Improved marching cubes using novel adjacent lookup table and random sampling for medical object-specific 3D visualization

Xuchu Wang[a], Yanmin Niu[b,a], Li-wen Tan[c], Shao-Xiang Zhang[c]
[a] Key Laboratory of Optoelectronic Technology and Systems of Ministry of Education, College of Optoelectronic Engineering, Chongqing University,Chongqing 400044, P.R.China
Email: xcwang@cqu.edu.cn
[b] College of Computer and Information Science, Chongqing Normal University, Chongqing 400050, P.R.China
[c] Department of Anatomy, College of Basic Medical Science, Third Military Medical University, Chongqing 400038, China

Abstract— **The marching cubes (MC) algorithm is a widely used routine to extract isosurfaces from volumetric data set. This method suffers from the exhaustive accessing cubes sequentially and the failure of directly separating the isosurfaces. To address this, an improved MC algorithm called AlutMC is proposed to track connected surfaces by combining a new adjacent lookup table and random sampling technique. AlutMC has a natural ability to separate the isosurfaces related to different components, which can efficiently segment 3D objects without extra expenditure and helpful for visualizing medical objects. Experimental results on three test data sets show the computational efficiency in identifying isosurfaces. Additionally, it is simple and efficient, not requiring complex data structure, and can be used as is into the MC and its improvements because it is quite cost-effective and self-contained.**

Index Terms— **Surface rendering, Isosurface extraction, Marching cubes, Lookup table, Isosurface separation, 3D object segmentation**

I. Introduction

In the past two decades, surface rendering has become an important technique for extracting meaningful and intuitive information from volumetric scalar data sets in the fields of computer graphics, computer vision, geometric modeling and animation [1], [2]. It usually consists of two steps named surface generation and surface rendering. Compared to the surface rendering step that adopts conventional computer graphics techniques to render the surface and takes full advantage of the graphics display hardware acceleration function, the surface generation step that converts the volume data into a surface representation is more essential to preparation for visualization, since it heavily depends on the constant density surface (isosurface) extraction methods [3].

The well-known isosurface extraction algorithm is marching cubes (MC) that follows a straightforward, practical approach to explicitly represent the implicit isosurface by polygons (typically, triangles set) [4]. MC essentially takes the "division and conquer" principle to divide the volumetric data set into cells and then generate triangles to approximate isosurfaces within each cell. The final surface representation is a cell-by-cell link of the within-cell isosurfaces. Based on it, successive improved versions have completed the various configurations of the original look-up table in order to obtain more accurate models with correct topology, and it also can be extended to compute and visualize fuzzy equivalents to crisp isocontours in uncertain scalar fields [5]. For more details, the reader is referred to a review [6]. However, MC and its improvements have a real-time and object of interest unawareness problem because it essentially need traverse each cell in the procedure of triangles generation on the large scale data sets, and the isosurfaces of all objects are equally extracted. This problem even becomes increasingly serious along with the increasingly massive volumetric data sets produced by recent technical breakthroughs in new imaging modalities such as CT, MRI, and other 3-D scanning technologies [7], [8].

To address this problem, some methods for reducing calculations have been explored to enhance triangle representations or to obtain more efficient implementations. The typical strategy includes data structure reorganization, division - and - conquer, extended lookup table and region growing, etc. Cui and Liu [9] proposed a simplified triangulation patterns for isosurface extraction by moving the positions of isosurface vertices from the cube edge interpolation points as in the standard MC to the centers of the corresponding occupied cube vertices. This method reduces the 254 configuration cases to 178 case while with comparable rendering quality. Likewise, Matsopoulos el al. [10] proposed to generate the polygons in the 15 predefined cases of the standard MC as well as additional cases by a generic rule. Their method orders the isosurface points directly in polygons rather than triangles, thus produces less triangles. It also can avoid the type A "hole problem" that occurs in the standard MC. Dyken et al. [11] proposed a high-speed MC method that reformulates the MC as a data compaction and expansion process and outfits the histogram pyramid on graphics hardware

Manuscript received May *, 2013; revised *, *; accepted * *, *.

doi:10.4304/jsw.9.10.2528-2537

for OpenGL 2.0 or comparable graphics APIs, where the histogram pyramid is an efficient hierarchical data structure recently introduced in GPU programming.

The category of extended lookup table can both speed up the case localization and reduce the degenerate triangles. For example, Lewiner et al. [12] proposed a single entry cubical lookup table to represent the 730 subcases of the enhanced lookup table in Chernyaev's MC 33 algorithm [13], which allows a topologically correct manifold surface with no crack, through the trilinear interpolation of the scalar field over each cube. Raman and Wenger [14] proposed an extended 3^8-entry MC lookup table that differentiates scalar values equal to the isovalue from scalar values greater than the isovalue. Their proposed method can avoid the degenerate triangles or any small areas, edges or angles. Also by designing lookup tables, Dietrich et al. [15], [16] proposed to generate better-shaped triangles by building a single one from eight different edge groups to identify the equivalent triples of edges. The re-triangulation in certain table entries to prevent this edge group from occurring focuses on only a few MC cases, namely case 5, 12, 11, and the complement of case 6. As an alternative approach, instead of removing edge groups entirely from the MC table, they turned to add an additional vertex in the cell' center and connect it to the intersection vertices of active edges, making it a more practical choice for isosurface grid generation [17].

Xi and Duan [18] proposed a new region-growing based isosurface extraction algorithm using marching triangles that can generate high-quality curvature-adaptive semi-regular meshes and preserve sharp features. The region-growing idea can also be applied in the cube tracing tasks. The basic support behind this method is the fact that for a specific isovalue only 5% cubes may contain triangles [19], so the extraction speed will be improved if a search strategy can avoid visiting empty cells. Lee and Lin [20] proposed a growing-cube isosurface extraction algorithm for medical volume data. In their method, a surface tracker is designed to avoid exhaustive searching isosurfaces cell-by-cell by introducing an edge-based look-up table to guide which cells to visit from a given cell. The idea of region growing in both methods is helpful for improving the efficiency of isosurface extraction, however, the growing-cube method needs repeatedly visit the look-up table during the isosurface generation, while not considering the storage strategy of cubes. Therefore, there still is a strong demand in practice to develop algorithms capable of improving the visiting speed of cells.

