j^*=I \quad (11)$$

$n_{i-j} = 0$ のとき

$$(j^*, k^*) = \begin{cases} w_{i-j}(1) \quad w_i(k)-1 \quad w_{i-j}(n_{i-j}) \text{ のとき} \\ \quad (j^*=j, k^*=w^{-1}_{i-j}(w_i(k)-1)) \\ w_{i-j}(n_{i-j}) \quad w_i(k)-1 \text{ のとき} \\ \quad (j^*=j, k^*=n_{i-j}) \\ w_i(k)-1 < w_{i-j}(1) \text{ のとき} \\ \quad F(w, I, k, n, j+1) \end{cases} \quad (12)$$

$n_{i-j} = 0$ のとき

$$(j^*, k^*) = F(w, I, k, n, j+1) \quad (13)$$

また、 $T_0(i)$ は具体的に以下に定義する。

$$T_0(0) = d_{null} \quad (14)$$

$$T_0(i)k = d_{null} \times (j^*-1) + \min \begin{cases} T_{k^*}(i-j^*), j^* = 1, \\ \text{, 其他の場合} \\ T_0(i-j^*)_{k^*} \end{cases} \quad (15)$$

k^*, j^* もFと同様に以下のように定義される。

$$(j^*, k^*) = F'(w, I, k, n, j) \quad (16)$$

初期値

$$j=1$$

$$i-j < 0 \text{ のとき } k^*=0, j^*=I \quad (17)$$

$n_{i-j} = 0$ のとき

$$(j^*, k^*) = \begin{cases} w_{i-j}(1) \quad w_i(k) \quad w_{i-j}(n_{i-j}) \text{ のとき} \\ \quad (j^*=j, k^*=w^{-1}_{i-j}(w_i(k))) \\ w_{i-j}(n_{i-j}) \quad w_i(k) \text{ のとき} \\ \quad (j^*=j, k^*=n_{i-j}) \\ w_i(k) < w_{i-j}(1) \text{ のとき} \\ \quad F'(w, I, k, n, j+1) \end{cases} \quad (18)$$

$n_{i-j} = 0$ のとき

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

ここで、FとF'の違いを図2に示した。iがk番目に $w_i(k)$ を指し示すとき、一段前のi-1が同一のyを指し示すときのi-1における指標が k^* で、一つ前のy(印)を指し示すポイントが k^* となる。 d_{null} は、距離と同じ次元で定義されるが、これはユーザが自由に設定できる。この値により対応が異なる例を図3に示す。

このアルゴリズムの特徴は、1)窓関数を設けて対象となる相手を選ぶための距離関数、およびそれに基づくソーティングと、実際に相手との距離を計算するための距離関数は同一でなくてよい、2) x から y の総合コストと y から x への総合コストが反射的でない、3)窓関数で定義される距離関数の順で並べられたとき、対応する相手同士を線でつなぐと飛び越しをおこさない、などの点にある。FとF'関数の定義は特徴3)のために工夫されたものである。

図4に部分音間の対応例を示す。a)-c)とも、窓関数はそれぞれの部分音の周波数のみを選び、これらの距離で並べられている。こうして並べられたものに対し、個々の部分音の距離 $D(x_i, y_{wi(k)})$ は、a), b)では周波数のみのスカラーより、またc)は x, y を周波数と振幅値の2次元ベクトルより算出している。各図とも、メンバー数に関係なく、近いものを見つけられていること、また、c)は振

幅が大きさも考慮されて対応がみつけれられていることがわかる。固定帯域に区切って、その中で少ないベクトルの対応を見つけていくやり方は、は対応する帯域外では対応がとれない。

4. 音接続への応用と持続楽音への演奏表情づけ

モーフィング区間を短時間とすることで、「音色の移行」とは別の以下の3つ機能を誕生させた。これらの様子を図5に示す。

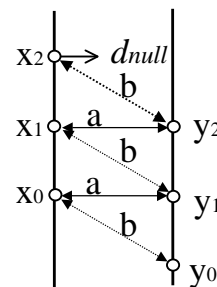
- (1) 異なる音色を持つ二つの原音の接続
- (2) 同一音色を持つ二つの原音の接続
- (3) 単音の伸縮機能

(1), (2)では二つの音色を100ms前後の短い時間でモーフィングさせたとき、知覚的にはもはやモーフィングではなく接続となることを意味している。

(1)は異なる音色のモーフィングで、本技術の本来の使用方法であり、音楽的に意味のあるものである。このモーフィングの区間をこれまでは1秒から2秒程度としてきたが、100ms程度とすると、異なる音色の接続となる。こうした接続は、音声素編の接続法としても期待され、別途検討を進めている[10]。

4.1 接続時間長の制御による演奏表情の付与

(2)は、同一音色の異なるサンプルを対象にした接続で、(1)を接続と捉えるなら、これを同一音色の場合に適用するものである。ひとつは、演奏表情の異なるものの接続も考えられる。弦楽器の場合など、どの時点で接続するかにより、スラー、スラーなし、などのボーイングの違いによる演奏表情の差が表現されると期待できる。



