

▶ **音空間レンダリング技術の開発
～シリコンコンサートホールの実現に向けて～**

- 土屋隆生(同志社大学),
岩谷幸雄(東北学院大学),
大谷 真(信州大学),
井口 寧(北陸先端科学技術大学院大学)

レンダリングと音空間レンダリング



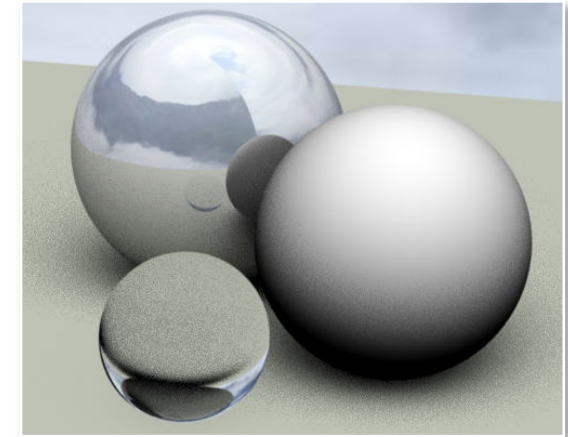
レンダリング

形状データ



画像による可視化

- ・レンダラー (GPU: Graphics Processing Unit)により, 2次元ディスプレイにレンダリング



音空間レンダリング

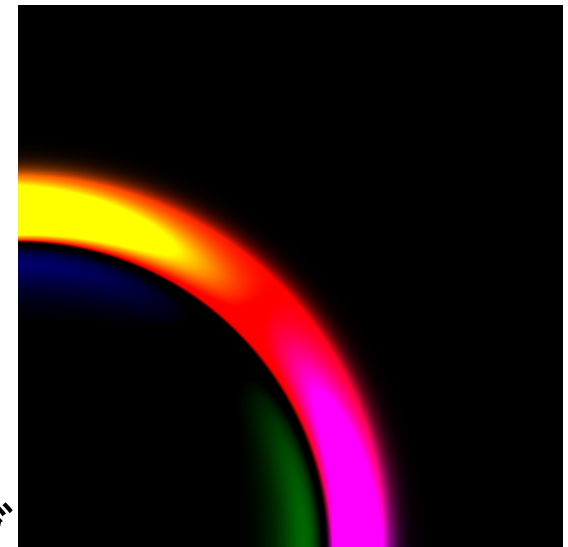
形状データ



音響信号による可聴化

波動性考慮の必要性

- ・音空間レンダラー (SPU: Sound field Processing Unit)により, スピーカにレンダリング



シリコンコンサートホール

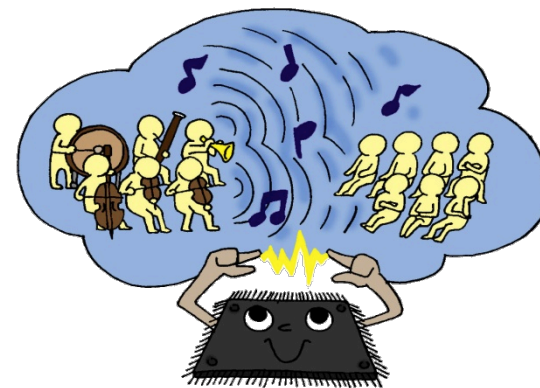


リアルタイム音空間レンダリング

- 音波の波動性を考慮した音空間レンダリングをリアルタイムで実現する技術

シリコンコンサートホール

- シリコンチップ化したリアルタイム音空間レンダラー(SPU)
- コン서트ホールの音響空間をシリコンチップ内に納めることで、家に居ながらにしてコンサートホールの音響を超臨場体感できる。

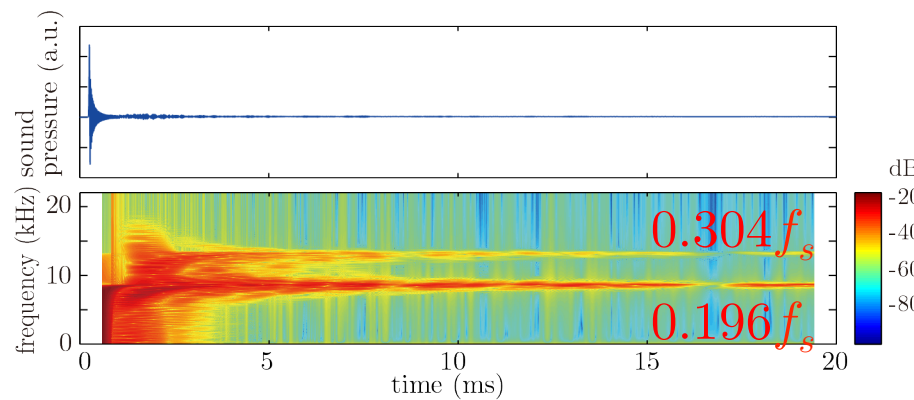
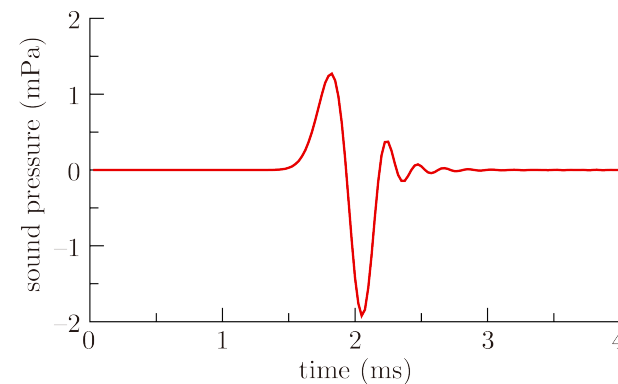
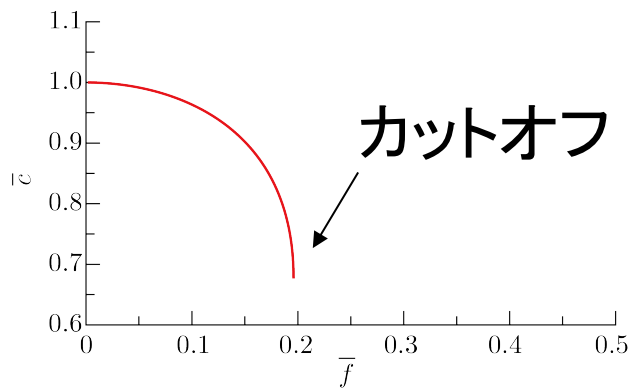


NICTの委託研究

音空間レンダリングの問題点1



□差分(FDTD)法系の手法では数値分散の誤差のために音質が劣化する。



インパルス応答

スペクトログラム

耳障りな雑音として可聴化される

音空間レンダリングの問題点2

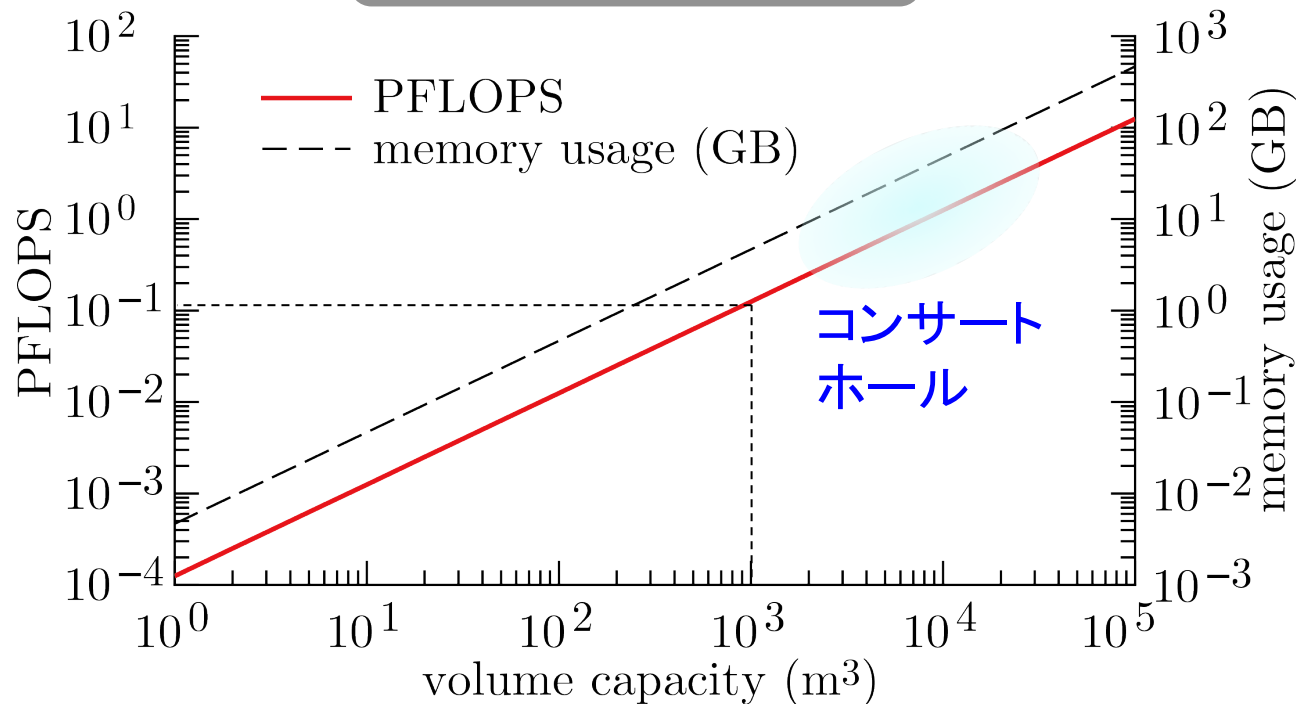


□数値分散の誤差を軽減するにはオーバーサンプリングが必要。



計算機資源の増大

125TFLOPS/1000m³



空気中で1秒間のインパルス応答を計算するのに必要な計算機資源

計算機資源の増大に対する対処法



NICT委託研究プロジェクト(H21-24)