In this paper, we present an improved MC algorithm, namely AlutMC, to reduce visiting the empty cells based on a new adjacent lookup table. Using this lookup table, the AlutMC tracks connected surfaces instead of exhaustive searching isosurfaces cell-by-cell, thus saves computation timing in identifying these isosurfaces. We also present the algorithmic implementation that ensures the AlutMC separate these isosurfaces related to different objects. The extracted isosurface by our AlutMC is identical to those of the MC and the growing-cube method, however, the implementation by AlutMC is more explicit and intensive. The isosurface extraction and disjoint components segmentation experiment on three test data sets validate the effectiveness of the AlutMC in comparison to these previous approaches. Therefore, the proposed AlutMC is quite cost-effective and self-contained.

The paper is structured as follows. Section II presents the proposed AlutMC method in details. Section III describes the experimental results on medical volume data sets and makes some discussions. Section IV concludes the paper and suggests the future work.

II. The AlutMC method

We propose an improved MC algorithm (named AlutMC) based on a single generic rule capable of generating the adjacent cube surface for all predefined cube configurations. The new algorithm generates the resulting adjacent cube surface by defining an adjacent lookup table (A-LUT) in every possible cube configurations, without resorting to any predefined cases. The AlutMC directly searches the unvisited cube based on A-LUT instead of the cell-by-cell visiting strategy in standard MC method, thus produces high visiting efficiency and separation ability via algorithmic implementation. These details is explained below.

A. Adjacent lookup table (A-LUT)

In the standard MC, each cell is equivalently visited. Although some processing involving empty cells is probably inevitable because generating a correct isosurface requires each cell to be visited at least once to judge its activeness (non-empty) property, it is still advisable to accelerate the MC by avoiding unnecessary operations on the non-active cells, since 30-70% of the processing time involves those cells [19]. Therefore, we consider a way to skip the empty cells by analyzing the connected property of two adjacent cells. For each cell that satisfies a configuration of the MC algorithm, there are six adjacent cells, however, not each of them needs isosurface extraction. For example, in the case 3 in the configuration, only the cells along the first, third, fourth, fifth, and sixth directions of the current cell have possible isosurfaces. Furthermore, among 15 cases, those cases 0, 1, 2, 3, 5 and 8 do not need to track six neighboring cells. Yagel et al. [21] pointed out that these six cases account for 90% of the cases encountered in extracting an isosurfaces in their applications. Therefore, the tracking approach leads to more saving in computation if the user specifies a seed cell and then starts searching seed's neighboring cells by the aid of some prior information. To this end, in this paper we propose an adjacent lookup table (A-LUT) to guide which cells to visit from a given cell. To create this table, we label each configuration and six tracking directions for a given cell. The relationship between this boolean A-LUT and the configuration lookup table is

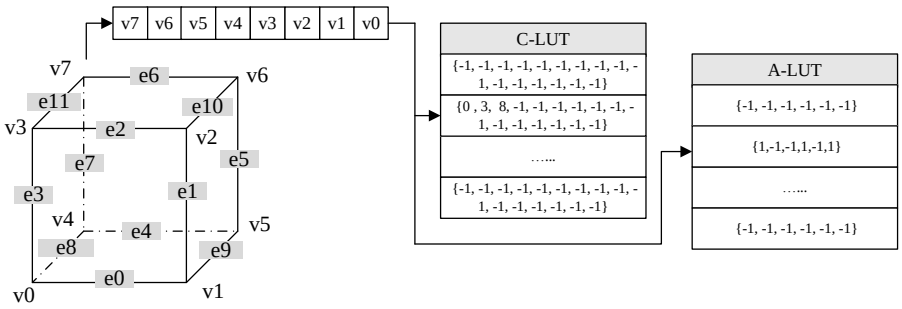


Figure 1. The relationship between the configuration lookup table and the adjacent lookup table.

shown in Figure 1 and a segment of this table is as follows.

$$\text{A-LUT} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 \\ \cdots & \cdots & & & & \\ 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (1)$$

where the ith entry corresponds to the ith case in the 256 possible ways in which a surface can intersect the voxel, and jth column corresponds to the jth face of a cube. 1 mean there are adjacent cube to calculate isosurface along this direction, while -1 means no adjacent cube to visit. For example, the fifth row means there are three adjacent cubes to visit along the second, third, and sixth directions. Different from [22], this table can be rewritten as 96 hex numbers or 6 bytes for storage.

B. Algorithmic implementation

We present the pseudo-code of the proposed AlutMC algorithm in Figure 2, which ensures the separability of disjoint isosurface components in this implementation. The flag $\mathcal{F}$ is an array that records the visiting status of each cube ($\mathcal{F}[p] < 0$ means the pth cube is empty or visited, otherwise unvisited). The random sampling technique is incorporated to avoid the visiting initialization of this array. For real medical image analysis application, the object number can be settled much less than the whole components and the main object of interest can be initialized by the parameter of sampling number per object. The stack $\mathcal{S}$ records the indices of the cubes to be tracked. The queue $\mathcal{Q}$ records all the possible active cubes. In the first "while-do" loop, the major components are to be built. In the upper second "while-do" loop, the queue $\mathcal{Q}$ is built via random sampling. For relative MC based algorithms the traversal is necessary because the activeness of each cell must be firstly judged, however, in our algorithm, this visiting is totally avoided. In the simple case that there is only one single connected object to be extracted, both the loop and the queue in our method can be omitted by presetting of $\mathcal{F}$, however, there is not any change on the basic flowchart of the proposed algorithm. In the lower second "while-do" loop, if the pop-up qth cube is unvisited, we construct its isosurface and change its flag, then collect the indices of its neighboring cubes and push them to the stack if they have not been visited. Later these cubes can potentially start a new growing to enter the "while-do" loop again.

