(MIRU2005) 2005 7

Interval-tree

† † ‡ † † †

†

‡

E-mail: ttakahashi@murase.m.is.nagoya-u.ac.jp

Face-Centered Cubic
Interval-tree

CT
Interval-tree

Interval-tree

A Speedup Method for Isosurfacing Volumetric Data Sampled with a Face-Centered Cubic Lattice

An Approach using Interval-Trees to Speedup of Active-Cell Search

T. Takahashi[†] I. Ide[†] Y. Mekada[‡] H. Murase[†] and T. Yonekura[† †]

[†] Graduate School of Information Science, Nagoya University, Japan
[‡] Life System and Technology, Cyukyo University, Japan
E-mail: ttakahashi@murase.m.is.nagoya-u.ac.jp

Abstract We propose a speedup method for isosurfacing volumetric data sampled with a Face-Centered Cubic (FCC) lattice in this report. This reduces the whole cost by using interval-trees for speedup of searching volumetric data for active-cells. We generated isosurfaces and measured the times of active-cell search and of Isosurfacing while changing thresholds sequentially. Two sets of volumetric data were prepared for the experiments. One of those was generated by metaball-functions and another was re-sampled from medical CT data. The experimental results demonstrated the effectiveness of our speedup method.

Keyword Face-Centered Cubic lattice Volumetric data Isosurfacing Speedup Interval-tree Active-cell search

1.

3

3

3

CT

3

CT

Marching Cubes MC [1]

[2,3] MC

3

MC

FCC

[4-9]

FCC

1 FCC

FCC

[6]

[6,8] [7,8]

CT [9]

[10-13]

2

1

[10]

[11-13]

FCC

Interval-tree

[11] FCC

Interval-tree

2

Interval-tree FCC

FCC

Interval-tree

FCC

FCC [6-9]

2 MC

FCC

3

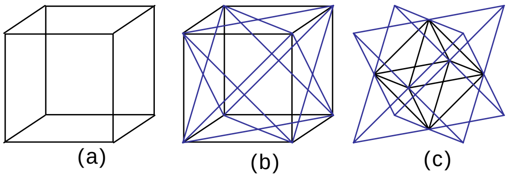
FCC

FCC

FCC

Interval-tree 4

5



1 **(a)** **8**
(b) **(c)** **FCC**
(a) **1**

2.

2.1. MC 法

MC

8

2

internal point

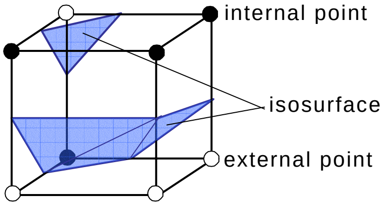
external point

2

256

22

MC [1]



2 MC

2.2. FCC 法

FCC FCC 6

8 4

4 2

1(c)

3

3 8 4

FCC

1 8 8 4 9

14

3 8

6 8 6

64

6 MC

8

4

FCC

[6-9]

2.3. 境界セル探索の高速化手法

2

[10]

[11-13]

1

8 8

[12]

FCC

2

2 2

[11,13]

Interval-tree 2 [11]

Interval-tree Interval-tree

i i.min i.max

J $S = \{ i_j \}\ (j = 0,1,2,...,\mathrm{J}-1)$

Interval-tree Interval-tree n

n.value n.value

n.min_sorted_list

n.max_sorted_list 2

n.value S

n n.left_node

```
< n.right_node   , ß9Ú
  n :  root; S :  /i_j ;  j    0,1,2,...,J  1
  make_interval_tree S, n
    n :  new_node  ;
    n.value :  median S  ;
    n.left_node :  NULL;   n.right_node :  NULL;
    S_Boundary :  3;   S_Upper :  3;   S_Lower :  3;
    j :  0;
    while  i_j  • S
      if  i_j.max   n.value  then   S_Lower :  S_Lower %o/i_j ;
      elseif  i_j.min  t n.value  then   S_Upper :  S_Upper %o/i_j ;
      else   S_Boundary :  S_Boundary %o/i_j ;
      j    ;
    n.min_sorted_list :  sort_min_inc  S_Boundary ;
    n.max_sorted_list :  sort_max_dec  S_Boundary ;
    if   S_Lower  z 3  then
      make_interval_tree S_Lower, n.left_node ;
    if   S_Upper  z 3  then
      make_interval_tree S_Upper, n.right_node;
```