「革新的な三次元映像技術による超臨場感コミュニケーション技術の研究開発」

課題ウ 五感コミュニケーションの中核的要素技術

同志社大学

FDTD法の高精度化



CE-FDTD法

北陸先端科学技術大学院大学

専用ハードウェアによる高速化



FPGA

東北学院大学・信州大学

新たな計算手法の試み



音響LGA法

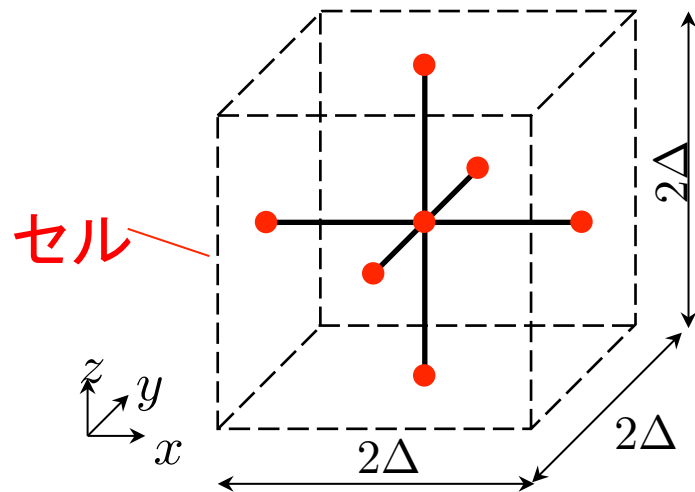


FDTDの高精度化

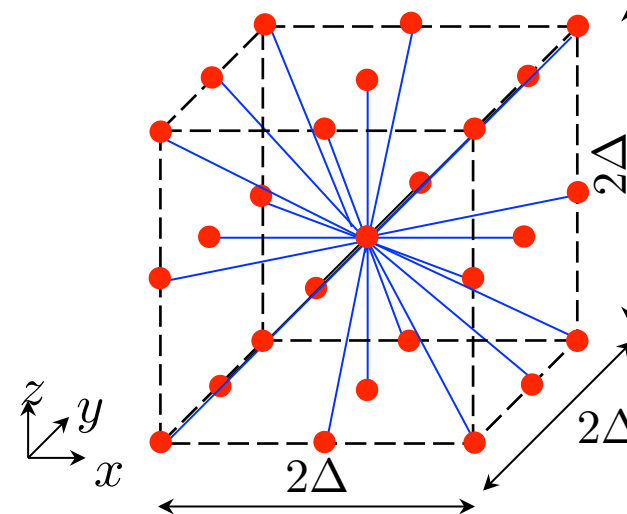
(CE-FDTD法)



□空間差分の高次化で対処



standard FDTD
standardleapfrog (SLF)



Compact Explicit-FDTD
(CE-FDTD)法

※コンパクト差分は陰的に空間微分値を計算することで高精度化しているが、本手法は陽的に高精度化可能

波動方程式の離散化の一般化



□斜め方向も考慮した離散化

$$\frac{1}{c_0^2} \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2}$$



$$\delta_t^2 p_{i,j,k}^n = \chi^2 \left\{ (\delta_x^2 + \delta_y^2 + \delta_z^2) + a (\delta_x^2 \delta_y^2 + \delta_y^2 \delta_z^2 + \delta_z^2 \delta_x^2) + b \delta_x^2 \delta_y^2 \delta_z^2 \right\} p_{i,j,k}^n$$



$$\begin{aligned} p_{i,j,k}^{n+1} = & d_1 (p_{i+1,j,k}^n + p_{i-1,j,k}^n + p_{i,j+1,k}^n + p_{i,j-1,k}^n + p_{i,j,k+1}^n + p_{i,j,k-1}^n) \\ & + d_2 (p_{i+1,j+1,k}^n + p_{i+1,j-1,k}^n + p_{i+1,j,k+1}^n + p_{i+1,j,k-1}^n \\ & + p_{i,j+1,k+1}^n + p_{i,j+1,k-1}^n + p_{i,j-1,k+1}^n + p_{i,j-1,k-1}^n \\ & + p_{i-1,j+1,k}^n + p_{i-1,j-1,k}^n + p_{i-1,j,k+1}^n + p_{i-1,j,k-1}^n) \\ & + d_3 (p_{i+1,j+1,k+1}^n + p_{i+1,j-1,k+1}^n + p_{i+1,j+1,k-1}^n + p_{i+1,j-1,k-1}^n \\ & + p_{i-1,j+1,k+1}^n + p_{i-1,j-1,k+1}^n + p_{i-1,j+1,k-1}^n + p_{i-1,j-1,k-1}^n) \\ & + d_4 p_{i,j,k}^n - p_{i,j,k}^{n-1} \end{aligned}$$

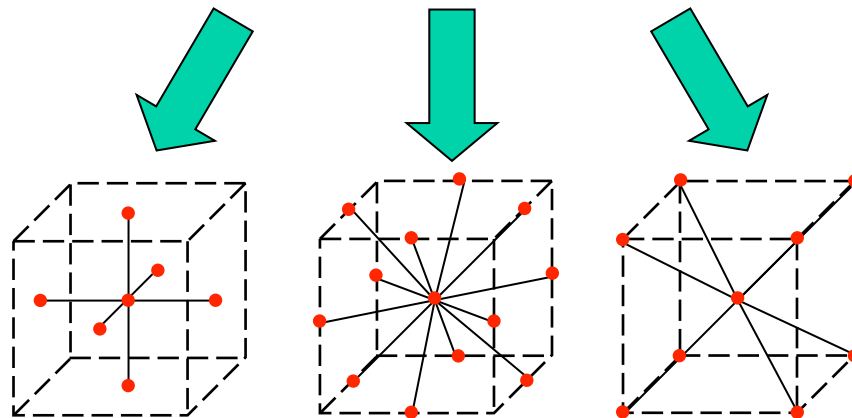
ただし $d_1 = \chi^2(1 - 4a + 4b)$, $d_2 = \chi^2(a - 2b)$, $d_3 = \chi^2 b$,
 $d_4 = 2(1 - 3\chi^2 + 6a\chi^2 - 4b\chi^2)$

さまざまな手法







□パラメータを変更することで精度を変更できる。

	a	b	d_1	d_2	d_3	d_4
SLF	0	0	$1/3$	0	0	0
CCP	$1/4$	0	0	$1/4$	0	-1
OCTA	$1/2$	$1/4$	0	0	$1/4$	0
IISO	$1/6$	0	$1/4$	$1/8$	0	-1
IISO2	$1/6$	$1/48$	$15/48$	$3/32$	$1/64$	$-9/8$
IWB	$1/4$	$1/16$	$1/4$	$1/8$	$1/16$	$-3/2$



各手法の χ 上限値とカットオフ周波数

	χ_m	\bar{f}_{c1}	\bar{f}_{c2}	\bar{f}_{c3}	\bar{f}_c
SLF	$1/\sqrt{3}$	0.196	0.304	0.5	0.196
CCP	1	0.5	0.5	0.333	0.333
OCTA	1	0.5	0.25	0.25	0.25
IISO	$\sqrt{3}/2$	0.333	0.5	0.377	0.333
IISO2	$\sqrt{3}/2$	0.5	0.5	0.333	0.333
IWB	1	0.5	0.5	0.5	0.5

 (1,0,0) (1,1,0) (1,1,1)方向 全体

音質を考慮した数値計算



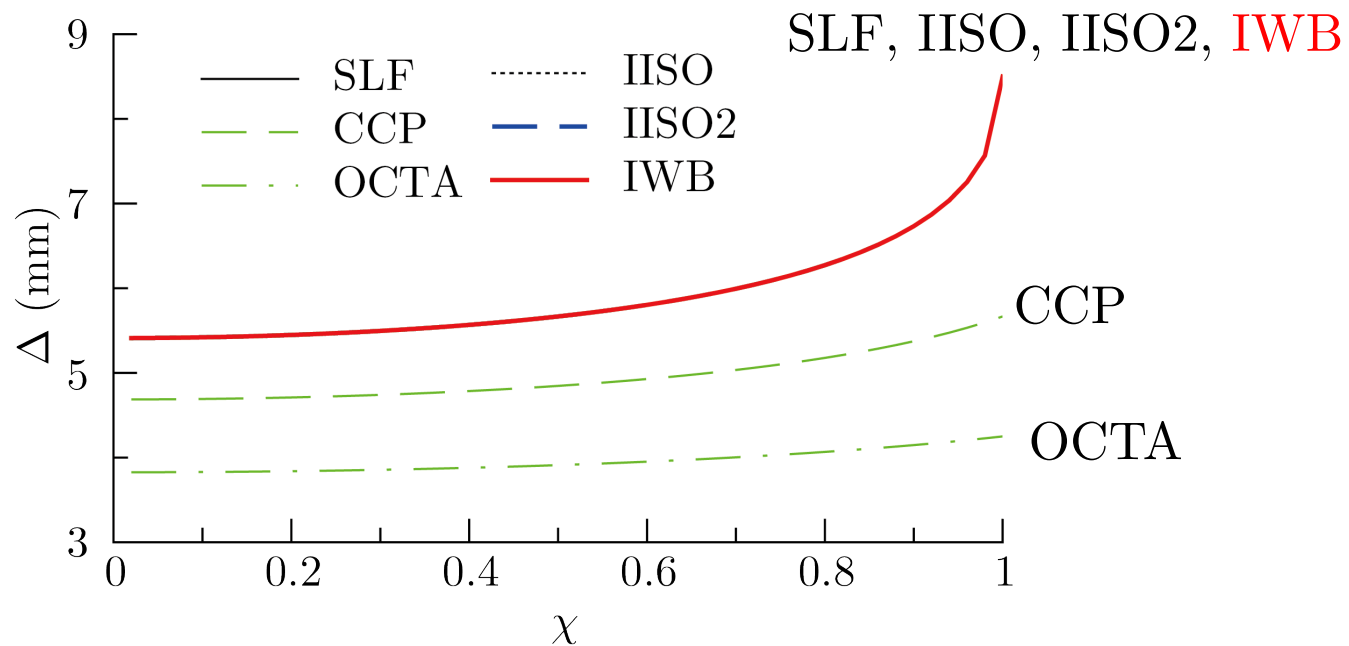
□音質はカットオフ周波数 f_c で制限される。

$$f_b = \bar{f}_c f_s \quad \longrightarrow \quad \Delta t = \frac{\bar{f}_c}{f_b} \quad \longrightarrow \quad \Delta = \frac{c_0 \bar{f}_c}{\chi f_b} \quad \longrightarrow \quad \Delta = \frac{c_0 \sin^{-1}(\chi)}{\pi \chi f_b}$$