The breakout of the "while $\mathcal{S} \neq \emptyset$ do" loop means the isosurface extraction of an object has been completed. These process will not finish until the $\mathcal{Q}$ is null, which means all non-empty cells have been visited. Therefore, this strategy extracts cells in a connected objected and avoids accessing cells sequentially. It has a natural characteristic of separating different objects since it is possible to repair the obtained digital isosurface in a locally bounded way so that the surface is homeomorphic and close to the 3D object, and the well-known topological problems of the MC reconstruction simply do not occur in the digitization of an r-regular object [23]. We can also solve the unusual ambiguity in some cubes via extending more cases [12], [24].

C. Comparison to the growing-cube method

The growing-cube method (GC) [18], [20] is a representative that tracks connected surfaces instead of exhaustive searching isosurfaces cell-by-cell. Although the basic idea of the both methods is similar, they differs from some significant aspects as follows.

- The difference on the lookup table to judge the adjacent cube to be visited. The lookup table in the GC method is based on the intersected edges, without considering the configuration cases. On the contrary, our proposed method is based on the 256 configuration cases. It is independent from the intersected edge.
- The difference on the complexity of obtaining the adjacent directions. The visiting speed on the lookup table in the GC method is much slower than on A-LUT. For example, for simple case 1 and complex case 6, the lookup table is visited by the GC method three times and six times to obtain the neighbor cells respectively, while A-LUT is just visited once by our method. Specially, our method is independent on the number of the intersected edges. Since this operation is very frequent in the deepest loop, this saving is admirable for extracting isosurfaces on large scale volumetric 3D data sets.

Input: volumetric data, iso-value k; configuration and adjacency lookup tables C-LUT; A-LUT; object num. N_o; sampling num. per object N_s
Output: triangular surface collection Σ

begin
  Stack $\mathcal{S} = \emptyset$; Queue $\mathcal{Q} = \emptyset$; Flag array $\mathcal{F} = 0$; Object index $b = 0$; $\Sigma = \emptyset$;
  while $b < N_o$ **do**
    $s = 0$; $\mathcal{R}_b = \emptyset$;
    while $s < N_s$ **do**
      Generate a random integer $p \in \Omega$ (p is the cell index from the left-bottom direction; Ω is the constraint set, e.g. $p \ll N_c$, position);
      if $\mathcal{F}[p] = 0$ **then**
        Compare the values of vertices the pth cell with k and then calculate its configuration index i;
        if $i \notin \{0, 255\}$ **then**
          $\mathcal{F}[p] = i$; $p \rightarrow \mathcal{Q}$; $b = b + 1$; $s = s + 1$; Break;
        else
          $\mathcal{F}[p] = 255$; $s = s + 1$; Continue;
      else
        $s = s + 1$; Continue;
    while $\mathcal{Q} \neq \emptyset$ **do**
      $p \leftarrow \mathcal{Q}$;
      if $\mathcal{F}[p] > 0$ & $\mathcal{F}[p] < 255$ **then**
        $p \rightarrow \mathcal{S}$;
        while $\mathcal{S} \neq \emptyset$ **do**
          $q \leftarrow \mathcal{S}$;
          if $255 > i = \mathcal{F}[q] > 0$ **then**
            Construct the isosurface in the qth cube: calculate the positions of points on the intersected edges; obtain the triangles according to the ith entry of C-LUT; compute the normal vector; triangulate the surface T; $T \rightarrow \mathcal{R}_b$; $\mathcal{F}[q] = 255$; remove q from $\mathcal{Q}$;
            Collect the indices $m_1, m_2, \cdots, m_t (t \leq 6)$ of the adjacent cubes using data in the ith entry of A-LUT;
            For each m_t;
            if $\mathcal{F}[m_t] = 0$ **then**
              Compare the values of vertices the m_tth cell with k and then calculate its configuration index j;
              $\mathcal{F}[m_t] = j$;
            $m_t \rightarrow \mathcal{S}$;
    $\mathcal{R}_b \rightarrow \Sigma$;
  return Σ;

Figure 2. Flowchart of the proposed AlutMC

- The difference on the separation of isosurfaces in algorithm implementation. Although we can segment the isosurfaces related to different objects by measuring their connectivity after MC processing, this procedure is time consuming. It is favorable to separate these isosurfaces directly during the MC processing. Unfortunately, the GC method cannot explicitly achieve this goal in its pseudo-code implementation. On the contrary, our algorithm has a natural ability to separate these isosurfaces in the "while $\mathcal{S} \neq \emptyset$ do" loop.
- The difference on the storage of the cells. Although the GC method builds the adjacent cells by a queue, it does not explicitly present the storage details, which increases the uncertainty of algorithmic implementation. For example, if a cell has three neighbors waiting for isosurface extraction, the data in these cells must be stored unless they are processed directly. However, it is storage-consuming to save the three coordinates of the eight vertices in each cell. On the contrary, our method is based on the stack data structure, it only save the left-bottom indices of the non-empty cells. Therefore, our method has the advantages of both the saving in storage space and

improved feasibility in algorithm implementation.

III. Experimental results and discussion

A. isosurface extraction experiment

We evaluated our AlutMC with comparison to the standard MC (MC), the growing-cube (GC) methods on the VC++ 2010 with Intel Duo 2.4G, 2G RAM PC under Windows XP SP3. We just adopt the surface tracking module of the GC method for fair comparison. The following three data sets were used in our evaluation: (1) $512 \times 512 \times 250$ foot CT images (Figure 3); (2) $512 \times 512 \times 335$ head CT images (Figure 4); (3) $512 \times 512 \times 236$ heart CT images (Figure 5).