9è &•9ê › ç é o ; Ç Ã Ú ã È9Ø2 ½ Á •&• ã & A o < Ç ; , Ô 2 ß é] , ;] é'" »/Û O -„ '"› æ Ñ!ª Ô Î ã æ 9Ø ; Ç = ù ™4F é4ø ' S_T 7 1f æ &• Ô 9Ú

```
  n :  root;  S_T :  3;
  find_intervals  S_T,  T, n
    if   T   n.value  then
      j :  0;
      while  i_j  • n.min_sorted_list  &  i_j.min   T
        S_T :  S_T %o/i_j  ;   j    ;
      if   n.left_node z NULL   then
        find_intervals  S_T,  T, n.left_node ;
    else if   T  ! n.value   then
      j :  0;
      while  i_j  • n.max_sorted_list   &   i_j.max  t  T
        S_T :  S_T %o/i_j  ;   j    ;
      if   n.right_node z NULL   then
        find_intervals  S_T,  T, n.right_node  ;
    else
      j :  0;
      while  i_j  • n.min_sorted_list
        S_T :  S_T %o/i_j  ;   j    ;
```

Interval-tree é t%½9Ø &• æ -Ð å ± Ž ê9Ø ™4F é ¨ " o ã ¨ Ê o é ú â ½ Ç9Ø Î º!Ç . ^ •&• æ!ª ¿ • '9Ø ½ . ^ Ç •&• Ð Ú ã È9Ø Ø é . ^ æ » Ô %l o5Y D 6 4!¢ ' - Á Ú ü æ Ø é . ^ é &5Â P é Ï'Î ÿ o é ± Ž ç Ô -Ð Ç ½ 9Ú Î é ± Ž K á Interval-tree æ ; , Ô Î ã ê T U] & , ; é-ð P Æ b ù Ò Ê å ¿9Ú FCC : é • '9Ø • é 2 ß é H/° Ñ!ª Ò9Ø •&• Ð Ú . ^ é c t%½ - Á Î ã â9Ø T U] & , ; é å ë Ç (» ã Ò á ¿ [11]9Ú

n ¡ Ä {$è4F æ Å ¿ á9Ø . ^ •-+ Ô 5Â P é Ï'Î Æ 9Ø"ç »"› æ • é5Â P é Ï'Î Ç ÿ o (» â ½

n ¡ Ä {$è4F ã T U]$è4F é Ï'Î"› å » ƒ Ì Ç ˆ æ (» â ½

FCC : é • '9Ø è(* é H/° ê) Ú Ò á ¿ ý é é9Ø ß (* æ4P Ò á ê. ø Ð á ¿ å ¿9Ú Ø Î â Ä$Â â ê9Ø FCC » : æ » Ò á Interval-tree !ª ¿ Ú7 1f { - Á æ ½ Ú 9Ø ß(* é H/°) Ú Ô Á å FCC O] X o S : o 2 é 2 ¸"› å T U]2)'Î é s û . Ò9Ø ƒ "› å T U] & , ; é å ë Ñ Þ Ú9Ú

3. - z » : é ^ '] - S

3.1. FCC B—B¥B B·B›B,B·BzB1BœB•B¥2)'Î

1b ñ é% Ç p 6 é O] X o S : o 2 Ç 3 4 02) É æ ; , Ð á ¿ ã Ò 9 Ø 2) É é &-Ð G i,j,k (i 0,1,..,I 1`9Ø j 0,1,..,J 1`9Ø k 0,1,..,K 1`) â-+ Ô9Ú O] X o S : o 2 Ç | (1) æ$ Ô Á æ T U]$è4F] â1j&ï Ò á ; , Ð á ¿ Î ã è - ã Ô 9Ú Î Î â p ê G i,j,k Ç ; , Ð á ¿ T U]$è4F â é Ï'Î -+ Ô9Ú ¡ Ä {$è4F2é ß é p 6 P é"ç » Ï'Î

'$_x$, '$_y$, '$_z$ â-+ Ò9Ø Ø é9Þ P é o Ç ; , Ð á ¿ T U]$è4F â é"ç » Ï'Î '$_p$ â-+ Ô ã È9Ø (2)9Ø(3) !ª ¿ á9Ø ¡ Ä {$è4F ã T U]$è4F é Ï'Î"› å » ƒ Ì Ç ˆ æ- Ã 9Ú

$$p \quad lJk \quad l j \quad i \qquad (1)$$

$$'_p \quad lJ\frac{'_z}{d} \quad l\frac{'_y}{h} \quad \frac{'_x}{w} \qquad (2)$$

$$\begin{pmatrix} '_x \\ '_y \\ '_z \end{pmatrix} \quad \begin{pmatrix} '_p \%l\, w \\ \left\lfloor \dfrac{'_p}{l} \right\rfloor \%h \\ \left\lfloor \dfrac{'_p}{lJ} \right\rfloor d \end{pmatrix} \qquad (3)$$