- Case 0 none $d_{null} < a < b$
- Case 1 $\longleftrightarrow d_{null} < 3b-2a$
- Case 2 $\longleftrightarrow d_{null} > 3b-2a$

Fig. 3 Examples of different results for various values of d_{null}

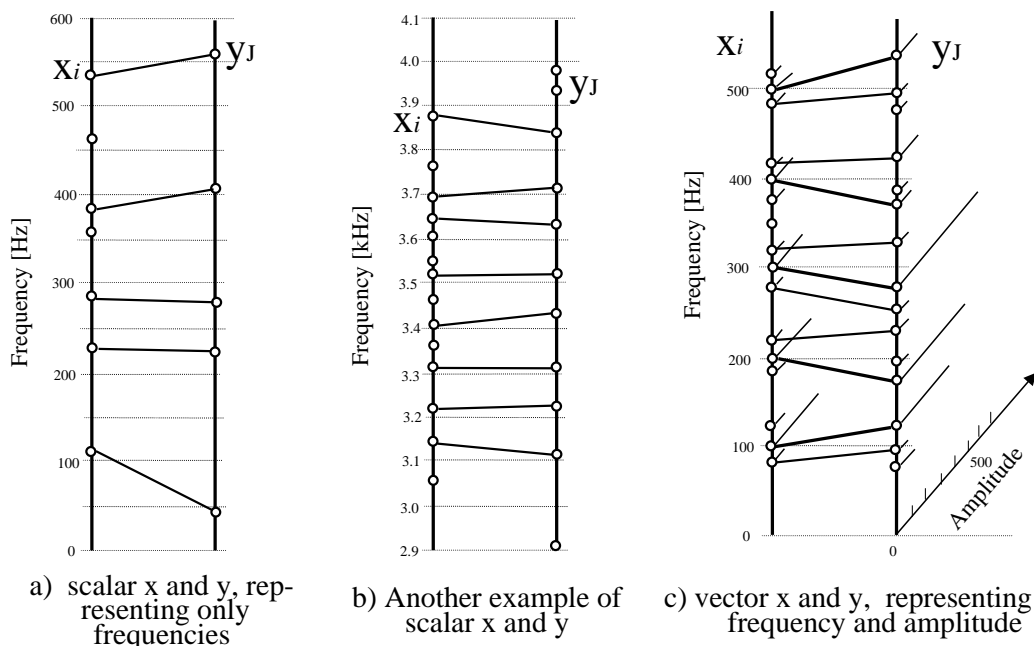


Fig. 4 Examples of partial to partial match

このほかに、同一音色でピッチが異なるものの接続が考えられる。これはモーフィングの応用、という点を考慮すると、ポルタメントという演奏表情を付加された接続ということができる。

4.2音の伸張と圧縮

(3)は同一音色、あるいは同一サンプルを対象に、この短時間のモーフィングを施せば音の伸縮が可能となることを現している。伸張では、同一サンプルを適当な場所をモーフィング区間とし、接続するものである。この伸縮機能そのものは、音楽的には大きな意味を持たないが、限られたサンプルしか得られないような場合はコンテンツ制作上非常に便利である。

5. 楽曲創作への応用

5.1「射干玉 II -尺八とコンピュータのための」への応用

筆者が制作した楽曲の多くは、新しい音合成方式の紹介、あるいは、その音合成方式から得られる新しい音色の音楽的役割を紹介することを目的としている。

「鏡石」[M1]では、フルート音からソプラノへのモーフィングを行った。このときのモーフィング区間は1.5秒程度である。また、Prosody++[M2]では、音の連続的な移行ではなく、中間音の合成を用いている。例えば女声とフルート音、女声と

クラリネット、などがその応用例である。

5.2「射干玉 II」で用いられた新しい機能

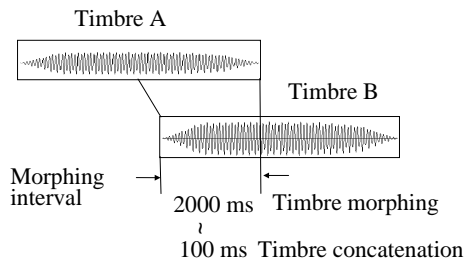
別の作品「射干玉 II」[M3]では、モーフィングの応用として、モーフィング区間の時間的適用範囲を100 msecの短時間から数秒の長時間まで広げた。ここでは、限定されたサンプル音(子供の声)から伸縮、圧縮などによりモーフィングが可能な長さまで調整した後、音楽的意味の大きなモーフィング音を合成した。すなわち、先に紹介した(1)から(3)の機能をすべて適用し、制作の意図を達成することができた。

