

# Optimization of Basis Functions for Both Reconstruction and Visualization<sup>\*</sup>

Edgar Garduño<sup>\*</sup> and Gabor T. Herman

*Department of Computer Science  
The Graduate Center  
City University of New York  
365 Fifth Avenue  
New York, NY 10016-4309  
USA*

---

## Abstract

Algebraic Reconstruction Techniques (*ART*) for the reconstruction of distributions from projections have yielded improvements in diverse fields such as medical imaging and electron microscopy. An important property of these methods is that they allow the use of various basis functions. Recently spherically symmetric functions (*blobs*) have been introduced as efficacious basis functions for reconstruction. However, basis functions whose parameters were found to be appropriate for use in reconstruction are not necessarily good for visualization. We propose a method of selecting blob parameters for both reconstruction and visualization.

*Key words:* Reconstruction from projections, Kaiser-Bessel functions, Non-cubic grids, Implicit surfaces, Visualization

---

## 1 Reconstruction from Projections

It is often desirable to acquire knowledge of the interior of an object or a body. Unfortunately in most cases it is difficult, or impossible, to directly observe the interior of objects. However, it is possible to get information regarding the structure inside an object from measurements resulting from the interaction of the object with some type of energy.

---

<sup>\*</sup> Supported by NIH Grant HL70472.

<sup>\*</sup> Corresponding author

*Email addresses:* edgargar@alumni.upenn.edu (Edgar Garduño),  
gherman@gc.cuny.edu (Gabor T. Herman).

In this article the measurements are considered to be line integrals (of some spatially-varying physical parameter, related to the local interactions of the object with the energy) through the object from the source to the detector. In practice, a large number of measurements along different lines with different orientations around the object are taken. The aim is to reconstruct the distribution of the spatially-varying physical parameter from the measured data.

In general there are two families of techniques for reconstruction (reconstruction algorithms): *transform* and *series expansion* methods [9]. In this article we are interested in the latter and, particularly, in the so-called algebraic reconstruction techniques (*ART*) because these have proved to be efficacious methods for a number of reconstruction tasks [9,10,12,15,16,18].

In these methods the problem of reconstruction is first discretized by assuming that an approximation of the distribution  $v$  to be reconstructed can be given by a linear combination of known basis functions; that is, as

$$v(r, \phi, \theta) \approx \sum_{j=1}^J c_j b_j(r, \phi, \theta), \quad (1)$$

where  $(r, \phi, \theta)$  are spherical coordinates,  $\{c_j\}$ ,  $j = 1 \dots J$ , is the set of coefficients that has to be determined by the reconstruction algorithm, and  $\{b_j\}$  is the set of known basis functions. The choice of the set of basis functions  $\{b_j\}$  greatly influences the result of the reconstruction algorithm [13,14,20]. The basis functions most commonly used are those which have a unit value inside a cubic volume and zero outside (known as cubic voxels). However, the approximations resulting from using cubic voxels are piecewise constant functions that have undesirable artificial sharp edges; it appears to be more appropriate to use functions with a smooth transition from one to zero.

In the field of reconstruction from projections Lewitt [13,14] and Matej [20] have proposed the use of basis functions, called *blobs*, with spherical symmetry and a smooth transition from one to zero. Blobs are a generalization of a well-known class of window functions in digital signal processing called *Kaiser-Bessel* [13]. The general form of a single blob is:

$$b(m, \alpha, a; r) = \begin{cases} \frac{I_m\left(\alpha\sqrt{1-\left(\frac{r}{a}\right)^2}\right)}{I_m(\alpha)} \left(\sqrt{1-\left(\frac{r}{a}\right)^2}\right)^m, & \text{if } 0 \leq r \leq a, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where  $r$  is the radial distance from the blob center,  $I_m$  denotes the modified Bessel function of order  $m$ ,  $a$  determines the support of the blob (*radius*) and  $\alpha$  is a parame-

ter controlling the blob shape. It is clear that the three parameters  $m$  (a non-negative integer),  $a$  and  $\alpha$  (non-negative real numbers) influence the results yielded by the reconstruction algorithm; therefore, the appropriate selection of them is highly important. The parameter  $m$  allows us to control the continuity of the blob: for  $m > 0$  the blob is a continuous function with  $m - 1$  continuous derivatives [14]. In what follows we will need to make use of the gradients of blobs which implies that we need to have  $m > 1$ . For  $m = 2$ , the blobs are already smooth functions with continuous first derivatives. The extra smoothness provided by larger values of  $m$  has an undesirable consequence: the values of the blobs will be small for a considerable distance inside the boundary. These values do not substantially change the results, but an important fraction of the computation would be spent on them. Therefore, in this paper we set  $m$  equal to 2.

Two justifications, besides the smooth transition from one to zero in a compact region of space, for the selection of blobs as basis functions are that there is a closed analytical formula for the three-dimensional Fourier transform of any blob defined by (2) (in case  $m = 2$ , it is the spherically symmetric function determined by

$$\hat{b}(2, \alpha, a; R) = \frac{(2\pi)^{\frac{3}{2}} a^3 \alpha^2}{I_2(\alpha)} \begin{cases} \frac{I_{\frac{7}{2}}\left(\sqrt{\alpha^2 - (2\pi a R)^2}\right)}{\left(\sqrt{\alpha^2 - (2\pi a R)^2}\right)^{\frac{7}{2}}}, & \text{if } 2\pi a R \leq \alpha, \\ \frac{J_{\frac{7}{2}}\left(\sqrt{(2\pi a R)^2 - \alpha^2}\right)}{\left(\sqrt{(2\pi a R)^2 - \alpha^2}\right)^{\frac{7}{2}}}, & \text{if } 2\pi a R \geq \alpha, \end{cases} \quad (3)$$

where  $J$  is the Bessel function), and that blobs are practically bandlimited [13]. We illustrate this in Fig. 1, which plots the value  $\log\left(\frac{\hat{b}(2, 13.36, 2.40; R)}{\hat{b}(2, 13.36, 2.40; 0)}\right)$  as a function of the frequency  $R$ . It is clear from this figure that  $\hat{b}(2, 13.36, 2.40; R)$  is less than a tenthousandth of its peak value if  $R \geq 1$  and it is less than a millionth of its peak value if  $R \geq 2$ . (Note that some of our variables are not dimensionless; for example  $a$  has dimensionality length and  $R$  has dimensionality inverse length. Throughout this paper we assume that there is a fixed, but unspecified, unit of length.) For evidence that the use of blobs in series expansion methods produces superior results than those produced by transform methods, see [11,12,15,18].

The individual blob functions  $b_j$  of (1) are shifted versions of the blob  $b$  defined in (2). We refer to the set of points  $\{p_j\}$  to which the centers of the blobs are shifted in such a blob representation as a *grid*.

Incidentally, the linear combination of blob-like basis functions approach has also been proposed as a way to model three-dimensional objects in the computer graphics field [3,4,6,22]. Some examples of basis functions with smooth transition from

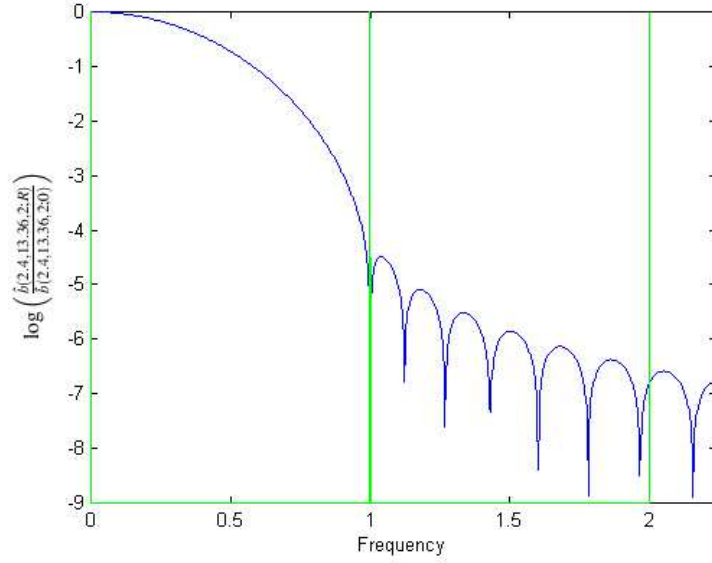


Fig. 1. The Fourier transform of a blob. We plot  $\log \left( \frac{\hat{b}(2,13.36,2.40;R)}{\hat{b}(2,13.36,2.40;0)} \right)$  as a function of the frequency  $R$ .

their maximum to zero are the Gaussian function, used in [3], and multiscale wavelets, used in [22,23].

