

Reconstruction of 3D Objects from 2D Cross-sections with the 4-point Subdivision Scheme Adapted to Sets*

Shay Kels[†] and Nira Dyn

School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel
e-mail: kelsisha@tau.ac.il; niradyn@tau.ac.il

Abstract: Reconstruction of 3D objects from 2D cross-sections is an intriguing problem with many potential applications. We approach this problem through a novel multi-resolution method based on iterative refinement of the sets representing the cross-sections. To that end, we introduce a new geometric weighted average of two sets, defined for positive weights (corresponding to interpolation) and when one weight is negative (corresponding to extrapolation). This new average can be used to interpolate between cross-sections of a 3D object in a piecewise way. To obtain a smoother reconstruction of the 3D object, we adapt to sets the 4-point interpolatory subdivision scheme using the new average with both positive and negative weights. The effectiveness of the new method is demonstrated by several examples.

1. Introduction

Methods for reconstruction of objects from cross-sections have many applications. An important area of application is medical imaging [3, 6, 18], where 3D images are reconstructed from 2D slices obtained by medical imaging devices such as computational tomography (CT). Among other areas of application are computer graphics and animation, where sequences of intermediate shapes are created between two or more 2D or 3D shapes [21].

The problem of reconstruction from parallel cross-sections has been studied extensively for the past four decades and there exists a significant body of literature on this topic. Some methods, known as *parametric*, attempt to solve this problem by finding correspondence between points on the boundaries of the given cross-sections. Intermediate cross-sections are then created by interpolating positions of the sequences of corresponding points. [4, 5, 23].

Another approach is to represent the cross-sections by *implicit functions* and then to interpolate between the implicit functions. A particularly popular method, originally proposed in [15], latter modified in [12, 18] and extended in [7, 8, 16], is based on the representation of a cross-section by its *signed distance function*. In this method the cross-sections are treated as sets, or equivalently as binary images. First, for each cross-section its signed distance function is computed. Next the signed-distance functions are pointwise interpolated by some univariate interpolation method, usually by linear or cubic spline interpolation. Finally the resulting function is thresholded at zero level, to obtain the reconstructed object.

Few works attempt to solve the problem by subdivision of sets. The main theme of this approach is the adaptation to sets of real valued subdivision methods (*see e.g.* [10]), by first expressing weighted

*This is a preprint version of the paper published in Computers & Graphics 35(2011)741 - 746, Elsevier.

[†]corresponding author

averages between several numbers by sequences of weighted averages between two numbers (*binary weighted averages*). Binary weighted averages of numbers are then replaced by binary weighted averages of sets. In [9], spline subdivision schemes are adapted to sets using the *metric average* of sets. In [25], the Chaikin subdivision scheme is adapted to sets, based on the *straight skeleton average* as a binary weighted average.

This work is a combination of the last two approaches. We develop a new weighted average of two sets based on the signed distance function. The new binary weighted average is defined for positive weights and also when one weight is negative, therefore it performs both interpolation and extrapolation. Then we adapt to sets the 4-point interpolatory subdivision scheme [11], using the new binary average. Our numerical simulations demonstrate the quality of the reconstruction.

Although in this work we focus on the reconstruction of 3D objects from 2D cross-sections, our ideas are immediately applicable in any finite dimension. In particular, our method can be used to interpolate between a sequence of 3D objects.

The structure of this work is as follows. In section 2 we introduce the new binary weighted average of sets, and interpolate piecewise between each two consecutive cross-sections. In section 3, we adapt to sets the 4-point subdivision scheme and use it for the reconstruction of objects from cross-sections. Section 4 presents applications of our method to various data sets. The complexity of the method is discussed in section 5. In section 6, we draw conclusions and propose directions for future research.

2. The new binary average of sets

2.1. The average of "simply different" sets

2D Sets representing cross-sections of a 3D object can be of complicated topology, see Figure 1. We approach the construction of the new average of such sets by reducing it to several simple problems. First we consider the new average in the simple case of two sets $A, B \subset \mathbb{R}^2$, such that one set is contained in the other, $B \subset A$, and the *difference set*,

$$A \setminus B = \{p \in A : p \notin B\} ,$$

has only one *connected component*. We call two such sets *simply different*. For two simply different sets A, B , one can think of the difference set $A \setminus B$ as a "vector" connecting the two sets. Therefore, when constructing a weighted average of two simply different sets, it is natural to try to mimic the very familiar case of a weighted average of two points in \mathbb{R}^2 . Given two points $p, q \in \mathbb{R}^2$ it is easy to observe the following properties of the weighted average $tp + (1 - t)q$, $t \in \mathbb{R}$:



FIG 1. A cross-section of CT scan of human pelvis

1. The weighted average passes through q and p at $t = 0$, and $t = 1$ respectively.
2. All averages are points on the same line. One way to express this idea is:

$$d_E(p_1, p_3) = d_E(p_1, p_2) + d_E(p_2, p_3) ,$$

where $p_i = t_i p + (1 - t_i) q$, $i = 1, 2, 3$, $t_1 < t_2 < t_3$ and d_E is the Euclidean distance.