6. おわりに

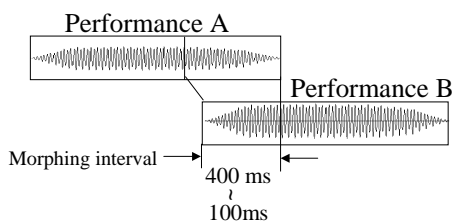
モーフィング技術を紹介し、これをモーフィング以外に用いて音の接続、同一音の伸縮、演奏表情付け、などに応用できることを示した。現在、筆者らは、音合成システム「おっくんしゃい」[5]の上にモーフィングアルゴリズムを実装するべく検討を進めている。なお、同システムは、研究、制作目的の使用を前提に、モニターとして配布することを計画している。URL は以下のとおりで、11月頃にダウンロード可能となるよう準備を進めている。

<http://www.brl.ntt.co.jp/info/srcm/Otkinshi/index-j.html>

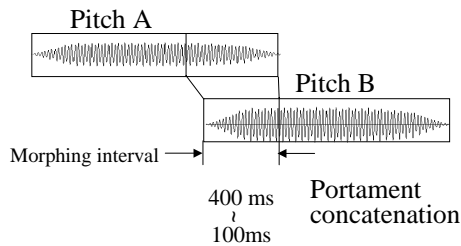
今後はモーフィング音の主観評価を行って品質を明確にした後、性能向上を図りたい。



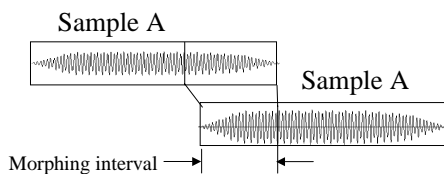
(1) Musically meaningful timber morphing



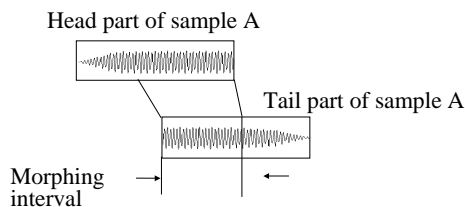
(2)-1 Different performance concatenation



(2)-2 Different pitch concatenation



(3)-1 Stretching a sample



(3)-2 Squeezing a sample

Fig. 5 Various applications of timbre Morphing and its functions

謝辞

この研究を行う機会を与えていただいた村瀬洋メディア情報研究部長, および研究を進めるにあたり日頃議論していただくメディア表現グループの諸氏に感謝します。

参考文献

- [1] N. Osaka, "Timbre interpolation of sounds using a sinusoidal model," *Proc. of ICMC'95*, pp. 408- 411, Banff, 1995.
- [2] 小坂直敏, "Sinusoidal modelによる音色の補間," 電子情報通信学会技術報告 SP95-130, pp. 9-16, 1996.
- [3] 引地 孝文, 小坂 直敏, "打弦, 撥弦, および弾性体振動のモルフィングの検討", 電子情報通信学会技術報告 SP96-111, pp. 23-28, 1997.
- [4] T. Hikichi and N. Osaka, "Sound timbre interpolation based on physical model- ing," *J. Acoust. Soc. Jpn.*, Feb. 2001.
- [5] 小坂直敏, 榊原健一, 引地孝文, "Windows上の音合成システム「おっきいんしゃい」の構築," 信学論D-II Vol.J84-D-II No.6, pp.946-954, 2001.6.
- [6] K. Fitz, L. Haken, S. Lefvert, M. O' Donnell, "Sound morphing using Loris and the reassigned bandwidth- enhanced additive sound model: Practice and applications," *Proc. of ICMC 02*, pp.393-400, Goteborg, 2002. 9.
- [7] 阿部匡伸, 「基本周波数とスペクトルの漸次変形による音声モーフィング」, 音響学会'95秋季大会, 2-1-8, pp.259-260, 1995. 9.
- [8] 坂野秀樹, 武田一哉, 鹿野清弘, 板倉文忠, 「包絡と音源の独立操作による音声モーフィング」, 信学誌A Vol. J81A No.2 pp.261-268, 1998. 2.
- [9] 松井九美, 河原英紀, 「聴覚情報表現上の距離を利用した音声の感情モーフィングの一検討」, 音響学会 '02秋季大会, 1-P-25, pp. 421-422, 2002.9.
- [10] 宮崎昇, 小坂直敏, 「高品質ピッチ変換と音モーフィングに基づく音声合成」, 音響学会 '02秋季大会, 1-P-24, pp.419-420, 2002.9.
- [11] K. J. Hebel and C. Scaletti, "The software architecture of the Kyma System," *Proc. of ICMC 93*, pp. 164-167, Tokyo, 1993.

音楽演奏

- [M1] 小坂直敏, 「鏡石 -- フルートとコンピュータのための」日本コンピュータ音楽協会主催コンピュータ音楽の現在 II, 神戸 ジーバックホール, 1996.7.13
- [M2] 小坂直敏, 「Prosody++ --フルート, バイオリン, チェロ, ピアノとコンピュータのための」MUSIANA '95, Louisiana Museum, Denmark, 1995. 9
- [M3] 小坂直敏, 「射干玉 II -- 尺八とコンピュータのための」けいはんなメディアフェスティバル 2002, 8.30.