## 2 Blob Parameters and Grids

The choice of the spatial arrangement of the set of points  $\{p_j\}$  is important for the quality of the final reconstruction. Three grids are of particular interest:

- a. A simple cubic grid (*sc*) is defined by

$$G_\Delta = \{(\Delta x_1, \Delta x_2, \Delta x_3) \mid x_1, x_2, x_3 \in \mathbb{Z}\}, \quad (4)$$

where  $\mathbb{Z}$  is the set of integers and  $\Delta$  is a positive real number (the sampling distance).

- b. A body-centered cubic grid (*bcc*) is defined by

$$B_\Delta = \{(\Delta x_1, \Delta x_2, \Delta x_3) \mid x_1, x_2, x_3 \in \mathbb{Z} \text{ and } x_1 \equiv x_2 \equiv x_3 \pmod{2}\}, \quad (5)$$

where  $\Delta$  is a positive real number.

- c. A face-centered cubic grid (*fcc*) is defined by

$$F_\Delta = \{(\Delta x_1, \Delta x_2, \Delta x_3) \mid x_1, x_2, x_3 \in \mathbb{Z} \text{ and } x_1 + x_2 + x_3 \equiv 0 \pmod{2}\}, \quad (6)$$

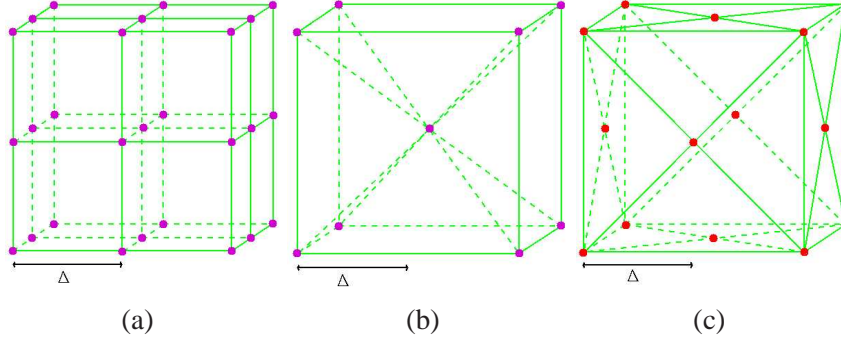


Fig. 2. Points in the simple cubic (a), body-centered cubic (b), and face-centered cubic (c) grids in a  $2 \times 2 \times 2$  portion of space (assuming  $\Delta = 1$ ). The rest of the points can be obtained by filling in space by the most natural repetition of the indicated  $2 \times 2 \times 2$  portion.

where  $\Delta$  is a positive real number.

In order to visualize these grids, we can use a small portion of them and take advantage of their periodic repetition, see Fig. 2.

To accommodate our discussion of parameter optimization, it is useful to introduce some additional notation. Let  $\text{III}_{G_\Delta}$ ,  $\text{III}_{B_\Delta}$ , and  $\text{III}_{F_\Delta}$  denote the (generalized) functions that one obtains by placing (unit-strength) impulses [5] at the grid points of  $G_\Delta$ ,  $B_\Delta$ , and  $F_\Delta$  defined in equations (4), (5), and (6), respectively. It is easy to verify that

$$\mathcal{F}(\text{III}_{G_\Delta}) = \frac{1}{\Delta^3} \text{III}_{G_{\frac{1}{\Delta}}} \quad (7)$$

and

$$\mathcal{F}(\text{III}_{B_\Delta}) = \frac{1}{4\Delta^3} \text{III}_{F_{\frac{1}{2\Delta}}}, \quad (8)$$

where  $\mathcal{F}$  denotes the three-dimensional Fourier transform.

The interest in the grids defined above is that the simple cubic grid is the most used and is the easiest to implement in current computers. It has been shown in [21,24] that the *bcc* grid is the most “efficient” sampling in three-dimensional Euclidean space when a function is bandlimited with a spectrum that is radially symmetric. To illuminate this statement, we consider a distribution  $\mathfrak{v}$  whose Fourier transform  $\hat{\mathfrak{v}}$  has the property that  $\hat{\mathfrak{v}}(R, \Phi, \Theta) = 0$ , if  $R \geq 0.5$  (i.e., the bandwidth of  $\mathfrak{v}$  is 1). Sampling  $\mathfrak{v}$  at points of  $G_1$  (which is the same as multiplying by  $\text{III}_{G_1}$ ) results in a Fourier transform which is the convolution of  $\hat{\mathfrak{v}}$  with  $\text{III}_{G_1}$ ; see (7). Due to the assumed property of  $\hat{\mathfrak{v}}$ , this Fourier transform coincides with the Fourier transform of  $\mathfrak{v}$  within a sphere of radius 0.5 centered at the origin (see Fig. 2(a)), and so  $\mathfrak{v}$  can be unambiguously recovered from the samples. However, the same cannot be guaranteed if sampling is done at points of  $G_\Delta$  with  $\Delta > 1$ , because of the resulting

overlap of the repeated spheres of radius 0.5 in the Fourier transform of the sampled function (*aliasing*). On the other hand, sampling  $\mathfrak{v}$  at points of  $B_\Delta$  results in the spheres repeating with their centers at the grid points  $F_{\frac{1}{2\Delta}}$ ; see (8). As can be seen from Fig. 2(c), to avoid aliasing we may set  $\Delta$  to  $\frac{1}{\sqrt{2}}$  (or anything less). The number of points needed to cover a fixed portion of space is lower when the points come from  $B_{\frac{1}{\sqrt{2}}}$  than when they come from  $G_1$ . This is the sense in which the *bcc* grid is more efficient than the *sc* grid. The *fcc* grid is more efficient than the simple cubic grid but is less efficient than the *bcc* grid. Interestingly the *bcc* and the *fcc* grids are reciprocals of each other in real space and Fourier space, as expressed in (8).

For reconstruction purposes, Matej and Lewitt [19] demonstrated that whenever a linear combination of blobs is employed to obtain an approximation of  $\mathfrak{v}$ , the grid used should be different from the simple cubic grid, and that the *bcc* grid is the most desirable. Therefore, we will consider the *bcc* grid for the set  $\{p_j\}$  where the centers of the blobs  $\{b_j\}$  should be located; i.e., the set is obtained by the intersection of some finite convex region of space with the  $B_\Delta$  of (5).

Having decided that we use  $m = 2$  and the *bcc* grid, there are three parameters to be chosen:  $\alpha$ ,  $a$ , and  $\Delta$ . Clearly, to be able to approximate arbitrary distributions using (1), the value of  $\Delta$  should be small. However, in a fixed volume of space, the number of grid points (and consequently the cost of our reconstruction algorithm) is proportional to  $\frac{1}{\Delta^3}$  and so practical considerations do not allow us to choose  $\Delta$  very small. The cost of reconstruction (in our implementation using footprints [14,20]) is also proportional to  $a^2$ , we soon report on the influence of the size of  $a$  on reconstruction quality. In our implementation, computational cost does not depend on  $\alpha$ , and so this parameter may be chosen purely based on the quality of the resulting reconstructions.