On each data set, three algorithms were evaluated under ten different threshold index $k = \{1, 2, \cdots, 10\}$ corresponding to $\{0, \cdots, 450\}$ on the foot set and $\{100, \cdots, 550\}$ on other two sets, both with steps of 50. The number of visited cells and triangles, as well as the execution times are reported in Table I. The number of visited cells and the number of triangles extracted by these algorithms are identical. This fact means both AlutMC and GC do neither loss nor add any cells or triangles in comparison to the standard MC. As a result, we just illustrated the rendered isosurface results in Figs. 3, 4, 5.

Experimental results show that on each data set, the number of visited cells and their triangles surface decrease with the increasing k, as seen from Figure 6, the foot data set is a relatively compact set in comparison with the head set and the chest set, and the chest set is most complex in these data sets. This basic fact results in the different computational savings by the three compared methods. In Figure 6, the execution times of the three algorithms are similar on the foot data set, while remarkably different on the chest data set, especially when the number of non-empty cells is huge. The AlutMC method consistently consumes less times on these data sets, and its advantage is positively related with the number of non-empty cells. Furthermore, another improvement GC is also superior to the standard MC in computational savings. Among these three data sets, the AlutMC performs best on the chest set in comparison to the GC and MC. In fact, this advantage mainly depends on the ratio of the number of non-empty cells over the size of volume data. To verify this observation, we also show the ratio information in Table II. When the ratio of the number of non-empty cells and the number of all cells increases, the computational savings by GC/MC, AlutMC/MC decrease, while by AlutMC/MC increase. Which means AlutMC consistently needs less time than GC, and has advantages over GC in data sets that has huge non-empty cells. In average, AlutMC needs 78.69% time of MC, and 91.06% time of GC on extraction the isosurfaces on these data sets.

The computational saving by AlutMC and GC over MC validates the feasibility to hasten MC by avoiding unnecessary operations on the non-active cells. The advantage of AlutMC over GC can be interpreted in the viewpoint of the program implementation. To find active cells, two "for-end" loops are added in the deepest loop in GC, in other words, at least 144 operations are conducted. On the contrary, these loops are unnecessary in AlutMC, since the lookup table can directly locate the active neighbor cells (see Figure 1). It is noticed that the computational savings by AlutMC are not proportional to the ratio of N_v/N_a. This fact is mainly due to the additional operations in the AlutMC program implementation, such as seeds generation by sampling, building a stack, pushing or popping an index from a stack, allocating and freeing a memory block, etc. Also, we did not conduct any code optimization in our programming implementation.

TABLE II.
Comparison on execution time of the standard marching cubes (MC), the growing-cube (GC), and the proposed AlutMC method.

Data set	$N_v/N_a(\%)$	GC/MC (%)	AlutMC/MC (%)	AlutMC/GC (%)
foot	1.56	84.96	78.21	92.06
head	2.37	86.75	79.16	91.25
chest	3.66	87.56	78.70	89.88

B. Segmentation experiment

In this subsection, we performed the isosurface segmentation evaluation of the proposed AlutMC method, since it can naturally cluster the isosurfaces related to an object during its searching procedure of neighboring active cells. Our testing was still based on the three CT data sets, and the thresholds are as same as the testing in the above subsection. For simplicity, we only report the results under threshold 300 on the foot, the head, and the chest data sets in Figures 8,9,10. We can find from the first subfigures in these figures that there are many small components generated by the isosurface extraction algorithm. This observation is also supported by the second subfigures, where although the components quantitatively reach to 400 in the foot set, 6000 in the chest set, and 8000 in the head set, the large components with huge cells are rather few. For instance, in the foot data set there are only 18 components having more than 10000 cells, in the head set the number is 3, and in the chest set the number is 10. This motivates us to segment the main components by using the number of visited cells that can be easily captured in the programming implementation of the AlutMC method. The resulting segmentation isosurfaces are presented in the rest subfigures of Figures 8, 9, 10. From where we can clearly find that the main components in each data set have been successfully separated from other components and the noisy components are removed simultaneously. This experimental result is also helpful for further fastening the AlutMC's speed in the isosurface extraction if we restrict the cell number of components within a predefined range.

The standard MC method cannot separate the main components because it process each cell sequentially,

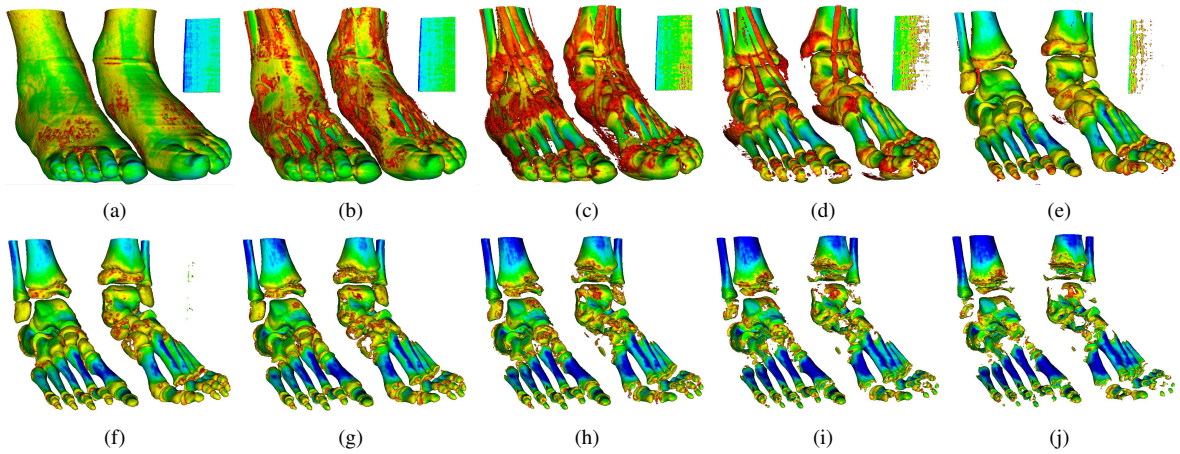


Figure 3. Rendered isosurfaces on foot data set, where k ranges from 0 to 450 with step 50.

