Ultramicroscopy 11 (1983) 307–314
North-Holland Publishing Company

SHORT NOTE

STEREOGRAPHIC REPRESENTATION OF THREE-DIMENSIONAL DENSITY DISTRIBUTIONS

Marin VAN HEEL

Fritz-Haber-Institut der Max-Planck-Gesellschaft, Faradayweg 4–6, D-1000 Berlin 33, Fed. Rep. of Germany

Received 14 April 1983

1. Introduction

The representation of three-dimensional density data in a manner that is transparent to the human observer is a problem that has a number of solutions. The solutions are necessarily of an approximate nature since the human visual system is not capable of understanding complicated three-dimensional (3D) density distributions by just looking at them with two eyes [1]. In our "world", virtually all objects are opaque, and we are accustomed to interpreting the three-dimensional shape of the surface of objects.

All solutions, therefore, rely on the definition of one or more "surfaces" ("contours") in the 3D data set which are presented to the observer in one way or another. One of the established techniques is the stacking of contoured sections (slices of the 3D data set) with transparent plastic spacer plates (cf. ref. [2]). This technique is used both in X-ray crystallography and in 3D electron microscopy. Although (stereo) photographs of "contour stacks" are sometimes presented as the result of a reconstruction, it is more common practice to present "balsa-wood" models of the object (cf. refs. [2,3]). Balsa-wood models are made by cutting out sheet material of the correct thickness at the position of a particular contour throughout the sections. The cut-outs of all sections are glued together to form a 3D model which can then be photographed (in stereo if desired).

The algorithm promoted here allows the calculation of "photographs of the model" without building a model. The technique allows rapid visual evaluation of results of 3D reconstructions at various density levels. Apart from being time-saving, this procedure has the advantage that the models need not be buildable, that is, the object may consist of subunits that do not touch each other at all; connectivity is not needed as nothing has to be glued together.

Comparable algorithms have appeared in the literature over the past few years (cf. ref. [4]), but these techniques have not yet been applied to the field of X-ray crystallography of proteins or 3D reconstructions of electron microscopical data (see, however ref. [5]).

Note that the output of the algorithm are digital images! In order to display digital images one needs a so-called raster-scan image display system, which is a computer peripheral device capable of producing a grey-tone image on a TV monitor. More common than image displays are vector graphics display systems which are capable of drawing vectors (lines) given the begin and end coordinates of the vectors; these systems are not suitable as output device for this stereo algorithm!

An attempt has been made to keep this paper self-contained such that programs can be written based on the information given here. A symbolic computer program to calculate stereo projections is included.

2. Stereo projection algorithm

The algorithm proposed here relies on the definition of a contour or a surface of the 3D density distribution, which surface is "projected" in different directions. Prerequisite for the correct working

0304-3991/83/0000-0000/$03.00

of the algorithm is that the "inside" and the "outside" of that part of the 3D density distribution of interest (the "object") can be distinguished from each other by characteristic density values. The simplest situation – which is used for our demonstrations – is that in which the density inside the object is higher than the density in its environment.

The "inside" versus "outside" decision can then be made for any particular "voxel" (3D image element) by comparing its density value with a predefined threshold value. If our prerequisite is not directly valid, preprocessing of the 3D data becomes necessary, a situation that will be discussed below.

In our algorithm (see the appendix for a symbolic program), each section of the 3D data set is treated separately. All the voxel density values inside a section are checked to determine whether the point lies inside or outside the object of interest. If a voxel V (see fig. 1) is found to lie inside the object, the distance VP from this point to its orthogonal projection P on a particular projection plane is determined, and a grey value is calculated that is high if the distance VP is short and low if that distance is large. If the projection point P already had a density value higher than the newly calculated one, that means that in scanning through the voxels in this section we already had encountered a voxel V′, also inside the object, which was closer to P than V.

Since in this algorithm the object is assumed to be fully opaque for voxel-density values higher than the threshold value (and transparent for lower values), in such a situation the voxel V′ shields off the influence of voxel V: V is invisible. Therefore, if P already had a density value higher than the newly calculated one, the old value prevails. The