Î Î â w9Ø h9Ø d Ø Ù x0² Ç 09Ø y0² Ç 09Ø z0² Ç 0 é ¡ Ä {4F4Ò ã Ò9Ø Ç 09Ø j Ç 09Ø k Ç 0 Ç9Ø Ø Ù x0²9Ø y0²9Ø z0² ã2€ å ý é ã Ô 9Ú

.% é Á å » ƒ Ì ˆ æ- Á Ú ü9Ø FCC O] X o S : o 2 é T U]2)'Î • é Á æ . Ò Ú9Ú 9è ¡ Ä {9ê : o 2 » | ã Ò á | (1) é Á æ T U]$è4F] â1j&ï Ò á ; , Ð Ú 3 4 02) É !ª ¿ 9Ú2) É é › ç é-Ð&›

$$F_{CC}(i,j,k)=\begin{array}{ll} f(\ iw,\ jh,\ kd\ ) & k\ \text{is even.} \\ f\quad i+\dfrac{1}{2}\ w,\quad j+\dfrac{1}{2}\ h,\ kd & k\ \text{is odd.} \end{array} \qquad (4)$$

1

(5)

FCC

3 $(i,j,k)=(0,0,0)$

$$w:h:d=1:1:\sqrt{2}/2 \qquad (5)$$

(7)

Interval-tree

Interval-tree 1

θ

FCC

FCC Interval-tree

2 $(\Delta_x,\Delta_y,\Delta_z)$

Interval-tree

θ S_θ S_θ

k

k

S_θ

Δ_{p_even}

Δ_{p_odd} (6) Δ_z k

13

Δ_k

(7) (8)

(7) (9)

$$\Delta_k=\frac{\Delta_z}{d}=\frac{\Delta_{p\_even}}{\text{IJ}}=\frac{\Delta_{p\_odd}}{\text{IJ}} \qquad (6)$$

8

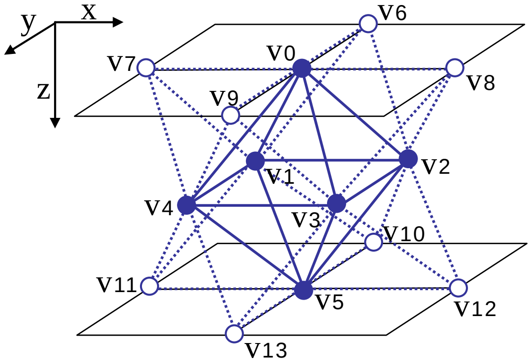
$$\Delta_{p\_even}=\begin{array}{ll} \text{IJ}\dfrac{\Delta_z}{d}+\text{I}\dfrac{\Delta_y}{h}+\dfrac{\Delta_x}{w} & \Delta_k\ \text{is even.} \\ \text{IJ}\dfrac{\Delta_z}{d}+\text{I}\ \ \dfrac{\Delta_y}{h}-\dfrac{1}{2}\ \ +\dfrac{\Delta_x}{w}-\dfrac{1}{2} & \Delta_k\ \text{is } odd. \end{array} \qquad (7)$$

[6-9]

$$\Delta_{p\_odd}=\begin{array}{ll} \text{IJ}\dfrac{\Delta_z}{d}+\text{I}\dfrac{\Delta_y}{h}+\dfrac{\Delta_x}{w} & \Delta_k\ \text{is even.} \\ \text{IJ}\dfrac{\Delta_z}{d}+\text{I}\ \ \dfrac{\Delta_y}{h}+\dfrac{1}{2}\ \ +\dfrac{\Delta_x}{w}+\dfrac{1}{2} & \Delta_k\ \text{is } odd. \end{array} \qquad (8)$$

$$\begin{array}{lll} (\Delta_{p\_even}\,\%\,\text{I})w, & (\Delta_{p\_odd}\,\%\,\text{I})w & \Delta_k\ \text{is even.} \\ \Delta_{p\_even}\,\%\,\text{I}+\dfrac{1}{2}\ \ w, & \Delta_{p\_odd}\,\%\,\text{I}-\dfrac{1}{2}\ \ w & \Delta_k\ \text{is odd.} \end{array}$$

$$\begin{array}{l} \Delta_x \\ \Delta_y= \\ \Delta_z \end{array} \begin{array}{lll} \dfrac{\Delta_{p\_even}}{\text{I}}\ h, & \dfrac{\Delta_{p\_odd}}{\text{I}}\ h & \Delta_k\text{is even.} \\ \dfrac{\Delta_{p\_even}}{\text{I}}\ \ +\dfrac{1}{2}\ h, & \dfrac{\Delta_{p\_odd}}{\text{I}}\ \ -\dfrac{1}{2}\ h & \Delta_k\ \text{is odd.} \\ \dfrac{\Delta_{p\_even}}{\text{IJ}}\ d, & \dfrac{\Delta_{p\_odd}}{\text{IJ}}\ d & \end{array} \qquad (9)$$



3 FCC 1 8 8 4 V0 V13 14 8 6 V0 V5

3.2. Interval-tree FCC

Interval-tree FCC

8 6

4.