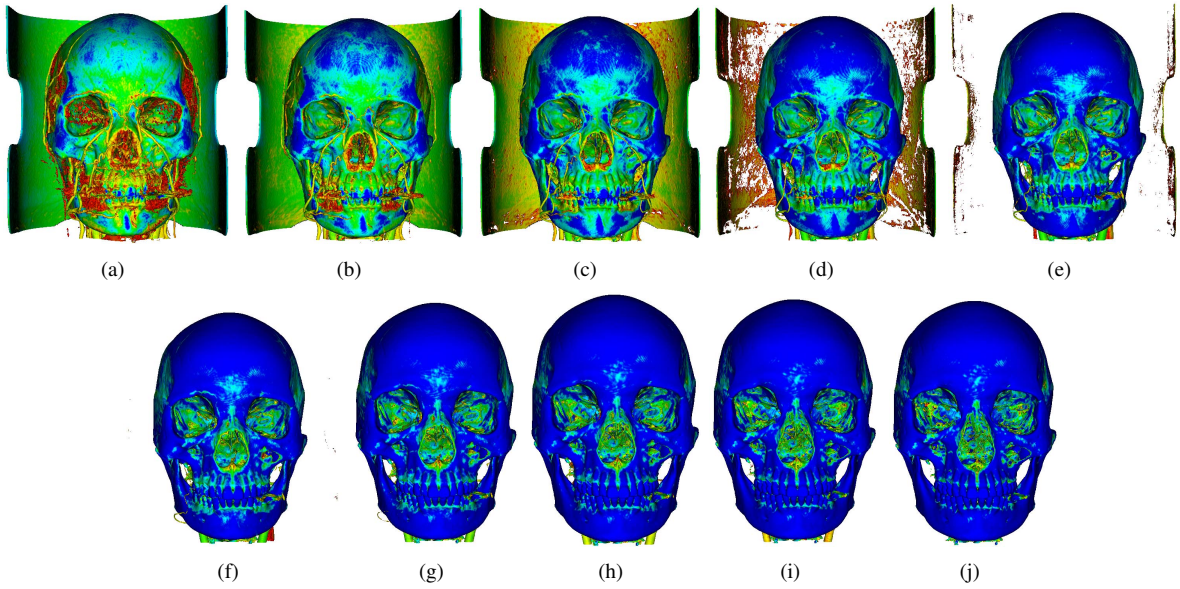


Figure 4. Rendered isosurfaces on head data set, where k ranges from 100 to 550 with step 50.

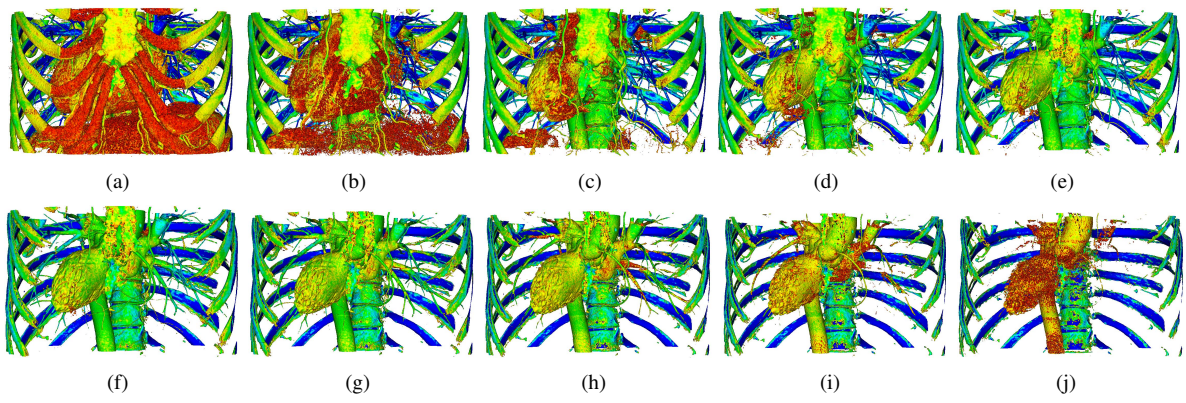
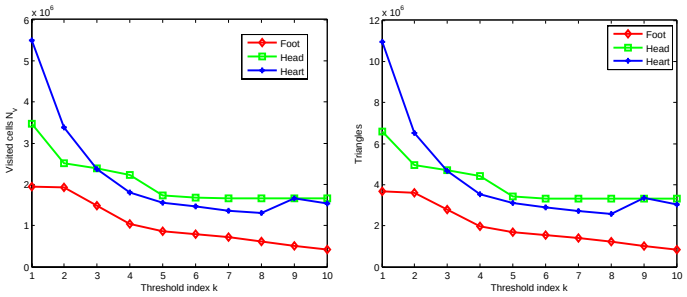


Figure 5. Rendered isosurfaces on head data set, where k ranges from 100 to 550 with step 50.

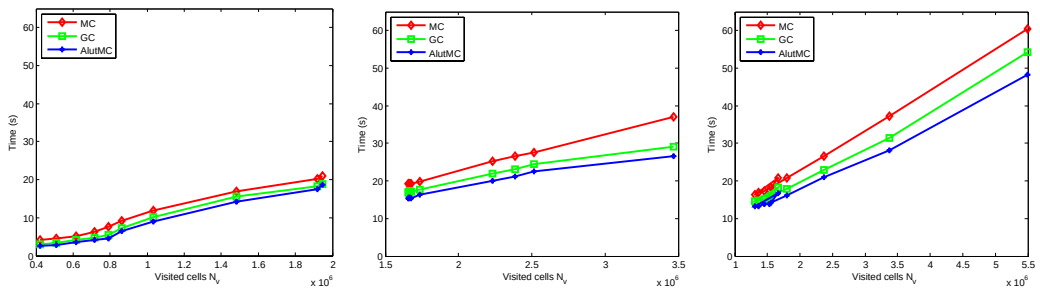
TABLE I.
EXECUTION TIMES OF THREE ALGORITHMS ON THREE CT DATA SETS, WHERE N_a, k, N_v, AND N_{tri} STAND FOR THE NUMBER OF ALL CELLS, ISO-VALUE, NUMBER OF VISITED CELLS, NUMBER OF TRIANGLES, RESPECTIVELY.