One reasonable criterion for choosing our parameters is that a linear combination of blobs with  $c_j = 1$ , for  $1 \leq j \leq J$ , should be an approximation of a constant valued function. In such a case, the right hand side of (1) is a convolution of the blob  $b$  in (2) with a truncated version of  $\mathbb{I}_{B_\Delta}$ , and so its Fourier transform is approximately  $\frac{\hat{b}}{4\Delta^3} \mathbb{I}_{F_{\frac{1}{2\Delta}}}$ . For this to best approximate the Fourier transform of a constant-valued function (which is an impulse at the origin) it is useful to select  $b$  in such a way that  $\hat{b}(2, \alpha, a; R)$  is zero-valued at the locations of  $F_{\frac{1}{2\Delta}}$  which have the smallest positive distance from the origin; i.e., at the frequency  $R = \frac{1}{\sqrt{2}\Delta}$ . Since  $I_{\frac{7}{2}}$  is not zero-valued and the smallest positive  $x$  for which  $J_{\frac{7}{2}}(x) = 0$  is  $x = 6.987932$ , it follows from (3) and the discussion in this paragraph that, for any  $a$  and  $\Delta$ , a reasonable choice is

$$\alpha = \sqrt{2\pi^2 \left(\frac{a}{\Delta}\right)^2 - 6.987932^2}. \quad (9)$$

Following this approach we reduce the number of unknowns from three to just two, i.e.,  $a$  and  $\Delta$ . Note that  $\alpha$  depends only on the ratio (and not on the actual values) of these two variables.

In Fig. 3 we report on an experiment based on those proposed in [20]. The plots represent the level sets (with indicated values) of the root mean square (*rms*) error between an appropriate constant and the right hand side of (1) (with  $c_j = 1$  for  $j = 1 \dots J$ ) sampled at an appropriate subset of  $G_1$  for various choices of  $\frac{a}{\Delta}$  and  $\alpha$  (we set, for this experiment,  $\Delta = \frac{1}{\sqrt{2}}$ ).

In Fig. 3 we indicate the locus of points for which (9) is satisfied (a hyperbola). As can be seen from Fig. 3, for any fixed value of  $\frac{a}{\Delta}$ , the  $\alpha$  determined by (9) provides a low rms error. If we restrict ourselves to points on the hyperbola indicated in Fig. 3, the rms error for a fixed  $\Delta$  can be decreased by using a higher value  $a$ , but this is at the expense of increased computational cost. Also, this study ignores resolution, which is another important criterion. There is no particular reason to believe (and, in fact, one is likely to suspect the contrary) that the parameters which are good for representing very smooth objects will also results in reconstructions of high resolution. We return to this issue below.

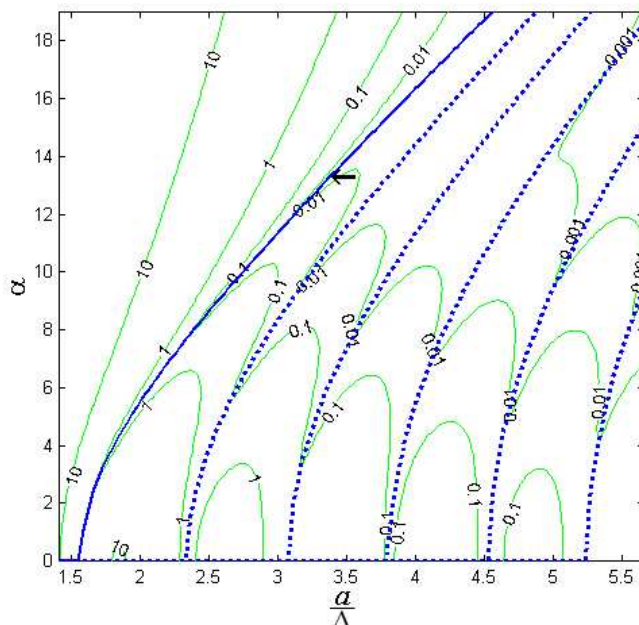


Fig. 3. The root mean square (*rms*) error between a constant-valued function and its approximation by a linear combination of blobs using several values  $\alpha$  and  $\frac{a}{\Delta}$  (with  $\Delta = \frac{1}{\sqrt{2}}$ ). The continuous heavy curve indicates the locus of the points (a hyperbola) which satisfy (9), the broken heavy curves indicate the hyperbolas which correspond to the other zero-crossings of the Fourier transform of  $b$ .

The hyperbola expressed in (9) by the continuous heavy line has been obtained by considering the smallest positive  $x$  for which  $J_{\frac{7}{2}}(x) = 0$ . There are also larger values of  $x$  which have this property, providing us with other blobs  $b$  whose Fourier transforms are zero-valued at the frequency  $R = \frac{1}{\sqrt{2\Delta}}$ . Each one of these values provide us with an alternative hyperbola (just replace the constant in (9) with the new value), these are shown in Fig. 3 by broken heavy lines. From the point of view of the discussion which led us to (9), the alternative versions are equally acceptable and, as can be seen from Fig. 3, they indeed provide us with relatively low rms errors.

### 3 Implicit Surfaces and Visualization

In computer graphics implicit surfaces [4] have been used to represent objects of different topologies and geometries. An *implicit surface*  $S$  is mathematically defined as

$$S = \{(r, \phi, \theta) | v(r, \phi, \theta) = t\}. \quad (10)$$

In many areas of science, (10) is used to visualize a reconstructed distribution described by (1). The assumption is that there is a fixed threshold  $t$  such that the object of interest consists of exactly those points at which the value of the distribution  $v$  is greater than the threshold. If the total volume of the object of interest is known (as is the case in some applications, such as electron microscopy), then  $t$  is uniquely determined by the criterion that  $S$  should enclose exactly the known volume. For computerized visualization of the object of interest it is then sufficient to display its surface  $S$ , as defined by (10).

A suitable method to visualize the surface in (10) is *raycasting*. In one of its forms this technique consists of casting a finite number of rays perpendicular to a plane, typically the computer screen, towards  $S$ ; this form of raycasting produces an orthogonal projection onto the plane. In order to produce a foreshortening effect in the final image (the farther the objects, the smaller they appear in it) it is possible to use a perspective projection in which all the rays cast from the plane intersect in a point called the *center of projection* [6,26]. Because we work with objects in which foreshortening is not important, we present images with orthogonal projection only.

For every ray we need to find the point  $q$  in  $S$  nearest to the plane and compute its distance and the normal to  $S$  at  $q$  (these are used to assign an intensity value on the computer screen [6,26]). In practice, finding the points  $q$  is computationally expensive. In general there is no prior estimate of how far  $q$  is from the plane. Based on an approach in [3], we designed the following methodology. We first do a preprocessing of the set  $\{p_j\}$  at the end of which, for every point on the plane from

which we need to cast a ray, we have the list of those grid points (arranged in order of increasing distance from the plane) whose associated coefficients can possibly influence the value of the distribution  $v$  anywhere along the ray. (These grid points all lie within a cylinder of radius  $a$  whose central axis is the ray in question.) This preprocessing is easily done by identifying on the plane the shadows of the blobs centered at the grid points, one-by-one in an appropriate order. In locating  $q$  for a particular ray, we make use of the associated list of grid points. For all grid points in the list (recall that these are arranged in order of increasing distance from the plane), we evaluate  $v$  at the projection of the grid point onto the ray (for this we need the blob coefficients for only a few grid points, all of which are at a similar position in the list), until we find (if ever) two consecutive projections  $q_a$  and  $q_b$  such that the value of  $v$  is below the threshold at  $q_a$  and is above it at  $q_b$ . Then  $q$  is located by a binary search between  $q_a$  and  $q_b$  (for this we need the coefficients of only those blobs which were used for calculating  $v$  at  $q_a$  and at  $q_b$ ).