3. The average moves along the line with a constant velocity vector $(p - q)$.

We also observe that for $t \in [0, 1]$ the average interpolates between the given points. For $t < 0$, the average extrapolates from p beyond q , and for $t > 1$ the average extrapolates from q beyond p . We aim to construct an average of two simply different sets satisfying the above three properties and extending the idea of interpolation and extrapolation.

Our construction is based on the *symmetric difference metric*, which is a widely used metric to measure dissimilarity between sets [20, 22]. The *symmetric difference* of two sets A and B is,

$$A\Delta B = (A \setminus B) \cup (B \setminus A) ,$$

and the *symmetric difference metric*¹ of $A, B \subset \mathbb{R}^2$ is defined by,

$$d_{\mathcal{L}}(A, B) = \text{Area}(A\Delta B).$$

First we introduce what we call the *distance average* of two simply different sets, which we later modify to get our new average with the desirable properties. The distance average is based on the method of interpolation of the *signed distance functions* introduced in [15]. The *signed distance* from a point p to a set A is defined by,

$$d_S(p, A) = \begin{cases} d_E(p, \text{Boundary}(A)) & p \in A \\ -d_E(p, \text{Boundary}(A)) & p \notin A , \end{cases}$$

with d_E the Euclidean distance from a point to a set. We use the signed distance function, since in the literature it is a basic model for implicit representation of sets. We define the distance average between two simply different sets as the set,

$$tA \widetilde{\oplus} (1-t)B = \{p : f_{A,B,t}(p) \geq 0\} , \quad (1)$$

where $f_{A,B,t}(p) = td_S(p, A) + (1-t)d_S(p, B)$. Note that $f_{A,B,t}$ is not the signed distance function of $tA \widetilde{\oplus} (1-t)B$.

It is easy to see that the distance average passes through the original sets at $t = 0$ and $t = 1$. We observe another important property of the distance average, which is

$$C_1 \cap C_3 \subseteq C_2 \subseteq C_1 \cup C_3 , \quad (2)$$

with $C_i = t_i A \widetilde{\oplus} (1-t_i) B$, $i = 1, 2, 3$ and $t_1 \leq t_2 \leq t_3$. Relation (2) can be validated as follows. Let $p \in C_1 \cap C_3$, it follows from (1), that $f_{A,B,t_1}(p) \geq 0$ and $f_{A,B,t_3}(p) \geq 0$. Consequently, since $f_{A,B,t}(p)$ is linear as a function of t , for any $t_2 \in [t_1, t_3]$, $f_{A,B,t_2}(p) \geq 0$. So $p \in C_2$ and thus $C_1 \cap C_3 \subseteq C_2$. From similar considerations for the complement set of $C_1 \cup C_3$, we get $C_2 \subseteq C_1 \cup C_3$.

We observe also that all averages are "points" on an abstract "line" of sets due to the distance relation,

$$d_{\mathcal{L}}(C_1, C_3) = d_{\mathcal{L}}(C_1, C_2) + d_{\mathcal{L}}(C_2, C_3) , \quad (3)$$

¹Technically, the "symmetric difference metric" is a metric on sets that are equal to the closure of their interior [20].



FIG 2. The distance average and the new average of two given sets.

which follows from (2) and Theorem 2 in [19]. However the distance average of sets lacks the constant velocity property, as can be observed from the following example. Consider the sets A and B shown in Figure 2 as clipped together. Note that the difference set $A \setminus B$ has only one connected component (the inner oval), therefore A, B are simply different. Let p be a point in the interior of $A \setminus B$. For such a point:

$$d_S(p, A) > 0, \quad d_S(p, B) < 0 \quad \text{and} \quad |d_S(p, A)| > |d_S(p, B)| .$$

Therefore for any $p \in A$,

$$\frac{1}{2}d_S(p, A) + \frac{1}{2}d_S(p, B) > 0 ,$$

and consequently $\frac{1}{2}A \widetilde{\oplus} \frac{1}{2}B = A$ (see Figure 2). Indeed, it is undesirable to get one of the original sets, as an equally weighted average of the two different sets, since such average does not reflect a continuous transition of shape between the two sets.

To overcome this difficulty we construct a new weighted average of two simply different sets as a reparametrization of the distance average, imposing the constant velocity property. For this purpose we define the function,

$$g_{A,B}(x) = \text{sign}(x) \frac{d_{\mathcal{L}}(xA \widetilde{\oplus} (1-x)B, B)}{d_{\mathcal{L}}(A, B)} . \quad (4)$$

It follows from (4) and (3), that the function $g_{A,B}$ is non-decreasing and $g_{A,B}(0) = 0, g_{A,B}(1) = 1$. The function $g_{A,B}$ reveals us how the average $xA \widetilde{\oplus} (1-x)B$ is located relative to the set B (which corresponds to $x = 0$). Next we define,

$$g_{A,B}^{-1}(t) = \arg \min_x \{|g_{A,B}(x) - t|\} . \quad (5)$$

Observe that if t belongs to the *range* of $g_{A,B}$, then

$$g_{A,B}(g_{A,B}^{-1}(t)) = t. \quad (6)$$

Since $g_{A,B}$ is non-decreasing, the optimal value of x needed in (5), is quickly calculated to desired precision by a simple *divide and conquer* method.