確保すべき帯域

$$\chi = \frac{c_0 \Delta t}{\Delta x} \text{ (CFL数)} \quad (1,0,0) \text{ 方向}$$

$f_b = 20\text{kHz}$ の場合



CFL数に対するグリッド間隔

音質を考慮した必要計算資源



メモリ

総格子点数 N に比例 1m^3 あたり $N = \frac{1}{\Delta^3} \propto f_s^3$

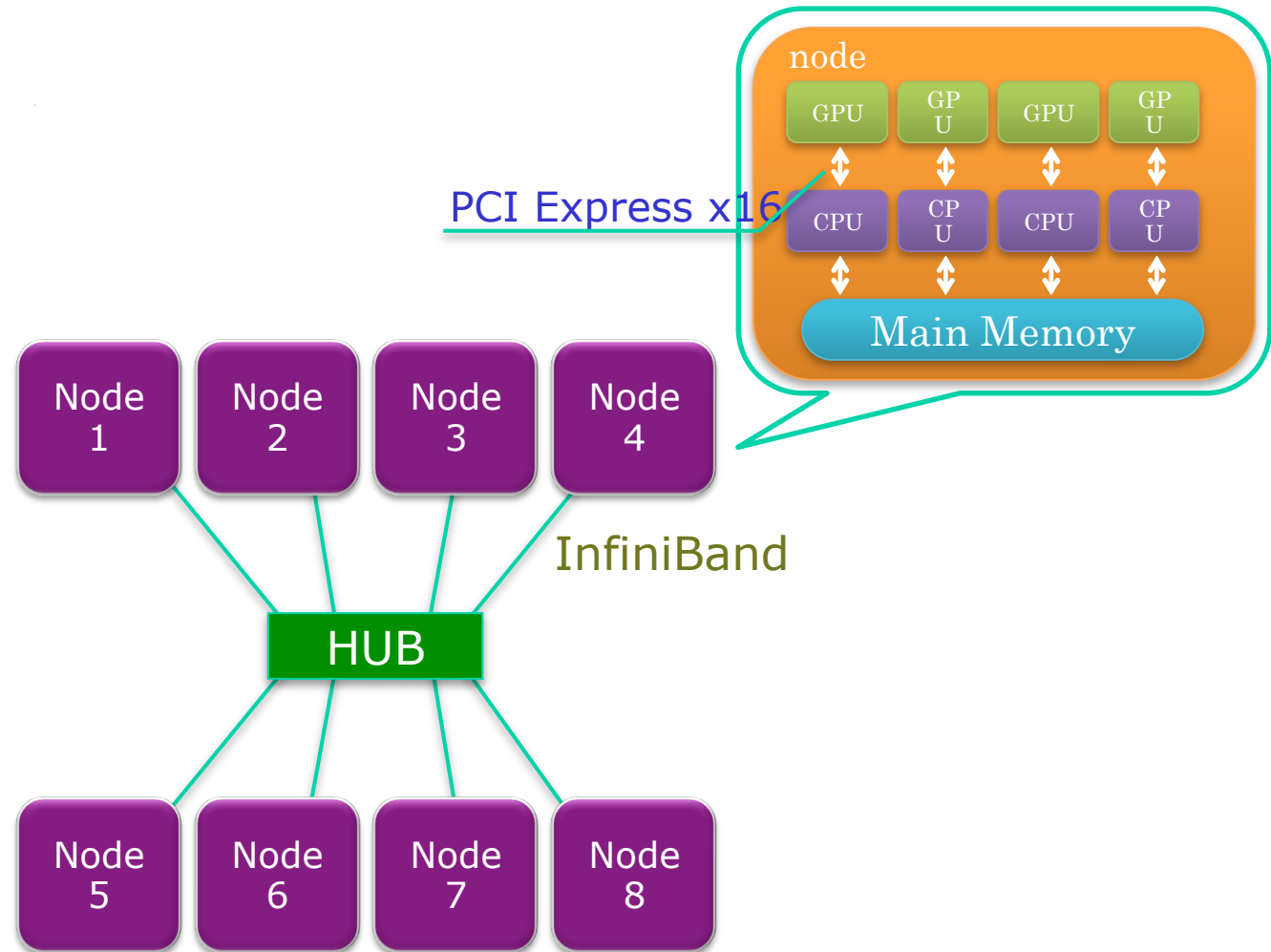
総演算量

N, f_s, n_c に比例 $1\text{m}^3\text{s}$ あたり $N_f = N n_c f_s \propto f_s^4$

$f_b=20\text{kHz}$ の音質確保に必要な計算資源

	χ_m	\bar{f}_c	f_s (kHz)	Δ (mm)	N $\times 10^6$	メモリ比	n_c	N_f $\times 10^{12}$	時間比
SLF	$1/\sqrt{3}$	0.196	102	5.77	5.20	1	9	4.78	1
CCP	1	0.333	60	5.66	5.51	1.06	15	4.97	1.04
OCTA	1	0.25	80	4.25	13.0	2.50	11	11.5	2.40
IISO	$\sqrt{3}/2$	0.333	60	6.54	3.58	0.688	21	4.52	0.95
IISO2	$\sqrt{3}/2$	0.5	60	6.54	3.58	0.688	31	6.67	1.40
IWB	1	0.5	40	8.50	1.63	0.313	31	2.02	0.42
SLF4	0.5	0.196	102	6.66	3.38	0.650	24	8.28	1.73