Data set	N_a	k	N_v	N_{tri}	MC time (s)	GC time (s)	AlutMC time (s)
Foot	65019129	0	1903787	3683764	20.92	18.92	18.63
	$(512 \times 512 \times 250)$	50	1877339	3613151	20.23	18.31	17.48
		100	1456062	2792339	16.89	15.49	14.24
		150	1019368	1987888	11.92	10.22	9.09
		200	855577	1684164	9.22	7.23	6.50
		250	786234	1552848	7.67	5.48	4.63
		300	709462	1404667	6.30	4.76	4.10
		350	612098	1212506	5.18	4.09	3.54
		400	508174	1006749	4.46	3.30	2.83
		450	419088	832006	4.18	3.08	2.62
Head	87214414	100	3465809	6598900	37.14	29.09	26.68
	$(512 \times 512 \times 335)$	150	2513547	4954992	27.55	24.41	22.47
		200	2385469	4716301	26.58	23.14	21.22
		250	2231346	4403379	25.24	21.93	19.97
		300	1736394	3430008	19.91	17.65	16.27
		350	1674200	3330875	19.23	17.06	15.46
		400	1666595	3319857	19.28	17.27	15.66
		450	1660348	3313055	19.25	17.21	15.58
		500	1663183	3321288	19.20	16.99	15.37
		550	1658801	3315559	19.17	16.98	15.40
Chest	59796709	100	5491887	10945083	60.37	54.21	48.17
	$(512 \times 512 \times 236)$	150	3375070	6514204	37.26	31.49	28.18
		200	2363599	4677329	26.58	23.02	20.97
		250	1797876	3542235	20.72	17.90	16.10
		300	1557917	3097749	18.25	15.87	14.21
		350	1457134	2908263	17.39	15.26	13.83
		400	1357638	2711399	16.89	14.74	13.22
		450	1303225	2576707	16.41	14.56	13.20
		500	1660274	3344551	20.78	18.35	16.65
		550	1531190	3045028	17.50	15.38	13.91



(a) (b)

Figure 6. Visited cells and the corresponding triangles of compared methods on three data sets.



(a) foot (b) head (c) chest

Figure 7. Execution time of the compared methods on three data sets.

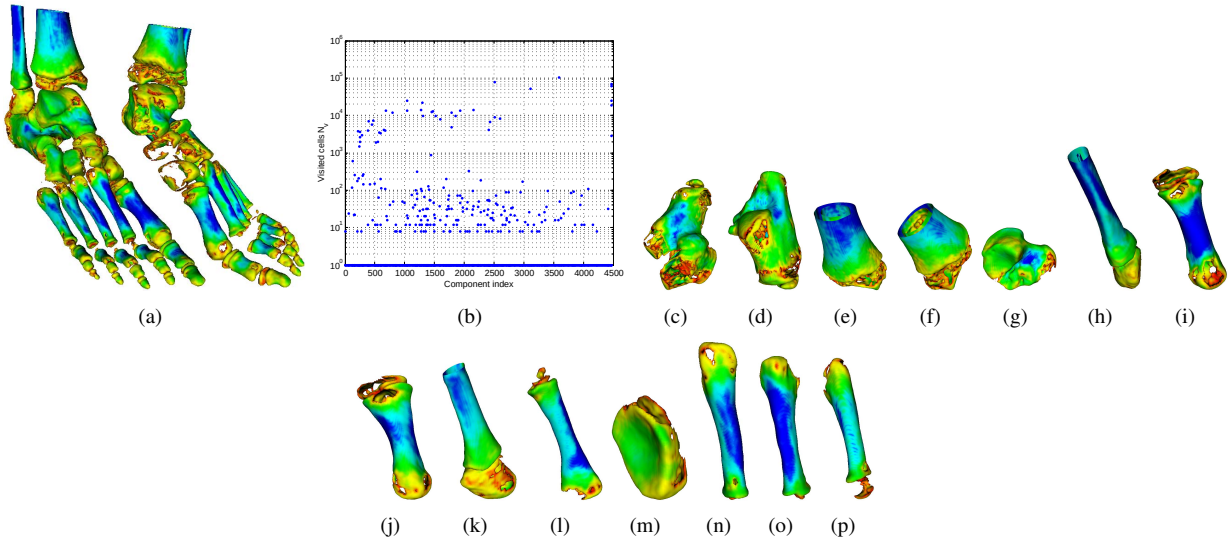


Figure 8. Segmentation result by AlutMC on the foot data set ($k = 300$). (a) whole rendered isosurface; (b) cell number of main components in (a); (c)-(p) segmentation results of the main components. The corresponding cell number is 102027, 79134, 67937, 59948, 52130, 24624, 24329, 22086, 18834, 14222, 14037, 13457, 13407, 13302, respectively.