We use $g_{A,B}^{-1}$ to reparametrize the distance average and to obtain the *new binary weighted average* for simply different sets,

$$tA \widetilde{\oplus} (1-t)B = g_{A,B}^{-1}(t)A \widetilde{\oplus} (1-g_{A,B}^{-1}(t))B . \quad (7)$$

It follows from (4) and (5), that $\tau = g_{A,B}^{-1}(t)$ is the value of the parameter, having the same sign as t , such that $d_{\mathcal{L}}(\tau A \widetilde{\oplus} (1-\tau)B, B)$ is as close as possible to $|t|d_{\mathcal{L}}(A, B)$. Thus the new average is

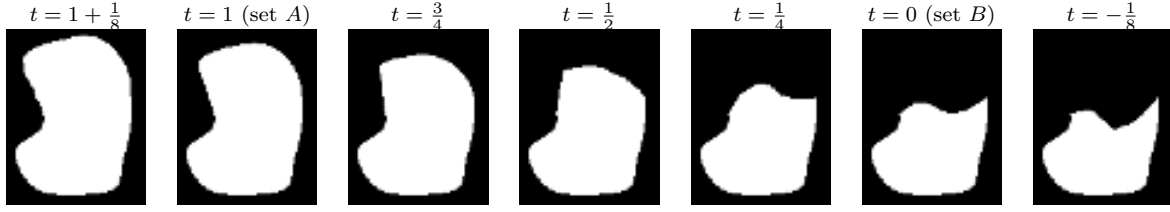


FIG 3. The average $tA \oplus (1-t)B$ of simply different sets for various values of the averaging parameter t .

the optimal reparametrization of the distance average relative to the "constant velocity" property. In most cases, for $t \in [0 - \delta; 1 + \delta]$ with small δ , (6) is satisfied, so the "constant velocity" property exists exactly,

$$d_{\mathcal{L}} \left(tA \oplus (1-t)B, B \right) = |t| d_{\mathcal{L}} (A, B) . \quad (8)$$

It can be observed from (4) and (5), that similarly to $g_{A,B}$ the function $g_{A,B}^{-1}$ is non-decreasing and $g_{A,B}^{-1}(0) = 0$, $g_{A,B}^{-1}(1) = 1$. Therefore the new average satisfies relations (2) and (3), with $C_i = t_i A \oplus (1-t_i) B$.

It follows from (2) that for simply different sets A, B , such that $B \subset A$ we have,

$$tA \oplus (1-t)B \subset B, \quad t < 0, \quad (9)$$

$$B \subset tA \oplus (1-t)B \subset A, \quad t \in (0, 1), \quad (10)$$

$$A \subset tA \oplus (1-t)B, \quad t > 1, \quad (11)$$

and

$$1A \oplus 0B = A, \quad 0A \oplus 1B = B . \quad (12)$$

The idea of reparametrization for gradual interpolation between two sets follows the one in [13], but our construction, which is based on the signed distance function, supports also extrapolation.

The example in Figure 2 demonstrates the advantage of our new average \oplus over the distance average $\widetilde{\oplus}$. In this example $g_{A,B}^{-1} \left(\frac{1}{2} \right) = 0.3594$. Figure 3 shows examples of the average $tA \oplus (1-t)B$ for varying values of the averaging parameter t . Observe how the average $tA \oplus (1-t)B$ cuts into the set B for $t < 0$, interpolates between the two sets for $t \in [0, 1]$ and expands beyond the set A for $t > 1$. This geometric behavior of the new average is in correspondence with relations (9) - (12), and carries the idea of extrapolation into the context of sets.

2.2. The average of general sets

We now consider the weighted average of two general sets $A, B \subset \mathbb{R}^2$, representing cross-sections of a complex 3D object, thus having complicated topology (see Figure 1). Formally, the average \oplus in (7) is well defined for any two sets. However the function $g_{A,B}$ in (4) dictating the change of the averaging parameter in (7) is global in its nature. Consequently, when used for general two sets (which are not simply different), the reparametrization creates interdependence between distant regions of the symmetric difference between the two sets, and so may lead to undesired results. To overcome this

problem, we break the average of two general sets into several averages of *simply different* sets, this way taking into consideration the local changes of the geometry between the two sets.

The main elements of the new average of two general sets A, B are the *connected components* of $A \setminus B$ and $B \setminus A$. For each *connected component* C of $A \setminus B$, we compute using (7),

$$R_C = tA' \bigoplus (1-t)B', \quad (13)$$

with $A' = C \cup (A \cap B)$ and $B' = A \cap B$. Note that the sets A', B' are *simply different*, $B' \subset A'$. A similar procedure with $1-t$ as the averaging parameter is applied to all connected components of $B \setminus A$.

The results of the above computations for simply different sets have to be carefully merged, taking into account interpolation and extrapolation properties induced by relations (9) - (12). First consider the case $t \in [0, 1]$. Let C be a connected component of $A \setminus B$ or $B \setminus A$, we obtain from (10),

$$A \cap B \subset R_C \subset ((A \cap B) \cup C),$$

with R_C defined in (13). Therefore, for $t \in [0, 1]$, we define the average of $tA \bigoplus (1-t)B$ as the union of all sets obtained in the computations for simply different sets, namely

$$tA \bigoplus (1-t)B = \left(\bigcup_{C \in CC(A \setminus B)} R_C \right) \cup \left(\bigcup_{C \in CC(B \setminus A)} \tilde{R}_C \right),$$

with $\tilde{R}_C = (1-t)A' \bigoplus tB'$. Here the notation $CC(D)$ stands for the collection of all connected components of a set D .