GPUクラスタ



GPU: NVIDIA Tesla M2075 × 32

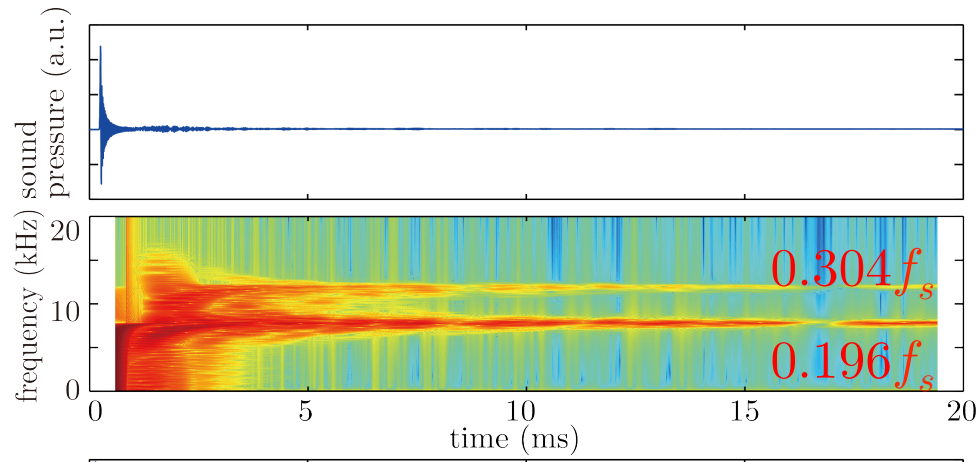
CPU: Intel Xeon E5620

InfiniBand: Mellanox 8X QDR

インパルス応答



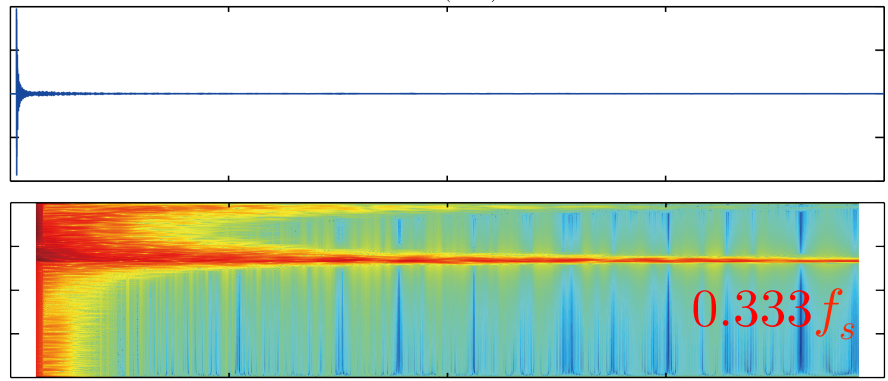
SLF



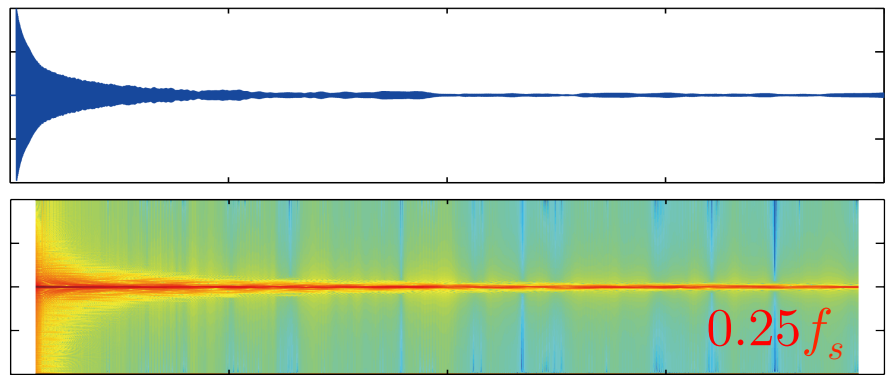
音圧波形

スペクトログラム

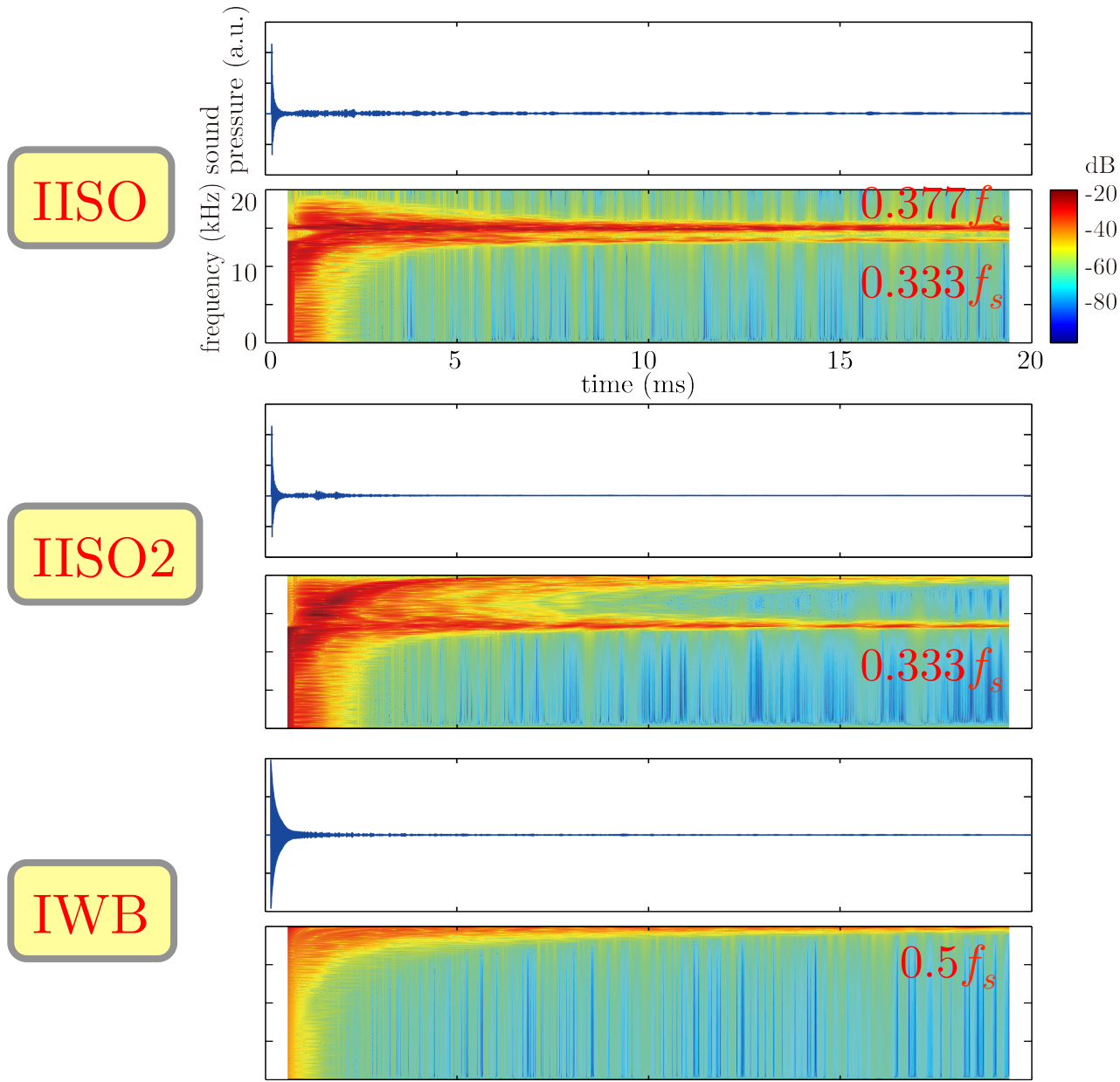
CCP



OCTA



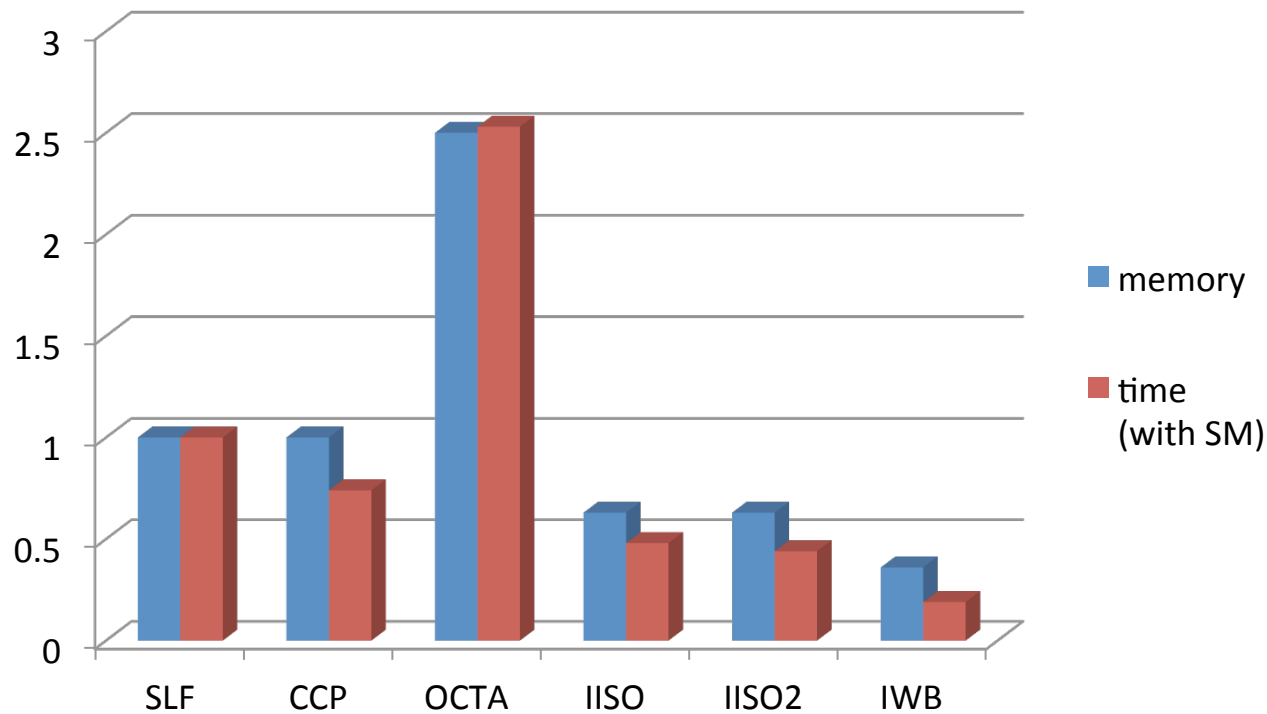
インパルス応答



音質を考慮した場合の計算時間比較

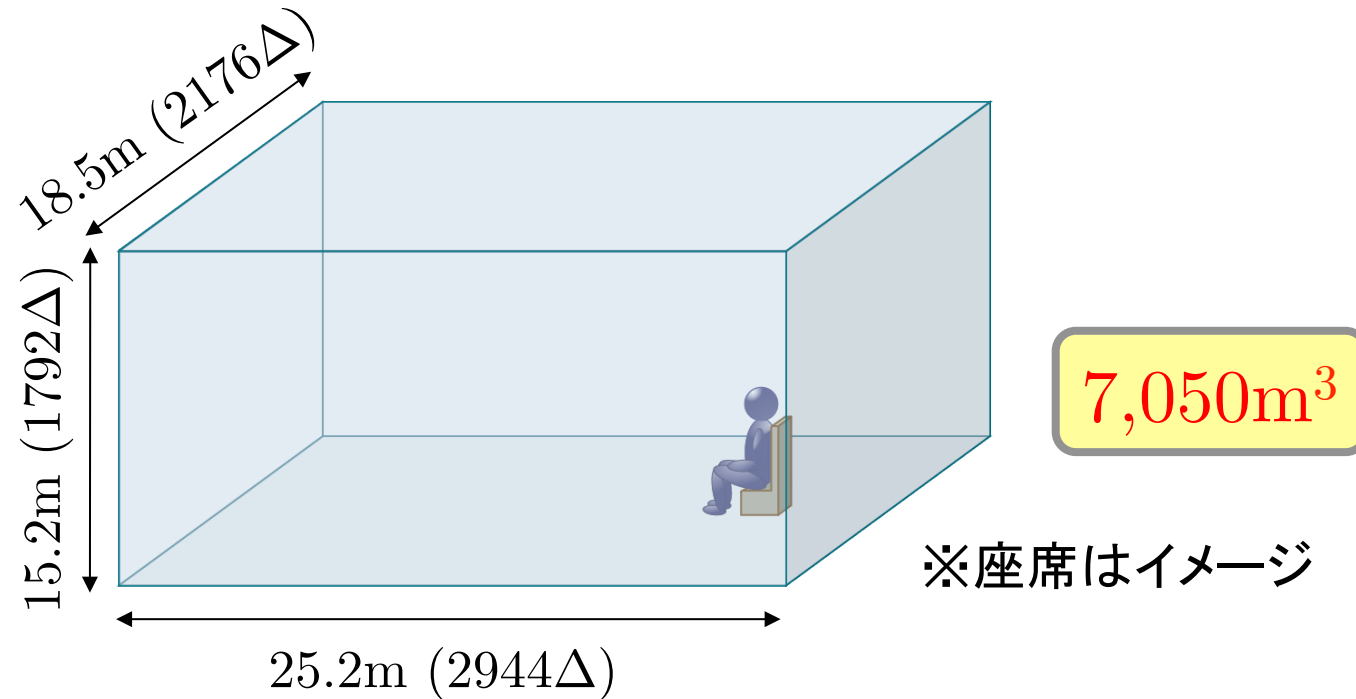


- $f_b=20$ kHz を確保する場合 (SLFを1)
- GPUクラスタ



IWBの計算時間はSLFの19%, メモリは36%

レンダリングモデル



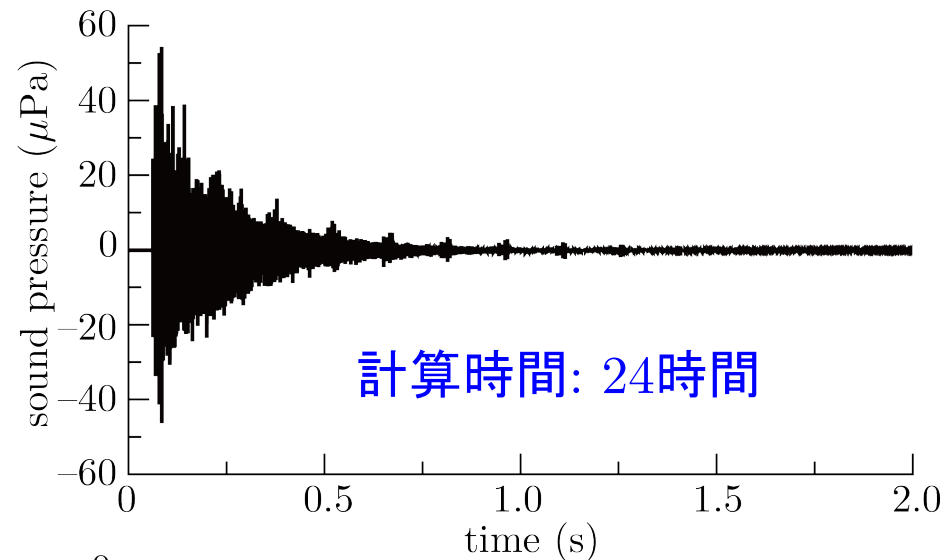
- 媒質: 空気 ($c_0=340\text{m/s}$)
- 壁の反射率: 0.9
- 格子点数: 1.15×10^{10}
- 使用メモリ: 85GB