Assuming that the approximation in (1) is exact, we know that  $v$  is a continuously differentiable function and that the gradient of  $v$ , at any point, is given by

$$\nabla v(r, \phi, \theta) = \sum_{j=1}^J c_j \nabla b_j(r, \phi, \theta). \quad (11)$$

The set  $\{c_j\}$  is produced by the reconstruction algorithm and we have closed formulas to compute  $\nabla b_j$  [13]. The representation obtained by (10) and raycasting is therefore an accurate representation of the object of interest, limited only by the reconstruction and thresholding processes.

#### 4 Selection of Blobs for Visualization

The principles described in Sections 1 and 2 have been applied in the field of Structural Analysis of Proteins by Transmission Electron Microscopy [2,15–17,25]. In particular, the authors of [2] obtained a set of coefficients  $\{c_j\}$  by applying ART to a set of projection images originating from a protein (the macromolecular complex DnaB·DnaC) imaged by a transmission electron microscope, using parameters  $\alpha$ ,  $a$  and  $\Delta$  satisfying (9). When we used the resulting set  $\{c_j\}$  to produce a visual representation of the surface of the macromolecular complex DnaB·DnaC by the raycasting method explained above, we found, to our surprise, that the surface representation had artifacts that were not observable in the slice-by-slice presentation of the reconstructed distribution, see Fig. 4. Clearly, the parameters which were considered "optimal" for reconstruction are not particularly good for visualization.

Just as the approach of approximating functions by linear combinations of blobs, the representation of surfaces using linear combinations of blobs requires finding a

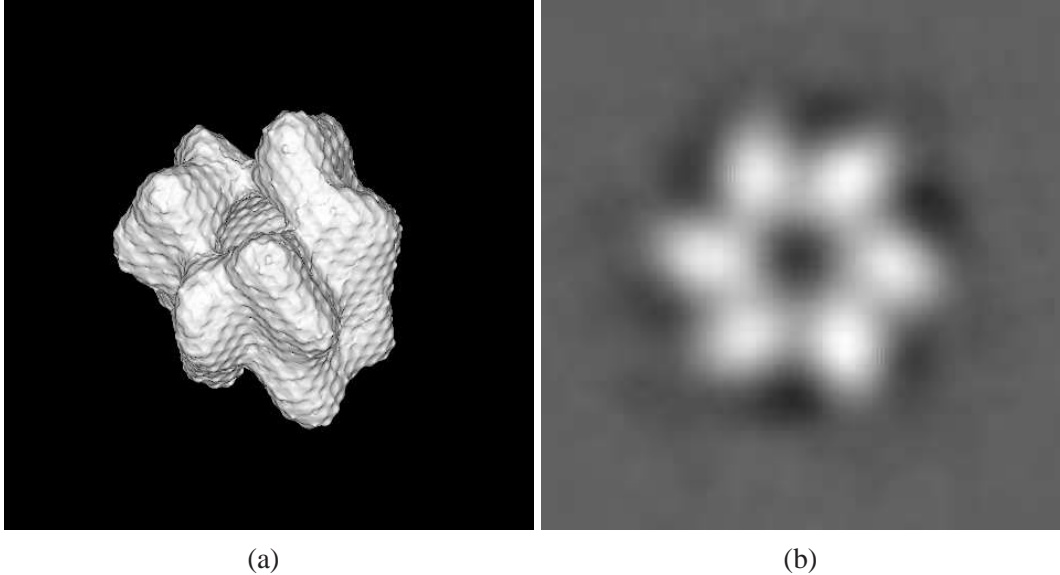


Fig. 4. (a) Surface representation of the macromolecular complex DnaB·DnaC reconstructed using ART and blobs with parameters  $a = 1.25$ ,  $\alpha = 3.60$  (throughout this paper we report on  $a$  and  $\alpha$  to an accuracy of 0.01, however the values actually used by our programs were always calculated so that (9) is satisfied to the accuracy of our computations) and placed on a *bcc* grid with  $\Delta = \frac{1}{\sqrt{2}}$  [2] (the distance between pixels for the display is 0.25). (b) A central slice from the reconstruction by ART of the macromolecular complex DnaB·DnaC with the same parameters as for (a) (the values of  $v$  are evaluated on a simple cubic grid with  $\Delta = 1.00$ ).

“good” set of parameters for the blobs and for the grid of blob centers. In Fig. 5 we illustrate the enormous influence of the values of  $a$  and  $\alpha$  on the appearance of the resulting surface.

In order to study further the effect of blob parameters for visualization, we used an approach similar to that in Section 2, but now we aimed at minimizing the error between a surface and its approximation, as measured by the difference between the surface normals. For this test we selected a distribution  $v_s$  with a constant value 1 inside a sphere and 0 outside. We then simulated the random conical tilt scheme of data collection (a common scheme in electron microscopy [7]) to generate the projections from  $v_s$ . The projections were utilized to create reconstructions using ART with different values for  $\frac{a}{\Delta}$  and  $\alpha$  (keeping again  $\Delta = \frac{1}{\sqrt{2}}$ ). For each set  $\{c_j\}$  produced by a reconstruction algorithm, raycasting was used to create a visualization of the implicit surface of the reconstructed sphere at threshold 0.5 (four of the resulting surface displays, for the same choices of parameters as were made for Fig. 5, are shown in Fig. 6).

For every ray we computed the angle between the normal to the true spherical surface and the normal to the implicit surface in the reconstructed distribution. We define the *rms error* to be the norm of the vector whose components are these angles (for all display points for which the casted ray crosses both surfaces). The results

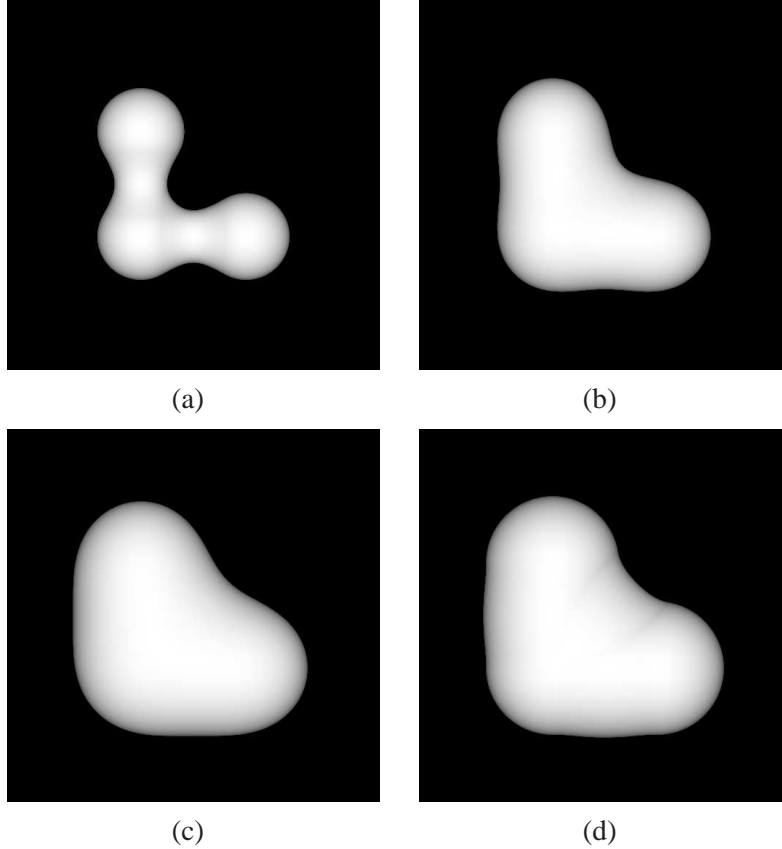


Fig. 5. In all cases the distribution  $\mathfrak{v}$  is defined by (1) using the grid  $B_{\frac{1}{\sqrt{2}}}$  with blob coefficient 1 at the points  $(0, 0, 0)$ ,  $(\sqrt{2}, 0, 0)$  and  $(0, \sqrt{2}, 0)$  and blob coefficient 0 at all other points. The displayed surfaces are defined by (10) with  $t = 0.5$ . The values of  $a$  and  $\alpha$  are (a) 1.25 and 3.60 (same as for Fig. 4(a)), (b) 2.40 and 13.36, (c) 3.20 and 18.85 and (d) 1.65 and 0.00. The first three pairs of  $(a, \alpha)$  satisfy (9) and the fourth is computed with the value of  $x$  for the second zero crossing of  $J_{\frac{7}{2}}(x)$ .

are displayed in Fig. 7 in a manner similar to the display in Fig. 3.