For $t \notin [0, 1]$ the extrapolation of simply different sets is to be preserved during the merging. Assume that $t > 1$, then for $C \in CC(A \setminus B)$ we have from (11),

$$((A \cap B) \cup C) \subset R_C,$$

meaning that R_C expands beyond the set A . Similarly, for $C \in CC(B \setminus A)$, we have from (9),

$$\tilde{R}_C \subset A \cap B,$$

meaning that \tilde{R}_C cuts into $A \cap B$. Thus we first join the results obtained for all $C \in CC(A \setminus B)$,

$$R_1 = \bigcup_{C \in CC(A \setminus B)} R_C,$$

then find the intersection of $A \cap B$ with all results obtained for $C \in CC(B \setminus A)$,

$$R_2 = (A \cap B) \bigcap_{C \in CC(B \setminus A)} \tilde{R}_C.$$

Finally the average of A, B is defined as

$$tA \bigoplus (1-t)B = (R_1 \setminus (A \cap B)) \cup R_2.$$

The average of A, B with the averaging parameter $t < 0$ is equivalent to the average of B, A with the averaging parameter $1-t$. The procedure of computation of the new average of two general sets A, B is summarized in Algorithm 1.

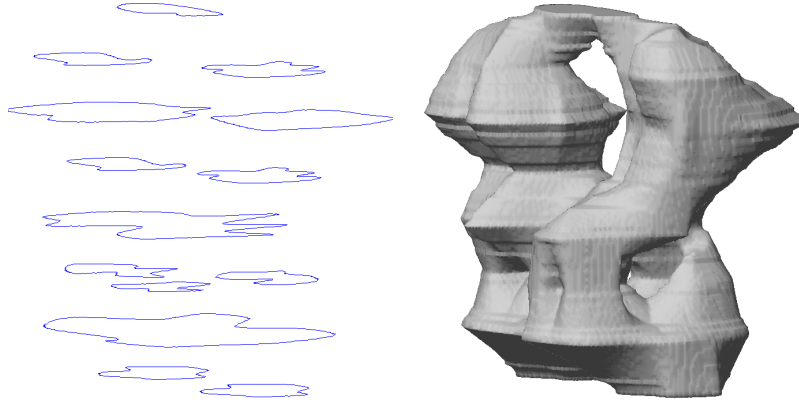


FIG 4. Reconstruction of an object from parallel cross-sections in a piecewise way.

Algorithm 1 ComputeAverage(A,B,t)

```

if  $t < 0$  then
    return ComputeAverage(B,A,1-t)
end if
Compute the sets  $A \cap B, A \setminus B, B \setminus A$ 
Compute  $CC(A \setminus B), CC(B \setminus A)$ 
    {Compute averages of simply different sets}
for all  $C$  in  $CC(A \setminus B)$  do
    Compute using (7):
         $RC = t((A \cap B) \cup C) \oplus (1-t)(A \cap B)$ 
    Add  $RC$  to the list of sets resultsA
end for
for all  $C$  in  $CC(B \setminus A)$  do
    Compute using (7):
         $RC = (1-t)((A \cap B) \cup C) \oplus t(A \cap B)$ 
    Add  $RC$  to the list of sets resultsB
end for
    {Merge the results depending on the averaging parameter}
if  $0 \leq t \leq 1$  then
    result = the union of all sets in resultsA and resultsB
else
     $R1$  = the union of all sets in resultsA
     $R1 = R1 \setminus (A \cap B)$ 
     $R2 = A \cap B$ 
     $R2 = R2 \cap$  (the intersection of all sets in resultsB)
    result =  $R1 \cup R2$ 
end if
return result
    
```

The new average \oplus performs well when there is a substantial intersection between the two sets. For the reconstruction of a smooth 3D object from its parallel 2D cross-sections, the assumption that the projections of two consecutive cross-sections into a parallel plane intersect significantly is natural.

Using the new average of sets we can reconstruct objects by interpolating between every two consecutive cross-sections. Note that the rate of change of the area of the symmetric difference of the cross-sections of the reconstructed object is piecewise constant, which justifies the term "piecewise

linear interpolation” for this reconstruction. An example of such reconstruction is shown in Figure 4.

Reconstruction from cross-sections in a piecewise way may result in an object whose boundary makes sharp turns in the transition from one layer to another, as can be seen in Figure 4. To overcome this problem and still interpolate the original data, we adapt in the next section the 4-point interpolatory subdivision scheme to sets, using our new binary weighted average.

3. The 4-point subdivision scheme of sets as a reconstruction tool

Subdivision schemes are efficient computational methods for the design, representation and approximation of curves and surfaces (*see e.g.* [10]). Following the approach in [9], we adapt the 4-point interpolatory subdivision scheme of [11] to sets, by first expressing the refinement rule in terms of repeated binary weighted averages of points, and then replacing the binary weighted averages of points by the binary weighted averages of sets, defined in the previous section.