レンダリング結果

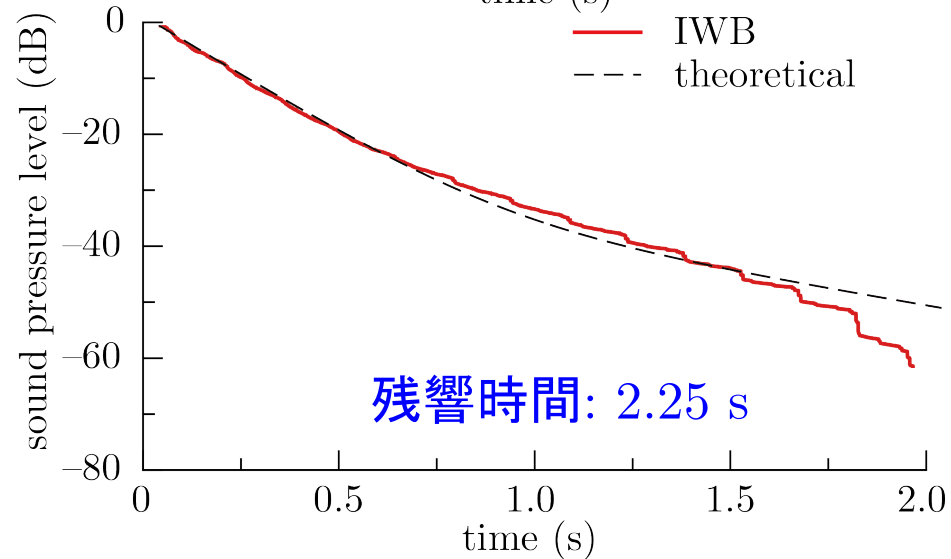


□ステージ中央からインパルスを放射

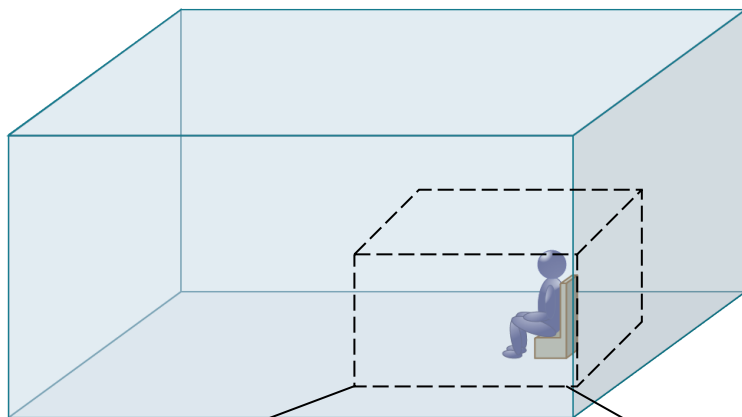
インパルス応答



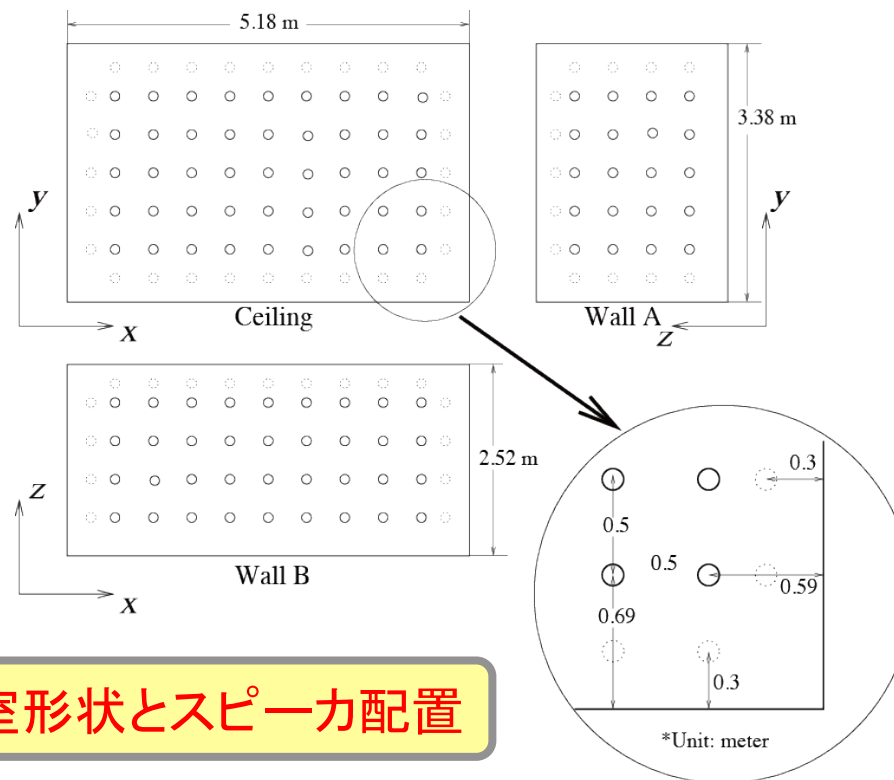
残響曲線



包囲型スピーカアレイ



包囲型スピーカアレイの外観



室形状とスピーカ配置

- ・床を除く5つの面に157個のスピーカを50cm間隔の格子状に配置
- ・4台のPC, 14台のD/Aで48kHzサンプリング, 16bitリニアPCMの157ch同時再生
- ・スピーカ位置に対応する仮想空間内の受音点に外向き法線方向の単一指向性を仮定
- ・パートごとに157点でインパルス応答を計算

音源と音源位置



part	position	part	position
Vn1	(5, 15.5, 1)	Cb	(16.5, 17.5, 1)
Vn2	(5, 19, 1)	Fl, Cl, Tp	(6.5, 22, 1.4)
Va	(13, 19, 1)	Ob, Fg, Hr	(11.5, 22, 1.4)
Vc	(13, 16.5, 1)	Tim	(9, 24, 1.8)

10m

17.9m

ステージ

音源位置

・パート別に無響室録音されたオーケストラ音源(ヘルシンキ大, 48kHz サンプリング)を使用

☆ベートーヴェン交響曲第7番第1楽章冒頭

☆マーラー交響曲第1番「巨人」第4楽章冒頭

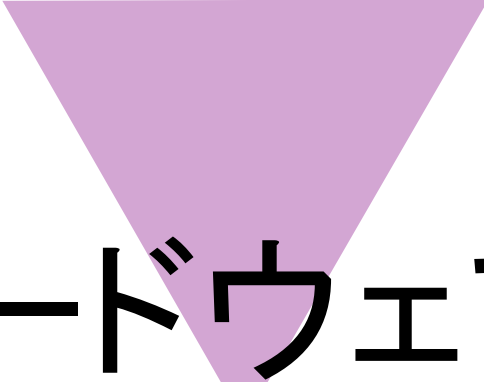
☆モーツァルト歌劇「ドン・ジョバンニ」からドンナ・エルヴィラのアリア

・12パートを8パートに集約し、パートごとに**無指向性音源**と**単一指向性点音源**を仮定、インパルスを放射(8点の音源位置によるインパルス応答を計算)

スピーカアレイでの再生結果



- 自然な残響を伴った立体音場が再現。
- アレイ内のどの位置においてもある程度音像が定位。
同時に複数人の聴取が可能。
- 音源を無指向性とするすると残響が相対的に強く感じられ、
単一指向性とすることで自然な音場が再現。
- ソプラノ独唱では、音源に指向性を付与したことで歌手
が前を向いているか後ろを向いているかを表現可能。
遠近の表現も可能。
- IWB法を用いれば、通常のSLF法よりも高音質で実用的
なレンダリングが可能。
- 客観的評価は今後の課題。
- 波動音響手法を用いて実用的可聴化(7,050m³, 40kHz
サンプリング, 157ch出力)が行えたのは意義深いことで
あり、今後の発展が期待される。



専用ハードウェアによる

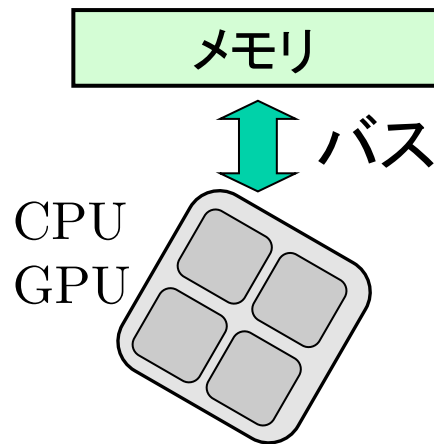
高速化

(FPGA)

メモリとデータ転送

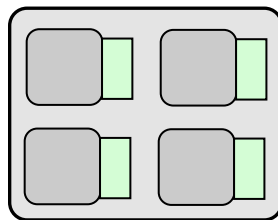


従来の演算方式



- ・並列化のためにコア数が増えれば増えるほど、転送データが大量になり、演算時間よりもデータ転送時間の方が時間がかかってしまう。

メモリ直結方式



- ・演算器とメモリを直結すれば、データ転送に時間がかからなくなる。


FPGAによる専用ハードウェアの設計

※Field Programmable Gate Array



□ 波動方程式を直接中心差分するFDTD (Finite Difference-Time Domain) 法を採用

$$\frac{1}{c_0^2} \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2}$$


$$p_{i,j,k}^{n+1} = 2p_{i,j,k}^n - p_{i,j,k}^{n-1} + \chi^2 \left(p_{i+1,j,k}^n + p_{i-1,j,k}^n + p_{i,j+1,k}^n + p_{i,j-1,k}^n + p_{i,j,k+1}^n + p_{i,j,k-1}^n - 6p_{i,j,k}^n \right)$$