Ignoring for now the exact choice of  $\alpha$ , what we see in Fig. 7 is that, generally speaking,  $\frac{a}{\Delta}$  should be neither too small (this results in bumps as in Fig. 6(a)) nor too large (this results in a blurred-out sphere as in Fig. 6(c)). To allow us to make a definite choice, we propose the following criterion:  $a$  should be chosen as small as possible consistent with achieving that if two blobs at nearest grid points in the grid  $B_{\Delta}$  are given coefficients 1 with all other blobs given coefficients 0, then the implicit surface thresholded at  $t = 0.5$  should enclose a convex set. Such implicit surfaces (for the values of  $a$  also used in Figs. 5, and 6 and with  $\alpha$  determined by (9), assuming that  $\Delta = \frac{1}{\sqrt{2}}$ ) are shown in Fig. 8. Fixing  $\Delta$  to be  $\frac{1}{\sqrt{2}}$  and using an  $\alpha$  which is determined by (9), we find that the smallest  $a$  which satisfies our new criterion is 2.40 (this corresponds to (b) in Figs. 5 and 6). The corresponding  $\alpha$  is 13.36 and the location of this  $(\frac{a}{\Delta}, \alpha)$  pair is indicated by arrows in Figs. 3 and 7.

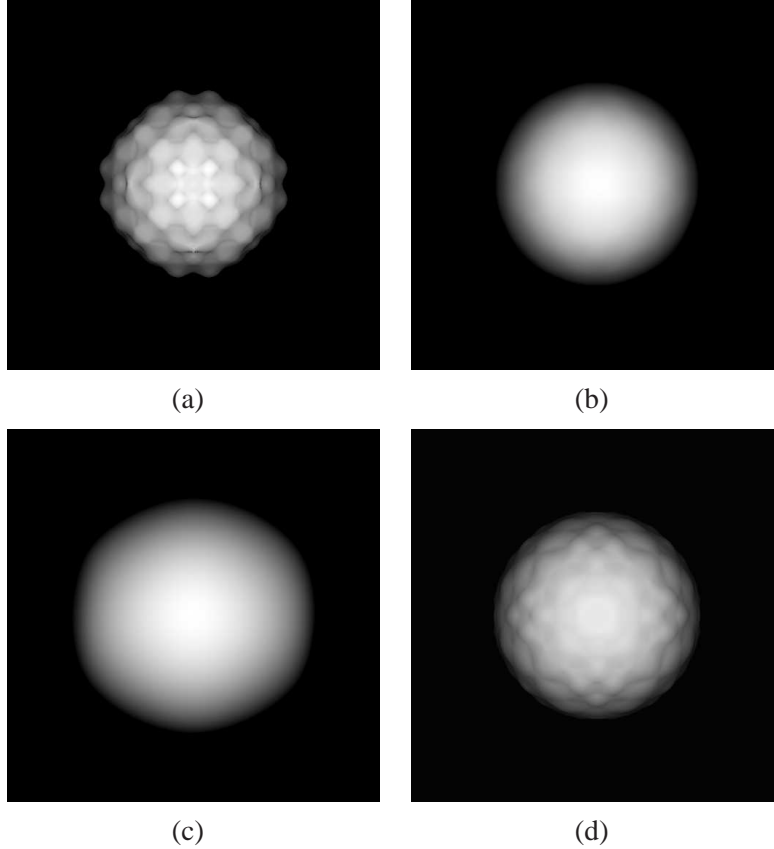


Fig. 6. Visualizations of the implicit surface ( $t = 0.5$ ) of reconstructions of a sphere. For the choices of the parameters in cases (a), (b), (c) and (d), see the caption of Fig. 5.

We see that, as long as we insist that (9) be satisfied, the value of  $a$  determined the criterion stated in the previous paragraph will be a fixed multiple of  $\Delta$ . (This can be easily derived by looking at (2) and expressing both  $a$  and  $r$  in it as multiples of  $\Delta$  and using the  $\alpha$  specified with (9). So the criterion, together with (9), uniquely determines  $\frac{a}{\Delta}$  (it is 3.39) and  $\alpha$  (it is 13.36). We still have the freedom of choosing  $\Delta$ ; its choice to a large extent depends on the nature of our measurements.

We could argue that the criterion just described could be combined with (9), but using the hyperbola associated with the next zero-crossing of the Fourier transform of (2), to obtain a lower rms error. For example, some points on the hyperbola associated with the second zero-crossing of  $b$  in Fig. 7 results in lower rms errors. Disappointingly, applying the previously-stated criterion ( $a$  should be chosen as small as possible consistent with achieving that if two blobs at nearest grid points in the grid  $B_\Delta$  are given coefficients 1 with all other blobs given coefficients 0, then the implicit surface thresholded at  $t = 0.5$  should enclose a convex set) for this hyperbola  $\left(\alpha = \sqrt{2\pi^2 \left(\frac{a}{\Delta}\right)^2 - 10.41711^2}\right)$  results in the pair  $a = 1.63$ ,  $\alpha = 0.00$  (any smaller value of  $a$  would result in an imaginary value of  $\alpha$ ); this already produces a convex surface, shown in Fig. 8(d). It can be seen in Fig. 6(d) that the

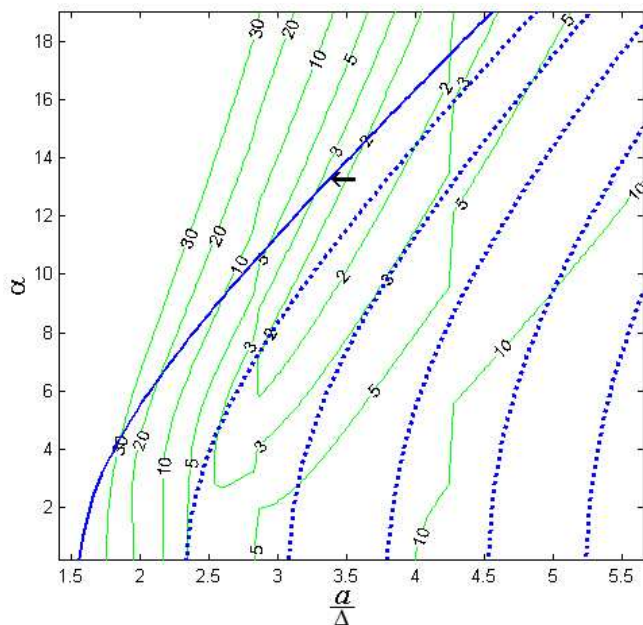


Fig. 7. The rms error between analytic normals to a sphere and normals to the implicit surface of its reconstruction at each display points for which the ray casted crosses both surfaces (see Fig. 6). The continuous heavy curve indicates the locus of the points (a hyperbola) which satisfy (9). The broken heavy curves indicate the hyperbolas which correspond to the other zero-crossing of the Fourier transform of  $b$ .

implicit surface produced with such parameters is a bad approximation of a sphere. Not surprisingly, for these values, the rms error presented in Fig. 7 is outside the region of “good” values. For this reason, we stick with the previously obtained parameters ( $a = 2.40$ ,  $\alpha = 13.36$ ) based on the hyperbola produced by the first zero crossing of  $b$ .