For points in \mathbb{R}^n , the 4-point subdivision scheme is defined by the following refinement rule, which is applied repeatedly for $k = 0, 1, 2, 3, \dots$,

$$P_{2i}^{k+1} = P_i^k, \quad (14)$$

and

$$P_{2i+1}^{k+1} = -w (P_{i-1}^k + P_{i+2}^k) + (1/2 + w) (P_i^k + P_{i+1}^k). \quad (15)$$

Here $\{P_i^k\}_{i \in \mathbb{Z}}$ are control points at refinement level k of the subdivision process and w is a *fixed tension parameter*. Usually w is chosen to be $1/16$. The coefficients in (15) sum to one, so (15) is a weighted average of the four points $P_{i-1}^k, \dots, P_{i+2}^k$, and therefore can be rewritten in terms of repeated binary weighted averages as,

$$\begin{aligned} P_{2i+1}^{k+1} &= \frac{1}{2} ((-2w) P_{i-1}^k + (1 + 2w) P_i^k) \\ &+ \frac{1}{2} ((-2w) P_{i+2}^k + (1 + 2w) P_{i+1}^k). \end{aligned} \quad (16)$$

The insertion rule (16) consists of two extrapolations between consecutive points and one $\frac{1}{2}$ -average between the results of these extrapolations. We adapt the refinement rule defined by (14) and (16) to sets by replacing binary weighted averages of points by the new binary weighted averages of sets. The refinement rule for the sets $\{F_i^k\}$ becomes:

$$F_{2i}^{k+1} = F_i^k, \quad (17)$$

and

$$F_{2i+1}^{k+1} = \frac{1}{2} E_{2i+1}^{k+1} \oplus \frac{1}{2} H_{2i+1}^{k+1}, \quad (18)$$

with

$$E_{2i+1}^{k+1} = (-2w_{k,i}) F_{i-1}^k \oplus (1 + 2w) F_i^k,$$

and

$$H_{2i+1}^{k+1} = (-2w) F_{i+2}^k \oplus (1 + 2w) F_{i+1}^k.$$

Note that the subdivision with the refinement rule (17)-(18) is interpolatory. The convergence of the scheme for $w < \frac{1}{8}$ to a continuous *set-valued function* and its approximation properties are proved in [14].

Data Set	Low Resolution	High Resolution	Refinement steps	Vol. of $R\Delta R'$ (voxels)	Vol. of $R\cup R'$ (voxels)	$d_H(R, R')$	Time (sec)
Pelvis	306x306x33	306x306x257	3	96333	1475582	0.0652	74
Tooth	306x306x17	306x306x257	4	211489	5709714	0.0370	48
Femur	306x306x17	306x306x257	4	176415	2606773	0.0677	32
Bunny	306x306x33	306x306x257	3	107605	4417735	0.0244	39

TABLE 1
Summary of experimental results.

We apply the above subdivision of sets to the reconstruction of objects from their parallel cross-sections by considering the given cross-sections as the input sets $\{F_i^0\}$ for our subdivision process. It generally suffices to perform only a small number of subdivision refinements (about 4) to obtain resolution adequate for visualization of the resulting object. It is not difficult to rewrite the recursive scheme defined by (17) and (18) as an iterative algorithm.

An example of our reconstruction process is shown in Figure 5. In this example an artificial object is reconstructed from 8 cross-sections using 4 refinement steps. It is easy to observe that this reconstruction is smoother than the one in Figure 4

4. Results

The algorithm described in the previous sections was implemented using MATLAB software environment. The input to the algorithm is a stack of binary images representing the cross-sections of the object. The in-slice (xy) resolution of the images is assumed to be sufficient, and the inter-slice (z) resolution of the stack is successively increased by the refinement process. A high-resolution binary volume is obtained after 3 – 4 refinement steps.

To verify the quality of the reconstruction by our method we use a collection of high-resolution volumetric models. The volumetric models were obtained from triangulated models available at [1], using the voxelization software in [2]. The volumetric models were sub-sampled in the z dimension and then reconstructed to the original resolution by our method.

To measure the quality of the reconstruction we use the *homogeneous symmetric difference metric*, which is defined for two 3D objects A, B by,

$$d_H(A, B) = \frac{\text{Volume}(A\Delta B)}{\text{Volume}(A\cup B)}.$$

This metric is suitable as a measure of the reconstruction quality, since it is indeed a metric on 3D objects and is invariant to scaling ([20], pp. 76-77). In addition, $d_H(A, B) \leq 1$.

Table 4 summarizes the results and running times for different data sets. In all examples, $w = \frac{1}{16}$ is used. For all four data sets we obtained small valued of d_H between the object R and its reconstruction R' . Since $d_H(R, R')$ measures the "relative error" in the reconstruction, our experiments indicated the quality of our method. The running times were measured on a standard 2.66GHz desktop with 32-bit OS. Visual results obtained for different data sets are presented in Figures 6-9. The volumes are visualized using MATLAB 'isosurface' algorithm. To reduce the aliasing effect inherent to visualization of binary volumes, the anti-aliasing filter of [24] was applied to the reconstructed volumes prior to isosurface extraction.