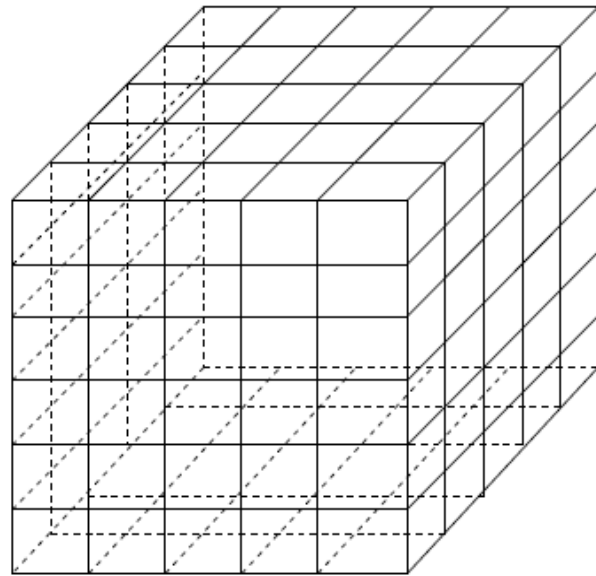
CFL数 $\chi = 1/2$

$$p_{i,j,k}^{n+1} = p_{i,j,k}^n - p_{i,j,k}^{n-1} + \frac{1}{4} \left(p_{i+1,j,k}^n + p_{i-1,j,k}^n + p_{i,j+1,k}^n + p_{i,j-1,k}^n + p_{i,j,k+1}^n + p_{i,j,k-1}^n - 2p_{i,j,k}^n \right)$$

シフト演算のみで実現可能!

- 
- ・回路面積を大幅に削減可能
 - ・演算器数増による並列化

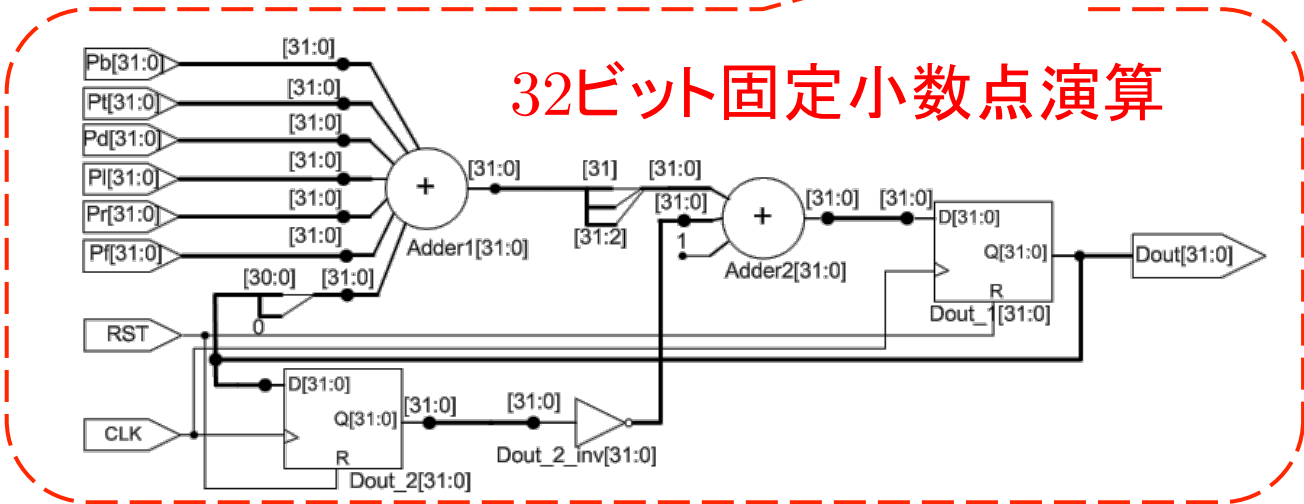
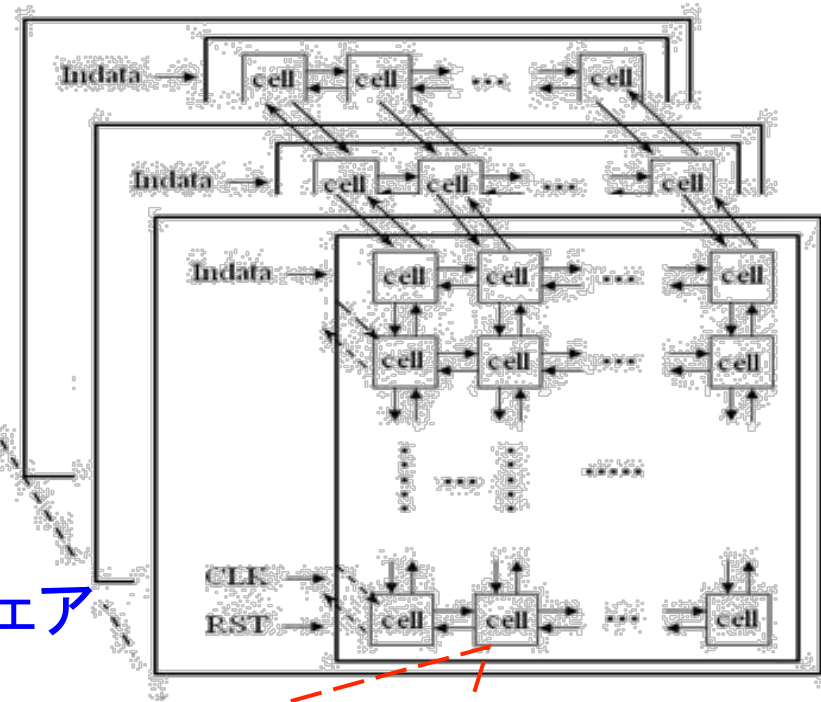
並列アーキテクチャ



3次元音場



ハードウェア構成



計算セル

32ビット固定小数点演算

回路規模と計算性能



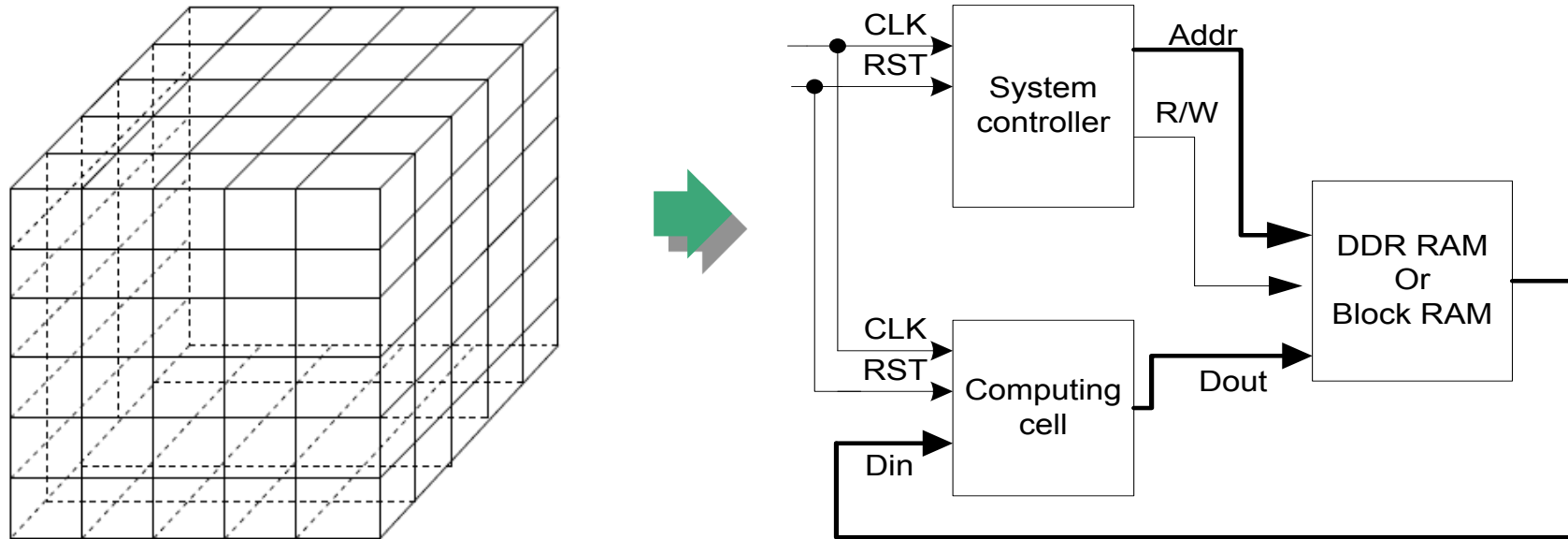
アルゴリズム	スライス	MCUPS
Yee-FDTD	142	73.9
SLF	81	153.7
HO-SLF	58	273.6

Red annotations: A bracket between SLF (81) and HO-SLF (58) is labeled "70%". A bracket between SLF (153.7) and HO-SLF (273.6) is labeled "1.8倍".