We illustrate the performance of these choices of the parameters used in conjunction with the electron microscopic data from the macromolecular complex DnaB·DnaC. The resulting implicit surface is shown in Figs.9(b); it is superior to the previously shown surface of Fig. 4(a), which is reproduced here as Fig. 9(a). It is also worth noting that it is just about impossible to anticipate the radical differences between the surface displays in Fig. 9 from the traditional slice-by-slice presentation of the reconstructed distributions, see Fig. 10.

## 5 Comparison with Explicit Surface Visualization

While the implementation of raycasting to visualize implicit surfaces is straightforward, such visualization is usually computationally demanding and slow because of the constant search for the intersecting points  $q$ . Polygon-projection methods are an

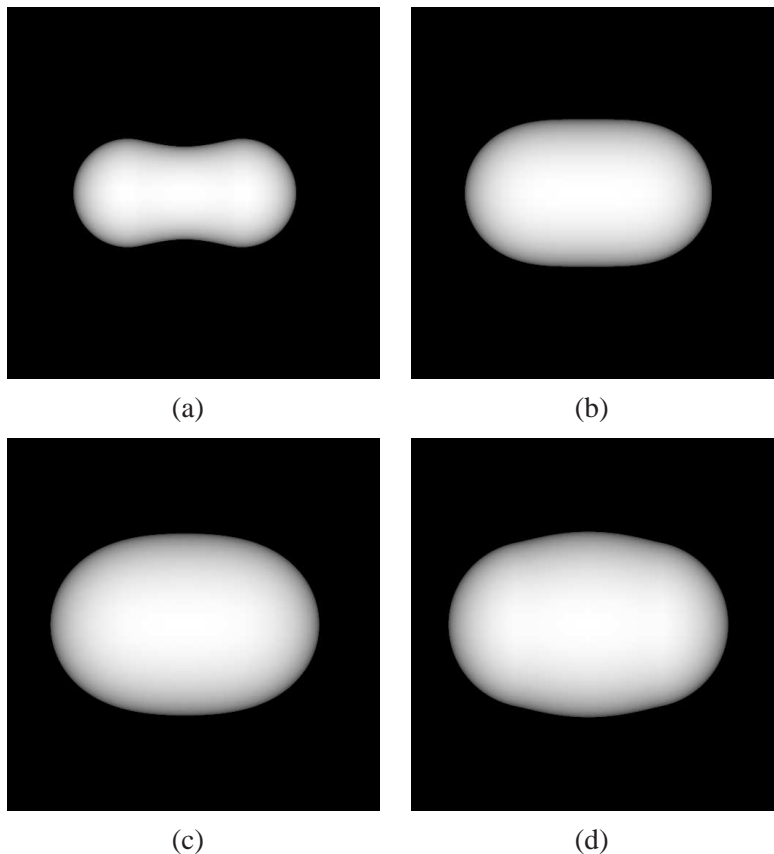


Fig. 8. Representations of the implicit surface at level  $t = 0.5$  for the combination of two blobs whose centers are immediate neighbors in the  $bcc$  grid  $B_{\frac{1}{\sqrt{2}}}$  and whose coefficients are 1. The parameters of the blobs match those in Figs. 5 and 6.

alternative because of their fast performance. These methods explicitly approximate a surface by a collection of polygons [6,8,26]. Here we compare the performance of the raycasting method described in Section 3 with that of the polygon-projection method implemented in the visualization software OpenDX<sup>TM</sup> [1].

In order to apply OpenDX<sup>TM</sup> to our reconstructions, we first need to evaluate the values at points of some grid (for our illustrations we used the simple cubic grid). For reasons already explained in Section 2, if  $B_{\frac{1}{\sqrt{2}}}$  was used for reconstruction, it is appropriate to estimate the reconstructed values, using (1), at points of  $G_1$ . Just like the raycasting method, the polygon projection method of OpenDX<sup>TM</sup> requires the specification of a threshold, based on which it automatically calculates the polygons which form the explicit surface to be displayed.

We applied this approach to the reconstructions of the macromolecular complex DnaB·DnaC whose central slices are reported in Fig. 10. (By the way, these slices are displays of the estimated values at points of  $G_1$ .) The calculation of the values at points of  $G_1$  took 10 s for the blobs with  $a = 1.25$  and  $\alpha = 3.60$  and 56 s for the blobs with  $a = 2.40$  and  $\alpha = 13.36$ . (All times are for a Pentium 4<sup>TM</sup> based

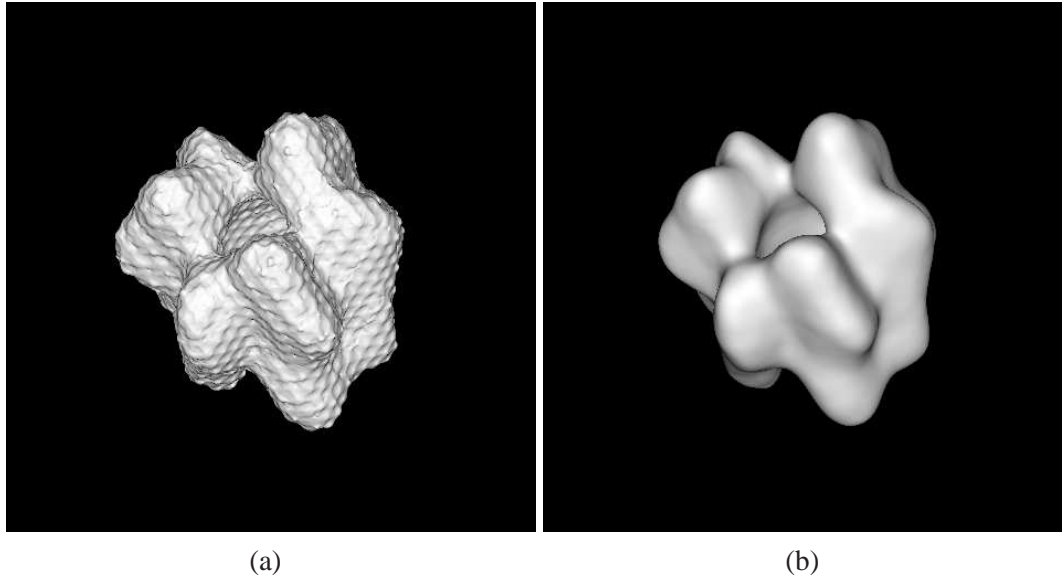


Fig. 9. Representation of the implicit surface for the macromolecular complex DnaB·DnaC. The reconstructions using ART and visual representations of the DnaB·DnaC were created with (a) parameters  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 1.25$  and  $\alpha = 3.60$  as used in [2], (b) parameters  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 2.40$  and  $\alpha = 13.36$ .