5. Complexity Analysis

Assume that the input is k binary images of resolution n^2 , and the target resolution for the reconstruction is m . For simplicity assume that $m = n$, so the output of the algorithm is a volume of n^3 voxels. By (18), three set averages by Algorithm 1 are to be computed for each constructed slice, so it is performed $O(n)$ times. The Boolean set operations and the labeling of connected components needed by Algorithm 1 have linear complexity in the number of pixels, so they are completed in $O(n^2)$. Since the number of connected components in $A \setminus B$ and $B \setminus A$ is a geometric feature that does not depend on the resolution of the underlying images, we assume it to be $O(1)$. For each connected component, distance maps are computed in $O(n^2)$ by the method in [17]. The reparametrization in (5) is done in $O(1)$ steps of $O(n^2)$. We conclude that the complexity of Algorithm 1 is $O(n^2)$, and therefore the complexity of the reconstruction scheme is $O(n^3)$, which is linear relative to the size of the output.

6. Conclusions

We introduced a novel method for the reconstruction of 3D objects from 2D cross-sections, which is also applicable in higher dimensions. In this work, for the first time, an interpolatory subdivision scheme, generating smooth curves, is adapted to sets and applied to the problem of reconstruction of objects from cross-sections. This way more than two consecutive cross-sections are involved in the reconstruction, which is different from most existing methods.

For the adaptation of the 4-point subdivision scheme to sets, we developed a new weighted average of two sets, interpolating between the two sets for positive weights, and extrapolating from one set beyond the other set when one weight is negative. The new weighted average ensures a gradual change in the symmetric difference when interpolating or extrapolating between two sets.

This work proposes new vistas for application of interpolatory subdivision methods to sets and other geometric objects. As a practical direction for future research we propose to investigate terrain reconstruction from level-lines, using interpolatory subdivision based on our new average of sets.

References

- [1] Aim@shape shape repository. <http://shapes.aim-at-shape.net/>.
- [2] A. H. Aitkenhead. Polygon mesh voxelisation, accessed February 2011. <http://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation>.
- [3] A. Albu, T. Beugeling, and D. Laurendeau. A Morphology-Based Approach for Interslice Interpolation of Anatomical Slices From Volumetric Images. *IEEE Transactions on Biomedical Engineering*, 55(8), 2008.
- [4] G. Barequet, M. Goodrich, A. Levi-Steiner, and D. Steiner. Contour interpolation by straight skeletons* 1. *Graphical Models*, 66(4):245–260, 2004.
- [5] G. Barequet and A. Vaxman. Nonlinear interpolation between slices. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 97–107. ACM, 2007.
- [6] A. Bors, L. Kechagias, and I. Pitas. Binary morphological shape-based interpolation applied to 3-D tooth reconstruction. *IEEE transactions on medical imaging*, 21(2):100–108, 2002.
- [7] D. Cohen-Or, A. Solomovic, and D. Levin. Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics (TOG)*, 17(2):116–141, 1998.
- [8] B. Csébfalvi, L. Neumann, A. Kanitsar, and E. Gröller. Smooth shape-based interpolation using the conjugate gradient method. In *Proceedings of Vision, Modeling, and Visualization*, pages 123–130. Citeseer, 2002.

- [9] N. Dyn and E. Farkhi. Spline subdivision schemes for compact sets with metric averages. *Trends in Approximation Theory*, pages 93–102, 2001.
- [10] N. Dyn and D. Levin. Subdivision schemes in geometric modelling. *Acta Numerica*, 11:73–144, 2003.
- [11] N. Dyn, D. Levin, and J. Gregory. A 4-point interpolatory subdivision scheme for curve design. *Computer Aided Geometric Design*, 4(4):257–268, 1987.
- [12] G. Grevera and J. Udupa. Shape-based interpolation of multidimensional grey-level images. *IEEE Transactions on Medical Imaging*, 15(6):881–892, 1996.
- [13] M. Iwanowski. Morphological Normalized Binary Object Metamorphosis. *Computer Vision and Graphics*, pages 626–632, 2006.
- [14] S. Kels and N. Dyn. The 4-point subdivision scheme of sets. *Technical Report, Tel-Aviv University*, 2010.
- [15] D. Levin. Multidimensional reconstruction by set-valued approximations. *IMA Journal of Numerical Analysis*, 6(2):173, 1986.
- [16] J. Marker, I. Braude, K. Museth, and D. Breen. Contour-based surface reconstruction using implicit curve fitting, and distance field filtering and interpolation. In *Proceedings of the International Workshop on Volume Graphics*, pages 95–102. Citeseer, 2006.
- [17] C. Maurer Jr, R. Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 265–270, 2003.
- [18] S. Raya and J. Udupa. Shape-based interpolation of multidimensional objects. *IEEE transactions on medical imaging*, 9(1):32–42, 1990.
- [19] F. Restle. A metric and an ordering on sets. *Psychometrika*, 24(3):207–220, 1959.
- [20] G. Shephard and R. Webster. Metrics for sets of convex bodies. *Mathematika*, 12:73–88, 1965.
- [21] G. Turk and J. O’Brien. Shape transformation using variational implicit functions. In *International Conference on Computer Graphics and Interactive Techniques*. ACM New York, NY, USA, 1999.
- [22] R. Veltkamp. Shape matching: Similarity measures and algorithms. In *smi*, page 0188. Published by the IEEE Computer Society, 2001.
- [23] D. Wang, O. Hassan, K. Morgan, and N. Weatherill. Efficient surface reconstruction from contours based on two-dimensional Delaunay triangulation. *International Journal for Numerical Methods in Engineering*, 65(5):734–751, 2006.
- [24] R. Whitaker. Reducing aliasing artifacts in iso-surfaces of binary volumes. *vv*, pages 23–32, 2000.
- [25] E. Yakersberg. Morphing between geometric shapes using straight-skeleton-based interpolation, M.Sc.thesis, Technion, Israel, 2006.