スライス: 4つのLUTで1つのスライスを構成。
回路規模を表す。

CUPS: Cells Updated Per Second

時分割アーキテクチャ

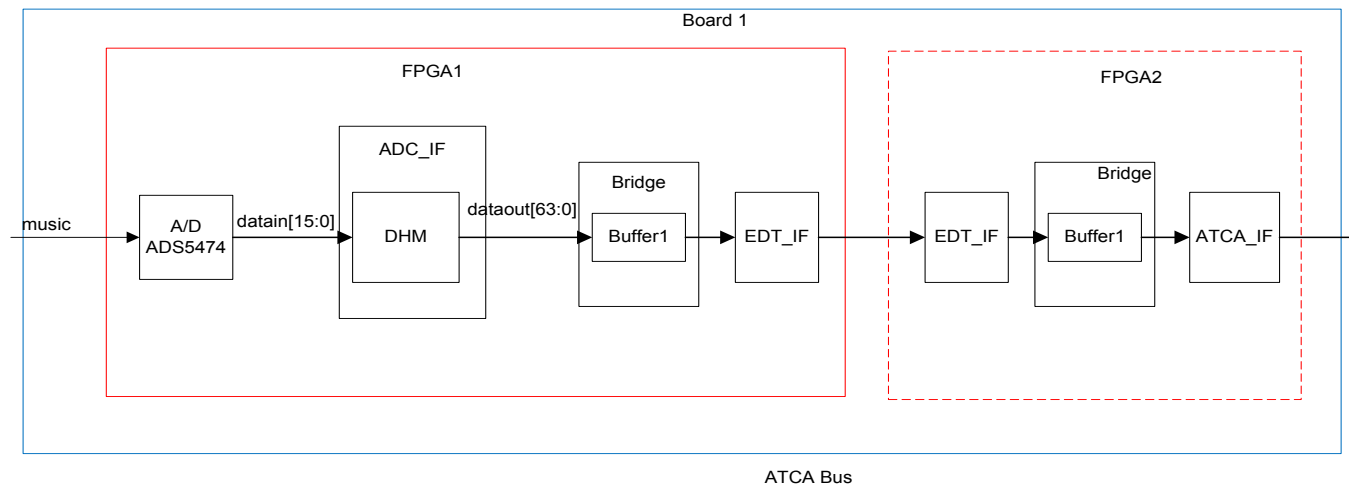


FPGA内には1,700個しか計算セルが実装できないので、3次元場の場合はメモリを計算セル外に配置して、CPUやGPUのような時分割処理を行う。

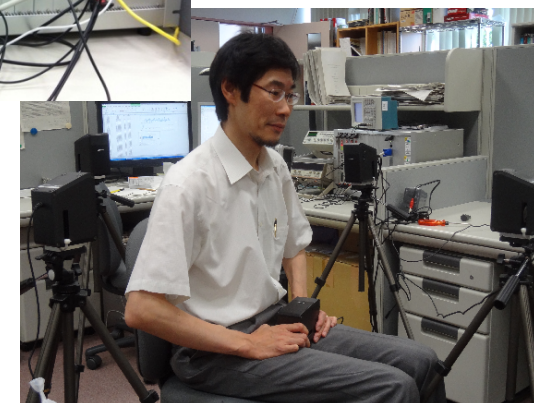
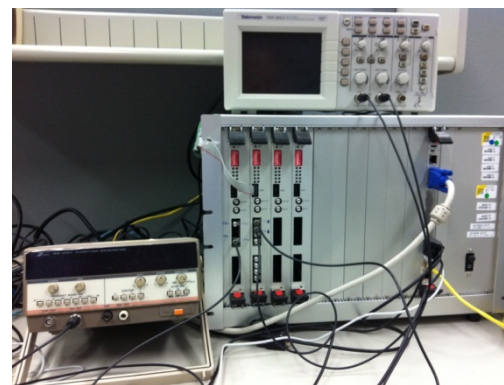
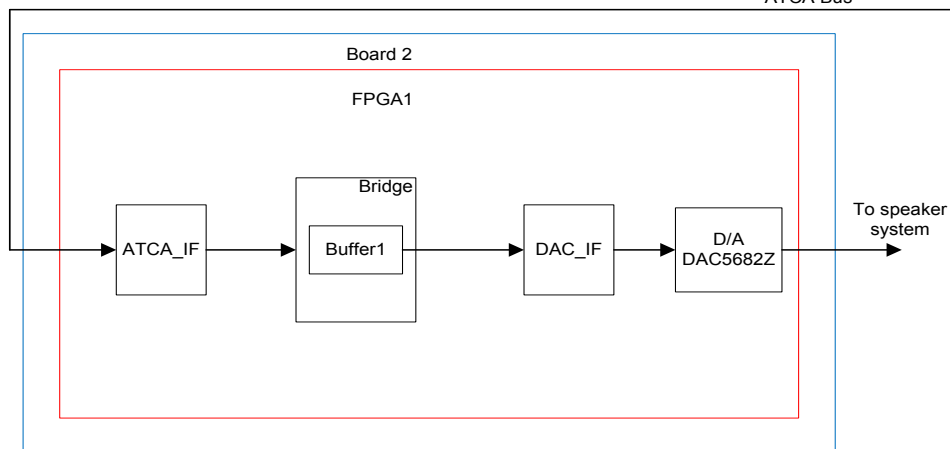
開発されたFPGAリアルタイムレンダラー



◆ FPGAボード上での実装



Xilinx
XC5VLX330T



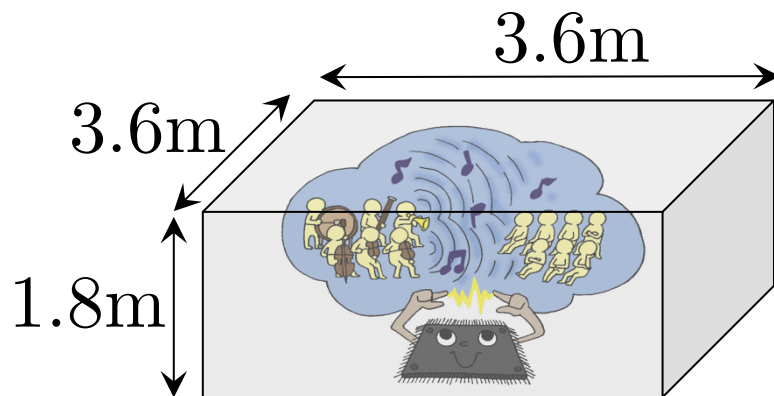
実装のブロック図

東京エレクトロデバイス
SPP3000

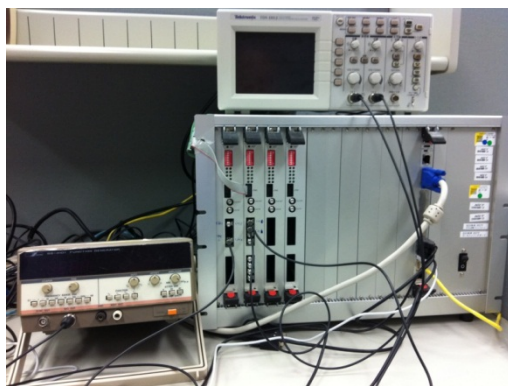
リアルタイムレンダラーの仕様



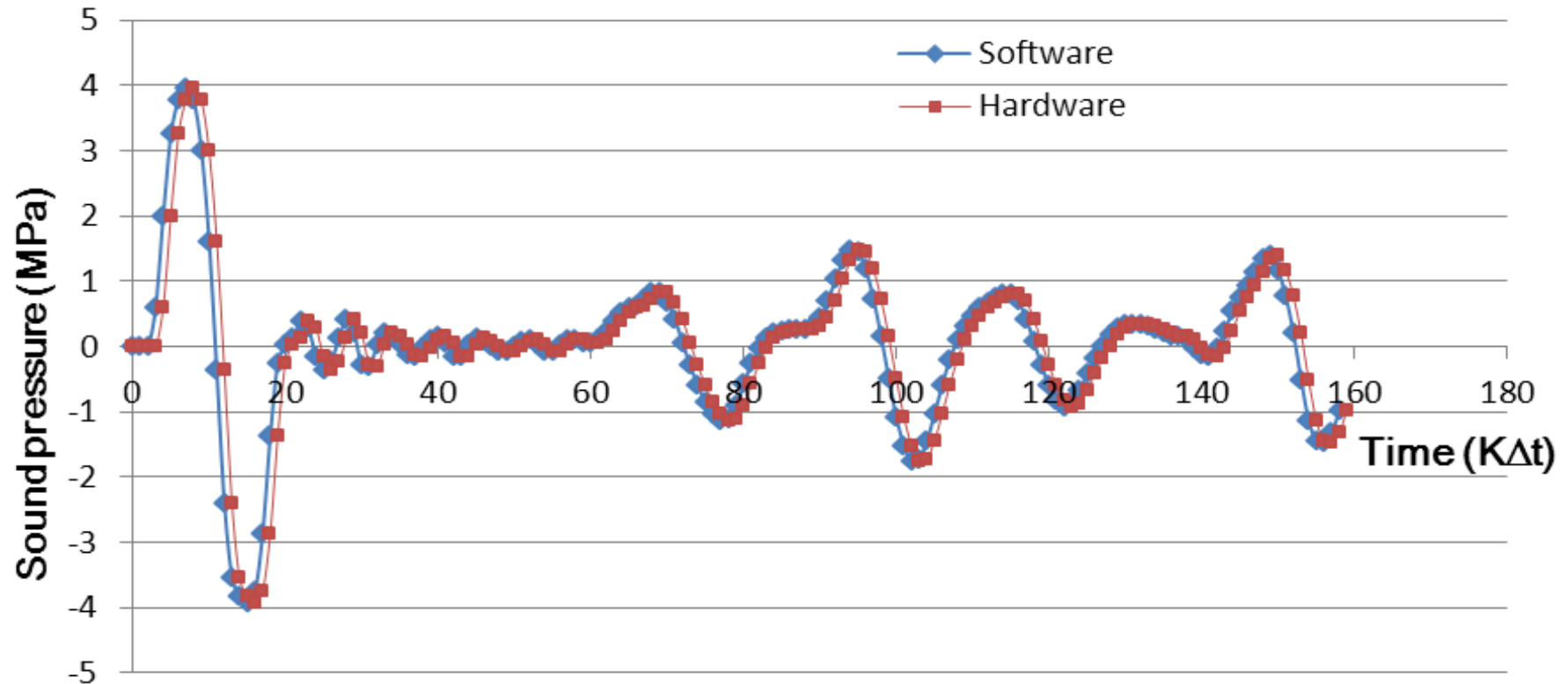
- 仕様対象空間: $1.8 \times 3.6 \times 3.6\text{m}$ の直方体領域
(23.3m^3)
- 上限周波数: 音声周波数帯域(3kHz)
 - ・ただし, サンプルング周波数 6kHz



- ・ $f_s = 6\text{kHz}$
- ・グリッドサイズ: 113mm
- ・ $16,384\text{k points}$



計算結果



2000ステップの計算時間

FPGA: 6.55 s, CPU: 10.01 s

クロック周波数

FPGA: 200MHz, CPU: 2GHz



実質15倍の
速度向上



新たな計算手法の試み

(音響LGA法)

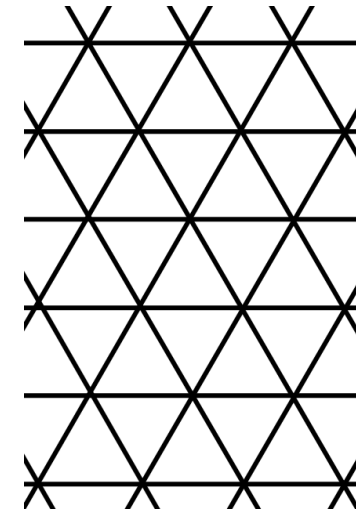
LGA法 (格子ガスオートマトン)