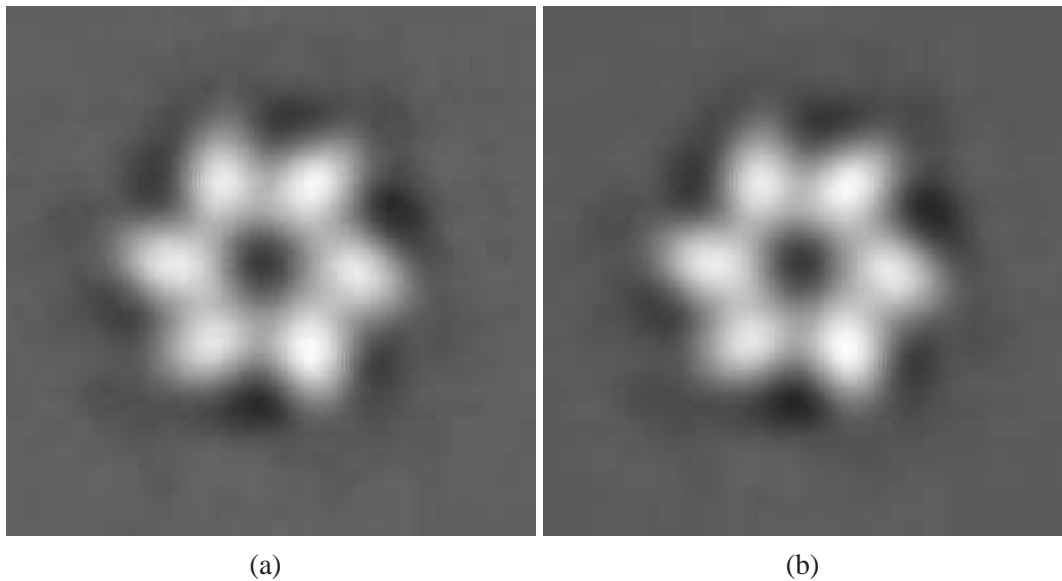


Fig. 10. Central section of the reconstructions of the macromolecular complex DnaB·DnaC by ART with (a) parameters  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 1.25$  and  $\alpha = 3.60$ , (b) parameters  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 2.40$  and  $\alpha = 13.36$ . Both images are central slices of the discretized density function  $\upsilon$  using the sc grid with  $\Delta = 1.00$ .

computer, 2 GHz, 512 Mbytes of RAM, under Linux©.) In both cases, the computation of the explicit surface took 2 sec. The display of the surface, once computed, is essentially instantaneous. These times are much better than what is needed for raycasting at the same resolution ( $480 \times 480$  pixels): 13 min 19 s for the blobs

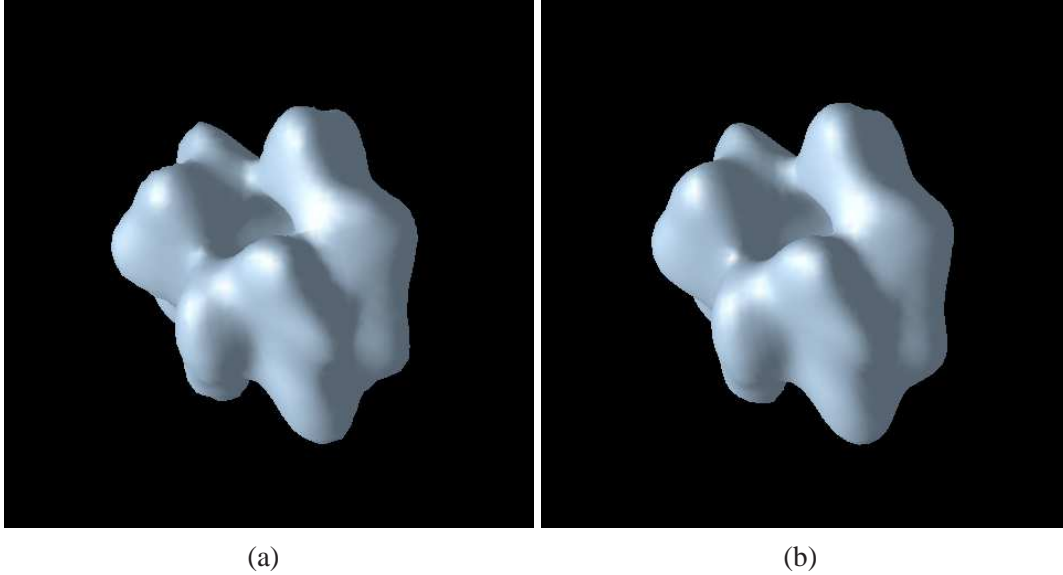


Fig. 11. Representations of the explicit surfaces obtained from the reconstructions of the macromolecular complex DnaB-DnaC using OpenDX with a  $128 \times 128 \times 128$  voxelized distribution over the  $sc$  grid ( $\Delta = 1.00$ ). Parameters used for reconstruction were: (a)  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 1.25$  and  $\alpha = 3.60$  and (b)  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 2.40$  and  $\alpha = 13.36$ . It is clear that, for this example, the general visualization software hides some important details; for a comparison with the corresponding outputs of the raycasting method, see Fig. 9.

with  $a = 1.25$  and  $\alpha = 3.60$  and 1 h 35 min 50 s for the blobs with  $a = 2.40$  and  $\alpha = 13.36$ . However, the reconstruction times (the calculation of the  $\{c_j\}$  by ART) are 32 h 15 min 36 s and 65 h 22 min 48 s, respectively, in these two cases and the time needed for specimen preparation and the collection of electron microscopic data is measured in weeks. Hence the quality of the visualizations may very well be considered more important than the computer time needed to produce them.

The results produced by the polygon-projection method in OpenDX<sup>TM</sup> are presented in Fig. 11. All parameters (for the blobs, the grid, the threshold, the assumed orientation of the surface, etc.) were selected to be the same as those in the corresponding displays in Fig. 9. With the computer graphic display methodology embedded in OpenDX<sup>TM</sup>, the explicit surface displays of Fig. 11 appear to be smoother than the implicit surface displays of Fig. 9, which is an advantage in case (a), but seems to be a disadvantage in case (b), which is the case that uses the blob parameters recommended by us in Section 4. In order to further increase the rendering resolution, it was necessary to sample the density function at the points of  $G_{\frac{1}{4}}$ . By using this resolution, the artifacts in Fig. 9(a) are also visible in the OpenDX rendering using reconstruction parameters  $a = 2.40$  and  $\alpha = 13.36$ , see Fig. 12. However, the increase of the sampling implies an increment in time (60 seconds) and memory (800 Mbytes) to generate the explicit surfaces using this software, but the time to render an image is still essentially instantaneous.

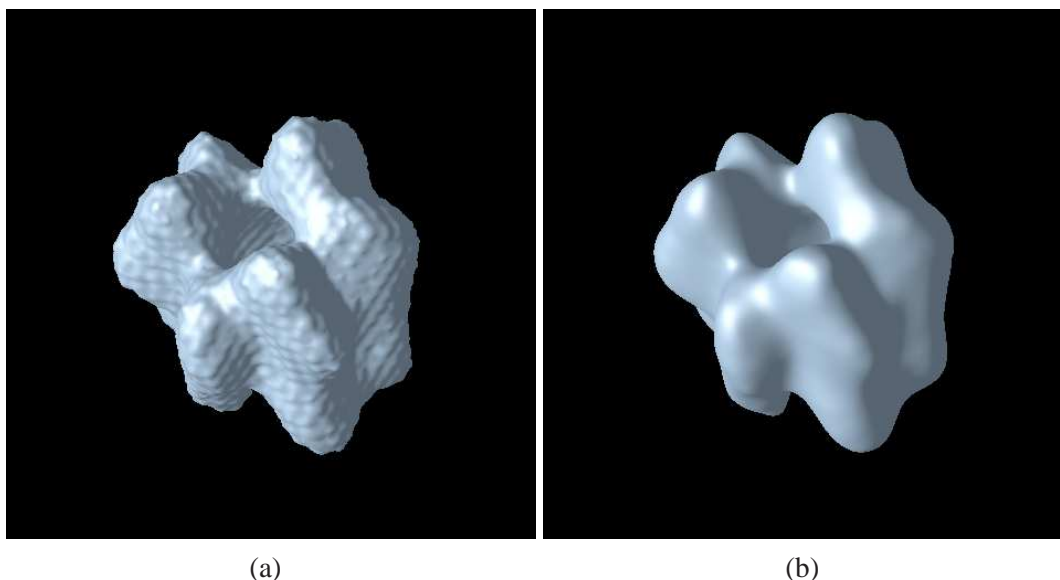


Fig. 12. Representation of the explicit surface obtained from the reconstructions of the macromolecular complex DnaB-DnaC using OpenDX with a  $400 \times 400 \times 400$  voxelized distribution over the *sc* grid ( $\Delta = 0.25$ ). (a) Parameters used for reconstruction were  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 1.25$  and  $\alpha = 3.60$ . (b) Parameters used for reconstruction were  $\Delta = \frac{1}{\sqrt{2}}$ ,  $a = 2.40$  and  $\alpha = 13.36$ . For this representation it is necessary to increase four times the memory allocated by OpenDX.