4.1. 実験方法

200 FCC

(

$(\mathrm{I,J,K}) = (128,128,181)$ $w:h:d = 1:1:\sqrt{2}/2$

2841804) CT FCC

($(\mathrm{I,J,K}) = (128,128,244)$

$(w,h,d) = (2.5,2.5,2.0)[mm]$ 3841992)

FCC

0

65535 2

x y z 2

θ 0.0 65535.0

Interval-tree

Interval-tree θ

CPU: Intel Xeon 3.0GHz, L2 : 512KB,

: 2.0GB

4.2. 結果と考察

Interval-Tree

1,312msec CT 1,610msec

θ with

Interval-tree without Interval-tree

4,5

active-cell

search isosurfacing

6,7

8 8 (a)

(f) θ

(g) (i) CT

(g) (h)

(i)

θ

(h) (i)

Interval-Tree

Interval-tree

4,5

(6,7)

n b

$O(n)$

Interval-tree

2

$O(b+\log n)$

$b \ll n$

Interval-tree

2

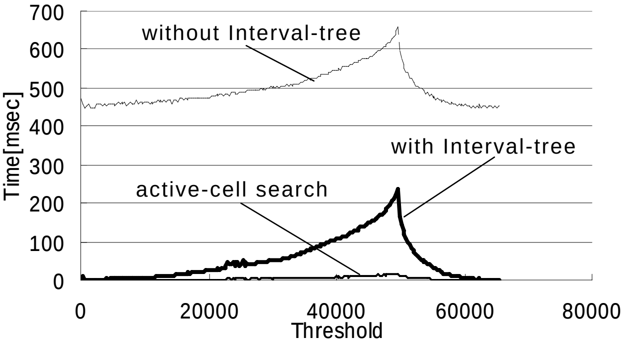
n i

$2ni$

i

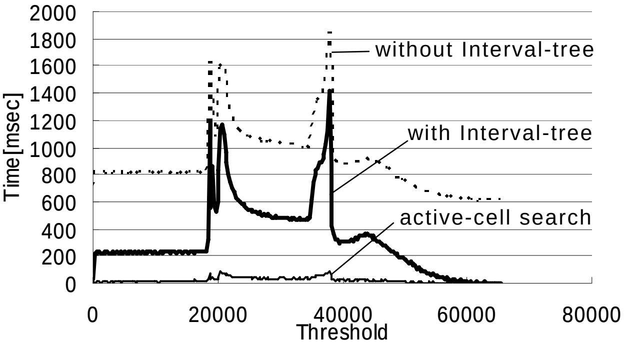
FCC

Interval-tree



4

(with Interval-tree) (active-cell search)



5 CT

(with Interval-tree) (active-cell search)

6

(Isosurfacing)

(active-cell search)

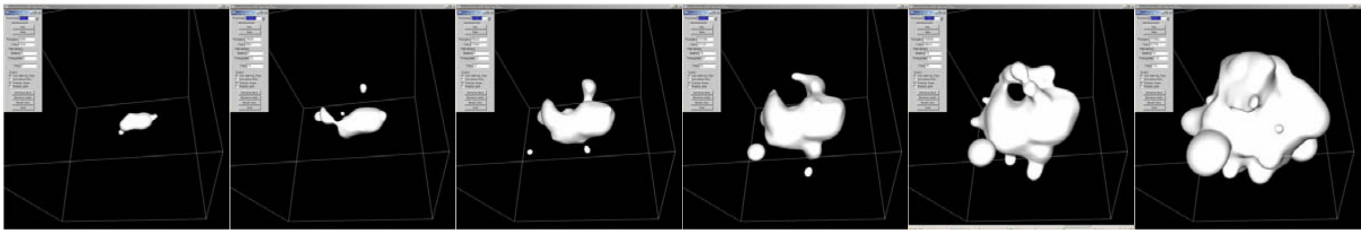
7 CT

(Isosurfacing)

(active-cell search)

5.

FCC



(a) θ =7500.0 (9,196) (b) θ =15000.0 (30,144) (c) θ =22500.0 (71,396) (d) θ =30000.0 (107,000) (e) θ =37500.0 (195,168) (f) θ =45000.0 (316,480)

(g) θ =40000.0 (650,100) (h) θ =32000.0 (1,002,792) (i) θ =37888.0 (3,037,672)

8 **(a)** **(f)** **(g)** **(i)** **CT** θ **3**