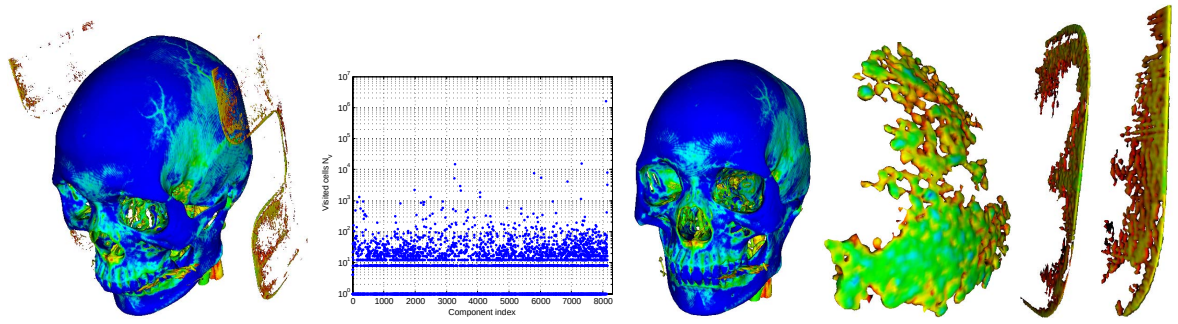


Figure 9. Segmentation result by AlutMC on the head data set ($k = 300$). (a) whole rendered isosurface; (b) cell number of main components in (a); (c)-(f) segmentation results of the main components. The corresponding cell number is 1569115, 15588, 15087, 7984, respectively.

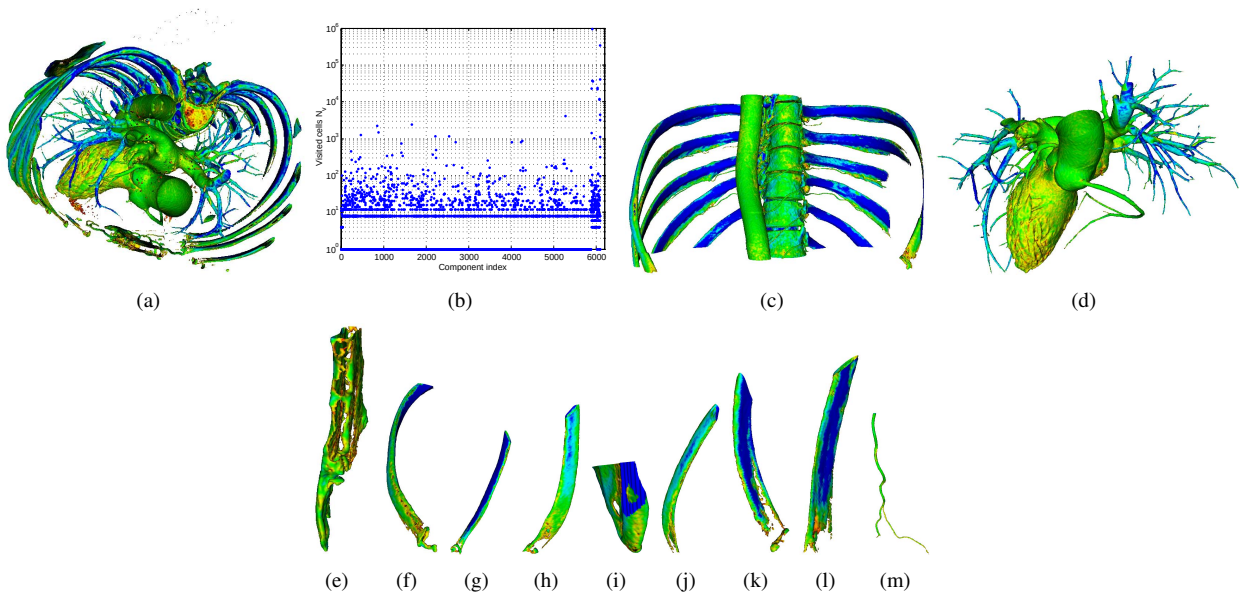


Figure 10. Segmentation result by AlutMC on the chest data set ($k = 300$). (a) whole rendered isosurface; (b) cell number of each component in (a); (c)-(m) segmentation results of the main components. The corresponding cell number is 933960, 344887, 40296, 37867, 35858, 23950, 22725, 21840, 11968, 11196, 4506, respectively.

without considering the neighboring cells of an active cell. The growing-cube (GC) method explicitly searches these neighboring active cells, however, it doesn't put any attention of the isosurface separation in its algorithmic implementation. Therefore, we did not do any isosurface separation comparison among the three methods since this advantage is exclusive for the proposed AlutMC method.

IV. Conclusions

We have presented an improved marching cubes method to extraction isosurfaces based on a new adjacent lookup table. Using this lookup table, the proposed approach tracks connected surfaces instead of exhaustive searching isosurfaces cell-by-cell. With random sampling and predefined seeds, it saves computation timing in identifying isosurfaces. Furthermore, the proposed algorithm has a natural ability to separate the isosurfaces related to different objects, which is very advisable for marching cubes -based segmentation tasks. Additionally, it does not require additional data structure to speed up traversing cells or to decimate triangular mesh. Therefore, the proposed method can serve as a replacement of the standard marching cubes because it is quite cost-effective and self-contained. In our future work, we will utilize the prior information related to the object for low memory constrained isosurface extraction by combining the proposed method and the statistical approach [25]. We also intend to simplify the pattern isosurfaces of local triangle data to improve the visual quality, since the extracted triangle data in our method has a clustering property for each object.

Acknowledgment

This work was partially supported by the Chinese National Science Foundation (60903142, 61190122), the China Post-doctoral Science Foundation (2013T60841, 2012M521677, Xm201306), the Natural Science Foundation of CQ-CSTC (cstc2011jjA40024), the Sci. & Tech. Research Project of Chongqing Municipal Education Commission (KJ120601), and the Fundamental Research Funds for the Central Universities (106112013CDJZR12001). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] G. M. Nielson, "On marching cubes," *IEEE Trans. Vis. Comput. Graph.*, vol. 9, no. 3, pp. 283–297, 2003.

[2] V. S. Lempitsky, "Surface extraction from binary volumes with higher-order smoothness," in *Proc. 23rd IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA, USA, 13-18 June 2010, pp. 1197–1204.

[3] S. Nagaraj and V. Natarajan, "Relation-aware isosurface extraction in multifield data," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 182–191, 2011.