It is difficult to make a visual comparison between the results obtained by our raycasting method shown in Fig. 9 and those produced by OpenDX with a finely discretized density function shown in Fig. 12, mainly due to the difference in lighting models. However, we can see that the result in Fig. 12(a) is a smoother representation of the same surface that is shown in Fig. 9(a); we speculate that such smoothing is due to lack of information in the discretized density function used as input to OpenDX™. Similarly, it seems that the surface in Fig. 9(b) is a sharper version of that in Fig. 12(b). However, for a fair evaluative comparison it would be necessary to make the lighting models the same and to use a mathematically described object whose features are known.

As stated at the end of Section 3, the accuracy of the visualization using raycasting is limited only by the quality of the reconstruction (the  $\{c_j\}$ ) and the accuracy of the threshold. The polygon-projection methods bring an additional source of inaccuracy into the process: the approximation of the implicit surface by a collection of polygons. Thus, if the reconstruction parameters and the threshold have been well chosen, we can expect that raycasting will be the more reliable visualization tool. Whether or not this is worth the very considerable additional computing time (and the consequent loss of capability of real-time interactive examination of the resulting surfaces) is highly dependent on the application; it is impossible to know ahead of time whether or not the more reliable visualization will lead to increased scientific knowledge.

## Acknowledgements

We are grateful for the help and advice received from Robert Lewitt and Samuel Matej regarding this research. We also thank Carlos Oscar Sánchez Sorzano for his comments that greatly improved this manuscript. We appreciate the assistance of Montserrat Bárcena in providing us with data and information regarding the macromolecular complex DnaB·DnaC.

## References

- [1] G. D. Abram and L. Treinish, “An extended data flow architecture for data analysis and visualization,” in *Proceedings of the IEEE Computer Society Visualization '95*, pp. 263–269, IEEE Computer Society, 1995.
- [2] M. Bárcena, T. Ruiz, L. E. Donate, S. E. Brown, N. E. Dixon, M. Radermacher, and J. M. Carazo, “The DnaB·DnaC complex: A structure based on dimers assembled around an occluded channel,” *European Molecular Biology Organization Journal*, vol. 20, pp. 1462–1468, 2001.
- [3] J. F. Blinn, “A generalization of algebraic surface drawing,” *ACM Transactions on Graphics*, vol. 1, pp. 235–256, 1982.
- [4] J. Bloomenthal, *Introduction to Implicit Surfaces*. San Francisco: Morgan Kaufmann, 1997.
- [5] R. N. Bracewell, *Two-Dimensional Imaging*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1995.
- [6] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice*. New York: Addison-Wesley Co., 2nd ed., 1996.
- [7] J. Frank, *Three-Dimensional Electron Microscopy of Macromolecular Assemblies*. New York: Academic Press, 1995.
- [8] E. Galin and S. Akkouche, “Incremental polygonization of implicit surfaces,” *Graphical Models*, vol. 69, pp. 19–39, 2000.
- [9] G. T. Herman, *Image Reconstruction from Projections: The Fundamentals of Computerized Tomography*. New York: Academic Press, 1980.
- [10] G. T. Herman, “Algebraic reconstruction techniques in medical imaging,” in *Medical Imaging, Systems Techniques and Applications - Computational Techniques* (C. T. Leondes, ed.), pp. 1–42, Amsterdam: Gordon and Breach Science Publishers, 1997.
- [11] F. Jacobs, S. Matej, R. M. Lewitt, and I. Lemahieu, “A comparative study of 2D reconstruction algorithms using pixels and optimized blobs applied to Fourier rebinned 3D data,” in *Proceedings of the 1999 International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, pp. 43–46, Egmond aan Zee, The Netherlands, 1999.

- [12] P. E. Kinahan, S. Matej, J. S. Karp, G. T. Herman, and R. M. Lewitt, "A comparison of transform and iterative techniques for a volume-imaging PET scanner with a large acceptance angle," *IEEE Transactions on Nuclear Science*, vol. 42, pp. 2281–2287, 1995.
- [13] R. M. Lewitt, "Multidimensional digital image representations using generalized Kaiser–Bessel window functions," *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, vol. 7, pp. 1834–1846, 1990.
- [14] R. M. Lewitt, "Alternatives to voxels for image representation in iterative reconstruction algorithms," *Physics in Medicine and Biology*, vol. 37, pp. 705–716, 1992.
- [15] R. Marabini, G. T. Herman, and J. M. Carazo, "3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs)," *Ultramicroscopy*, vol. 72, pp. 53–65, 1997.
- [16] R. Marabini, E. Rietzel, R. Schröder, G. T. Herman, and J. M. Carazo, "Three-dimensional reconstruction from reduced sets of very noisy images acquired following a single-axis tilt schema: Application of a new three-dimensional reconstruction algorithm and objective comparison with weighted backprojection," *Journal of Structural Biology*, vol. 120, pp. 363–371, 1997.
- [17] R. Marabini, M. C. San Martin, and J. M. Carazo, "Electron tomography of biological specimens," in *Contemporary Perspectives in Three-Dimensional Biomedical Imaging* (R. J. L. Coatrieux, ed.), vol. 30 of *Studies in Health Technology and Informatics*, pp. 53–77, IOS Press, 1997.
- [18] S. Matej, G. T. Herman, T. K. Narayan, S. S. Furuie, R. M. Lewitt, and P. E. Kinahan, "Evaluation of task-oriented performance of several fully 3D PET reconstruction algorithms," *Physics in Medicine and Biology*, vol. 39, pp. 355–367, 1994.
- [19] S. Matej and R. M. Lewitt, "Efficient 3D grids for image-reconstruction using spherically-symmetrical volume elements," *IEEE Transactions on Nuclear Science*, vol. 42, pp. 1361–1370, 1995.
- [20] S. Matej and R. M. Lewitt, "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Transactions on Medical Imaging*, vol. 15, pp. 68–78, 1996.
- [21] H. Miyakawa, "Sampling theorem of stationary stochastic variables in multi-dimensional space," *Journal of the Institute of Electronic and Communication Engineers of Japan*, vol. 42, pp. 421–427, 1959.
- [22] S. Muraki, "Volumetric shape description of range data using "Blobby Model"," *Computer Graphics*, vol. 25, pp. 227–235, 1991.
- [23] S. Muraki, "Multiscale volume representation by a DoG wavelet," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 109–116, 1995.
- [24] D. P. Petersen and D. Middleton, "Sampling and reconstruction of wave-number-limited functions in  $N$ -dimensional Euclidean spaces," *Information and Control*, vol. 5, pp. 279–323, 1962.

- [25] M. C. San Martin, C. Gruss, and J. M. Carazo, "Six molecules of SV40 large T antigen assemble in a propeller-shaped particle around a channel," *Journal of Structural Biology*, vol. 268, pp. 15–20, 1997.
- [26] A. H. Watt, *3D Computer Graphics*. Reading, Mass.: Addison-Wesley Pub. Co., 2nd ed., 1993.