□ 流体解析分野で考案 [Frisch *et al.*, '87]

□ アルゴリズム:

- 空間領域を多角形グリッドで離散化
- 格子点上にランダムに粒子を配置
- 粒子は, 衝突規則に従って, 隣接する格子点に移動
- 空間的・時間的平均によりマクロな流体特性を算出



□ 特徴:

- 格子点間の演算が独立
 - 論理演算のみで記述可能
- 原理的に並列化に適しており, 数値的にも安定

音波伝搬シミュレーションに適用 → 音場LGA法

音場LGA法に必要な演算

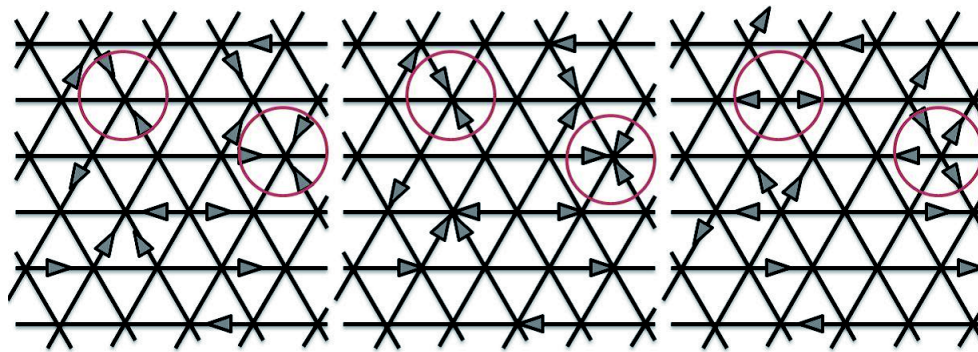


□ 格子点における粒子衝突演算

$$n_i(\mathbf{x} + \mathbf{c}_i, t + 1) = n_i(\mathbf{x}, t) + \Delta_i, \quad i = 1, \dots, 6$$

論理演算のみで記述

□ 粒子の衝突規則

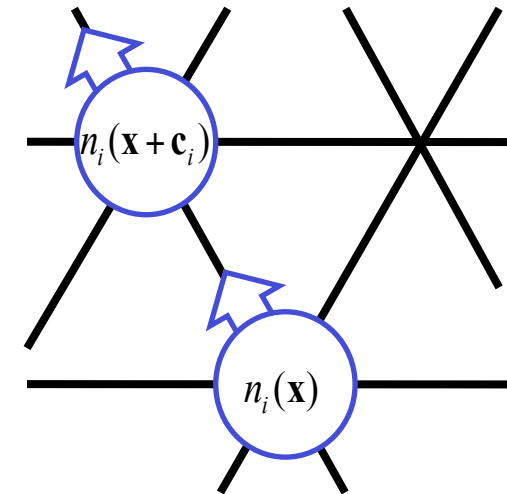


衝突前

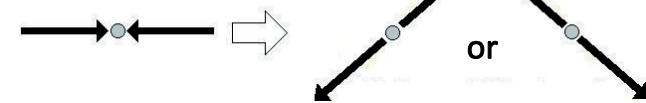
衝突

衝突後(散乱)

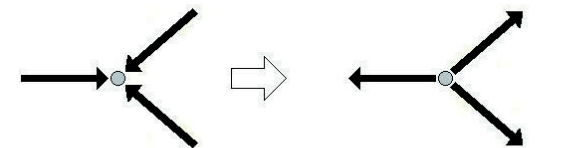
単純なルール



Binary collision (3 patterns)



Trinary collision (2 patterns)



(a) Collision

(b) Scattered

FHPモデルにおける2体及び3体衝突規則

3次元音響 LGA 法による音場解析



□ 正弦波音波の球面放射を再現

- FCHC モデル (Face Centered HyperCubic) を採用
- $1024 \times 1024 \times 1024$ 格子 (約 $0.34 \times 0.34 \times 0.34$ m) の立方領域を解析
 - ・ 隣接格子点間距離 $r \approx 332 \times 10^{-6}$ m (補正係数 $\alpha = 2$) (図の赤矢印部分の格子)
- 音源信号: 8 kHz 標本化正弦波
 - ・ アップサンプリング: 2^8 倍
- 10 ms の伝搬計算

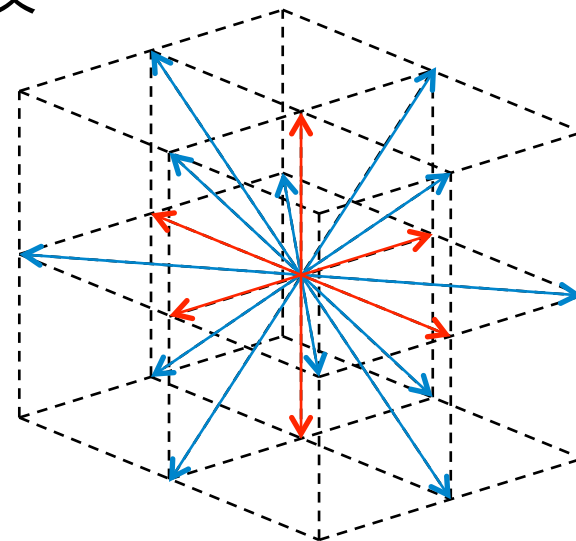
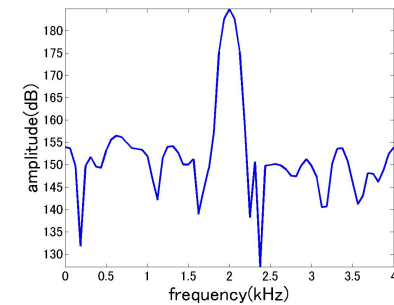
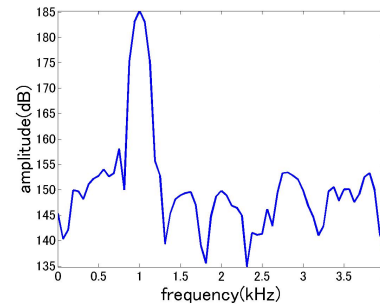
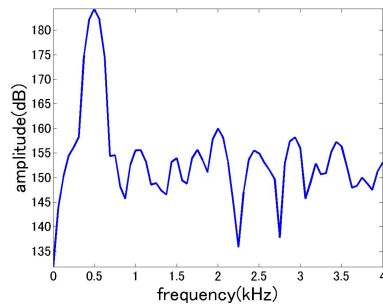
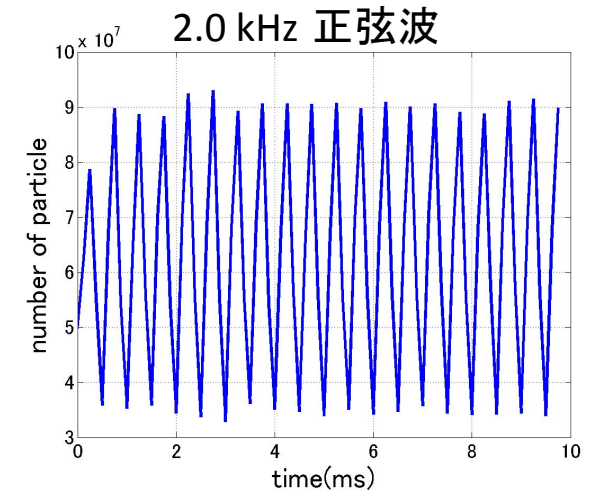
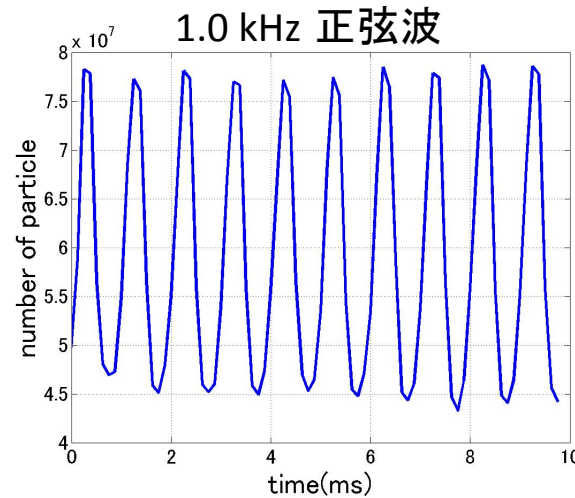
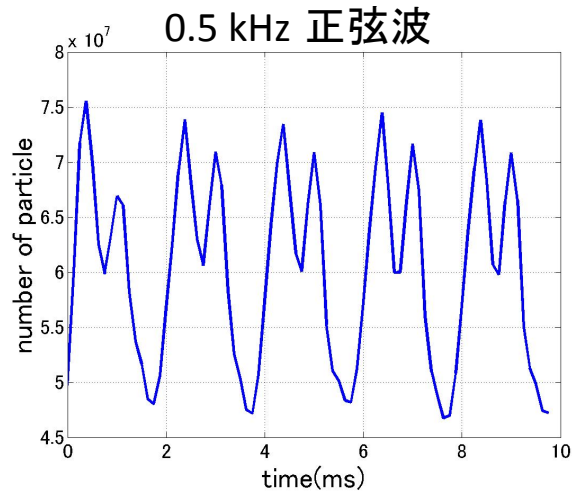


図: FCHC モデル (Face Centered HyperCubic)
(4次元超面心立方格子の3次元射影) 34

3次元音場解析結果



□ 0.25, 0.5, 1.0, 2.0, 3.0 kHz の正弦波放射を再現



- 正弦波の概形を確認
- 伝搬波形のひずみ大きい

シリコンコンサートホール



実現は20～30年後？

それまでは要素
技術の開発

プロジェクトの今後



- 本研究で開発された音空間レンダリング技術は、立体的な音響空間を自由に創出できるため、今後音響分野のみならず、立体映像と融合することでバーチャルリアリティーの世界を広げていくものと期待。
- 2020年に開始される8Kハイビジョンの本放送における22.2chマルチチャンネルオーディオ放送に対する本技術の応用も期待。

デモンストレーション(3号館)



□ GPUレンダリングと8chスピーカアレイシステムによるホール音響の実演