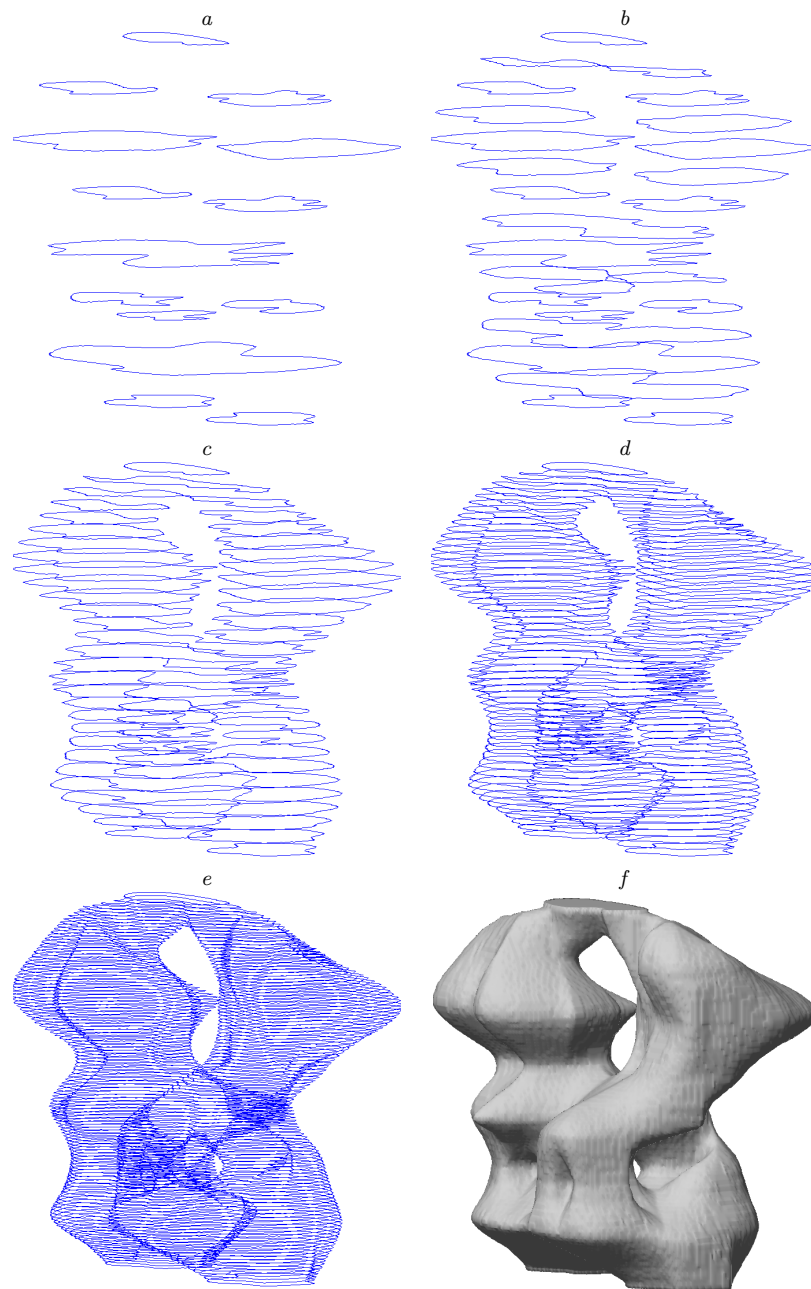


FIG 5. Reconstruction of an object from parallel cross sections: a. the initial cross-sections; b,c,d,e. the refined cross-sections after 1,2,3,4 subdivision steps respectively; f. the visualized object.