[4] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[5] K. Pöthkow, B. Weber, and H.-C. Hege, "Probabilistic marching cubes," *Comput. Graph. Forum*, vol. 30, no. 3, pp. 931–940, 2011.

[6] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.

[7] D. A. Rajon and W. E. Bolch, "Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models," *Computerized Medical Imaging and Graphics*, vol. 27, no. 5, pp. 411–435, 2003.

[8] D. Feltell and L. Bai, "A new marching cubes algorithm for interactive level set with application to MR image segmentation," in *Proc. 6th Int'l Symposium Advances in Visual Computing (ISVC 2010), LNCS 6453*, Las Vegas, NV, USA, November 29-December 1 2010, pp. 371–380.

[9] S. H. Cui and J. Liu, "Simplified patterns for extracting the isosurfaces of solid objects," *Image and Vision Comput.*, vol. 26, no. 2, pp. 174–186, 2008.

[10] K. K. Delibasis, G. K. Matsopoulos, N. A. Mouravliansky, and K. S. Nikita, "A novel and efficient implementation of the marching cubes algorithm," *Computerized Medical Imaging and Graphics*, vol. 25, no. 4, pp. 343–352, 2001.

[11] C. Dyken, G. Ziegler, C. Theobalt, and H.-P. Seidel, "High-speed marching cubes using histopyramids," *Comput. Graph. Forum*, vol. 27, no. 8, pp. 2028–2039, 2008.

[12] T. Lewiner, H. Lopes, A. W. Vieira, and G. Tavares, "Efficient implementation of marching cubes cases with topological guarantees," *Journal of Graphics Tools*, vol. 8, no. 2, pp. 1–15, december 2003.

[13] E. V. Chernyaev, "Marching cubes 33: Construction of topologically correct isosurfaces," CERN¦CN-95-17, Tech. Rep. 1-8, 1995.

[14] S. Raman and R. Wenger, "Quality isosurface mesh generation using an extended marching cubes lookup table," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 791–798, 2008.

[15] C. A. Dietrich, C. E. Scheidegger, J. L. D. Comba, L. P. Nedel, and C. T. Silva, "Edge groups: An approach to understanding the mesh quality of marching methods," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1651–1666, 2008.

[16] C. A. Dietrich, C. E. Scheidegger, J. M. Schreiner, J. L. D. Comba, L. P. Nedel, and C. T. Silva, "Edge transformations for improving mesh quality of marching cubes," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 1, pp. 150–159, 2009.

[17] C. A. Dietrich, C. E. Scheidegger, J. L. D. Comba, L. P. Nedel, and C. T. Silva, "Marching cubes without skinny triangles," *Computing in Science and Engineering*, vol. 11, no. 2, pp. 82–87, 2009.

[18] Y. Xi and Y. Duan, "A novel region-growing based isosurface extraction algorithm," *Computers & Graphics*, vol. 32, no. 6, pp. 647–654, 2008.

[19] A. van Gelder and J. Wilhelms, "Topological considerations in isosurface generation," *ACM Trans. Graph.*, vol. 13, pp. 337–375, October 1994.

[20] T.-Y. Lee and C.-H. Lin, "Growing-cube isosurface extraction algorithm for medical volume data," *Computerized Medical Imaging and Graphics*, vol. 25, no. 5, pp. 405–415, 2001.

[21] R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill, "Octree-based decimation of marching cubes surfaces," in *IEEE Visualization*, 1996, pp. 335–342.

[22] X. Wang, Z. Wang, Y. Niu, S. Zhang, L. Tan, and J. Jin, "Improved marching cubes by combining case lookup table and adjacency lookup sub-table," *Journal of Chongqing Univ. (Natural Science)*, vol. 35, no. 12, pp. 68–77,83, 2012.

[23] P. Stelldinger, L. J. Latecki, and M. Siqueira, "Topological equivalence between a 3D object and the reconstruction of

its digital image," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 126–140, 2007.
[24] T. Etiene, L. G. Nonato, C. E. Scheidegger, J. Tierny, T. J. Peters, V. Pascucci, R. M. Kirby, and C. T. Silva, "Topology verification for isosurface extraction," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 6, pp. 952–965, 2012.
[25] B. Duffy, H. Carr, and T. Möller, "Integrating isosurface statistics and histograms," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 2, pp. 263–277, 2013.

Xuchu Wang received his Ph.D. degree in instrument science and technology from Chongqing University (CQU) China in 2007. He then joined CQU in July 2007 and is currently an associate professor in the College of Optoelectronic Engineering and a research faculty in the Key Laboratory of Optoelectronic Technology and Systems of Ministry of Education. He received the "university-wide outstanding PhD dissertation award" and "10 university-wide distinguished high-level journal paper award" in 2007 for his research work on biometrics. His research interests are pattern classification and machine learning, medical image analysis and 3D reconstruction, biometrics (mainly based on fingerprint and face), and practical PR/biometrics system development, where he had more than 20 peer-reviewed papers published.

Yanmin Niu received her BSc and MSc degrees in instrument science and technology from Chongqing University in 1997 and 2003, respectively. Currently she is an associate professor in the College of Computer and Information Science, Chongqing Normal University. Her research interests include medical image analysis and machine learning, where she had more than 10 peer-reviewed papers published.

Li-Wen Tan is currently a senior experimentalist at the Department of Anatomy, College of Basic Medical Science of the Third Military Medical University, Chongqing. His current research interests include three-dimensional biomedical imaging, medical image analysis and applied anatomy visualization.

Shao-Xiang Zhang received his Ph.D. degree in clinical anatomy from the Third Military Medical University, Chongqing, China in 1991. He worked at the Anatomisches Institut der Universitt Bonn, Germany from 1991 to 1994. He is currently a distinguished professor at the Department of Anatomy, College of Basic Medical Science of the Third Military Medical University, Chongqing, China. His main research interests include three-dimensional biomedical imaging, medical imaging anatomical analysis and visualization.