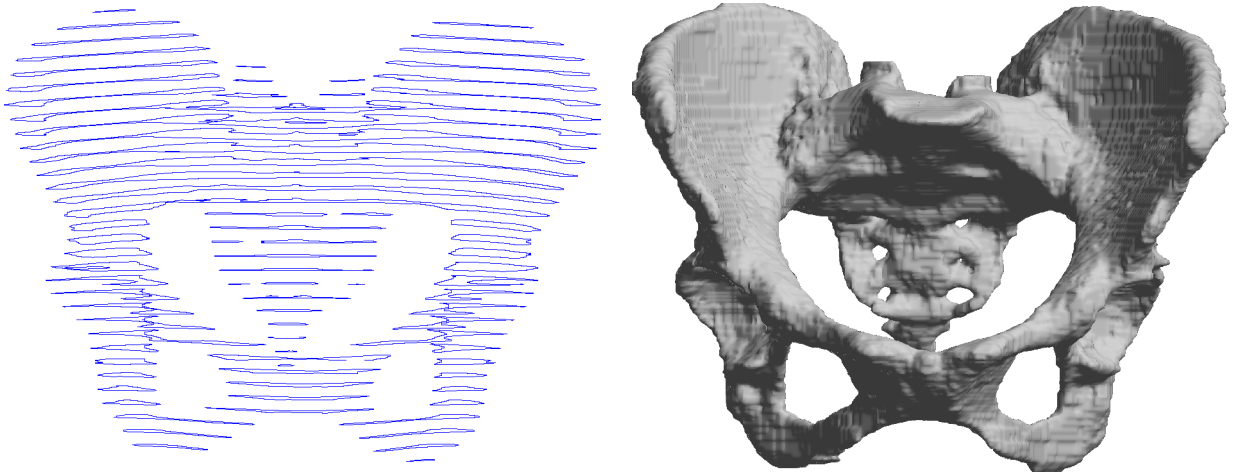


FIG 6. *Reconstruction of a pelvic bone.*

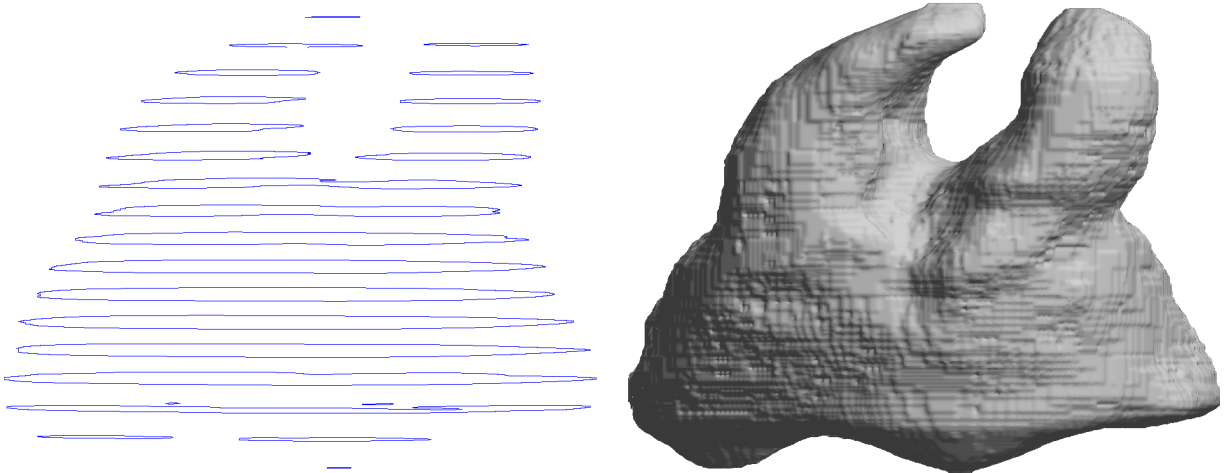


FIG 7. *Reconstruction of a tooth.*

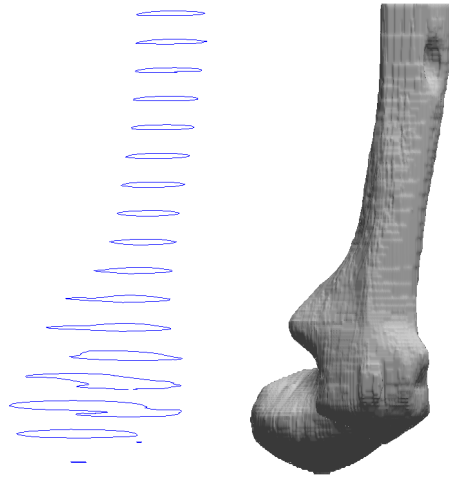


FIG 8. Reconstruction of a femur bone.

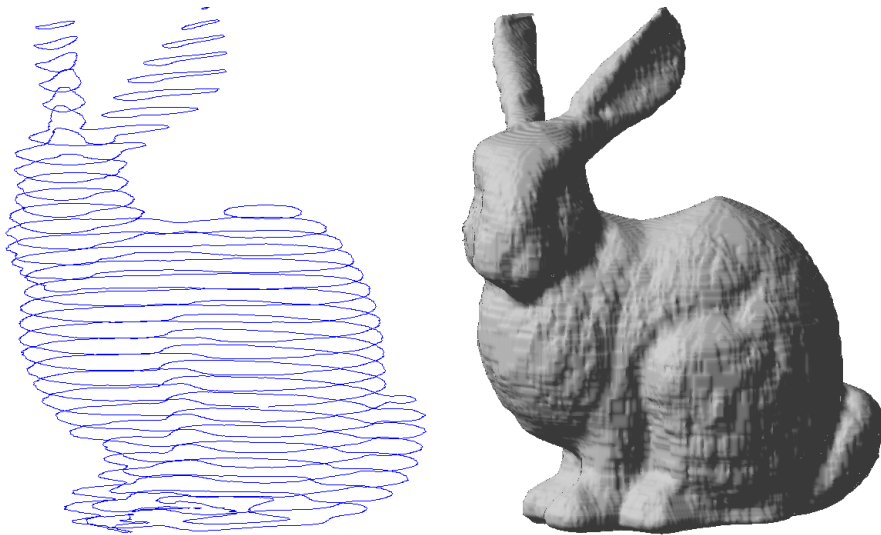


FIG 9. Reconstruction of the Stanford Bunny.