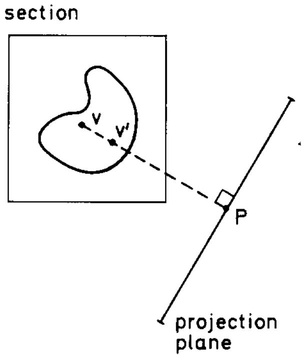
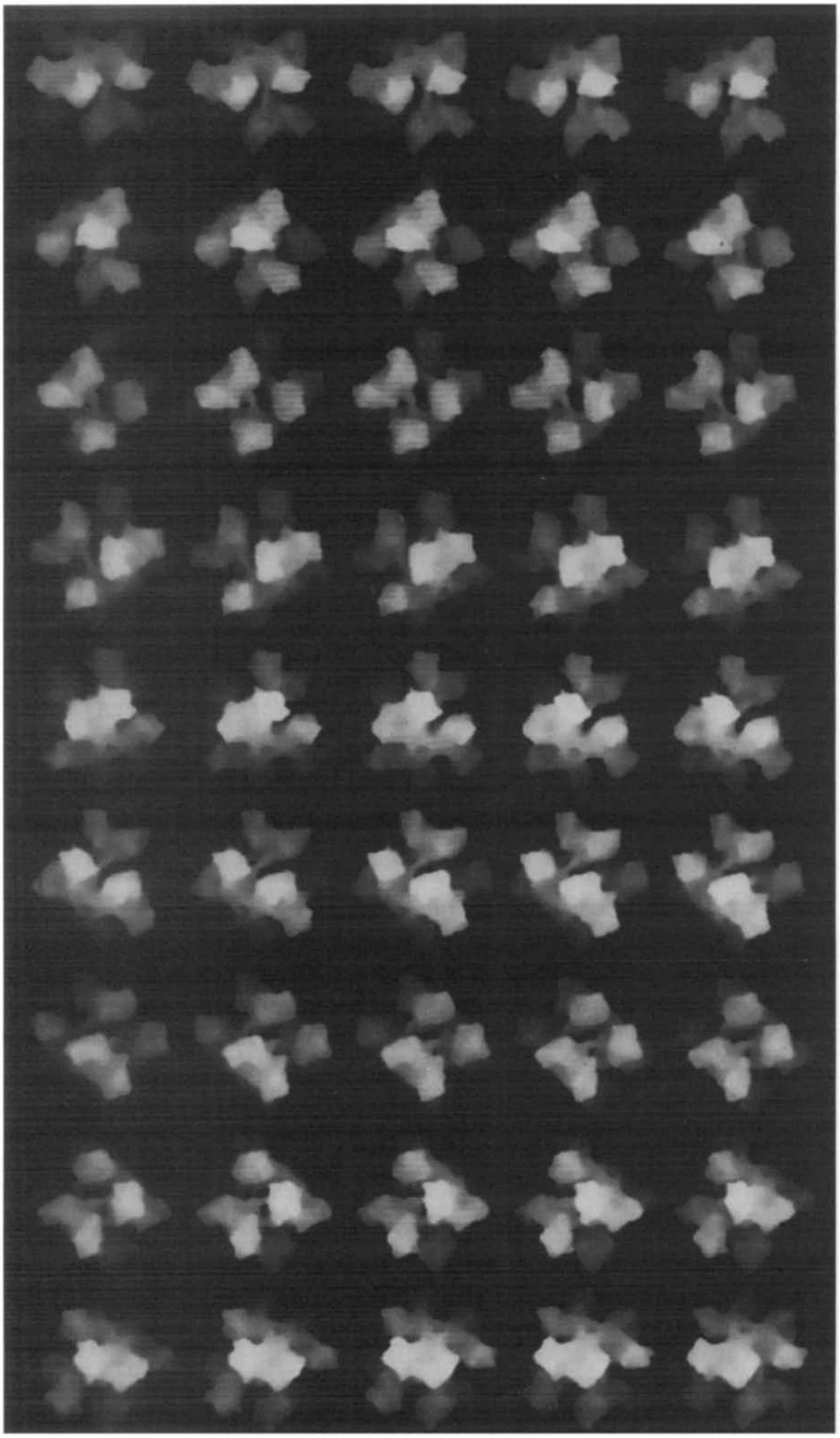


Fig. 1. Voxel V′ prevents voxel V from contributing to the projection in a certain direction as the distance to the projection V′P is shorter than the distance VP. This property of the algorithm accounts for the removal of hidden surfaces.

projection point P should actually be seen as the sampling point in the projection line that is closest to the "mathematical" projection point of the voxel considered (see also note 4!).

After a full scan through the 3D section, only those points inside the object that are closest to the projection plane – the "surface" of the object oriented towards the projection plane – contribute to the density distribution in that plane. The algorithm thus removes the contributions from "hidden voxels" which is equivalent to "hidden line removal" procedures in vector graphics. Since the density in the projection plane depends on the distance from this plane to the surface of the object, the brightness of the image is a code for the closeness of any particular part of the object to the projection plane; this technique is known as "depth queueing" (cf. ref. [4]). Note that for the calculation of many projections simultaneously, the input section only has to be scanned once (one pass over the input data); this is important in the

Fig. 2. A continuous sequence of stereo projections. Every two neighbouring images can be combined to a stereoscopic impression either by using a stereo veiwing aid or by training both eyes to look at two neighbours simultaneously. Angles between two successive projections are 8°. The object imaged is the electron-density distribution of the hemocyanin of *Panulirus interruptus* as determined by X-ray diffraction techniques [7]. The data were pretreated by a 3D filtering procedure. This hemocyanin molecule consists of six bean-shaped subunits in a 32 point-group symmetry [7]. The original 3D data had the three-fold axis parallel to the Z-axis of the data set (the Z-axis corresponds to the vertical axis in the stereo images). To create more interesting stereo images, the 3D volume was rotated over 45°. The direction of view in the top-left image of this sequence makes an angle of 45° with the three-fold axis which passes in between the three "beans" in the top half of the molecule; it also makes an angle of 45° with one of the two-fold axes of the structure.



light of efficient data handling in the computer.

When pairs of projection images are calculated with a projection angle difference of about 4 to 20° (cf. ref. [1]), it is possible to present these pairs as stereo pairs. A more complete way of presenting the 3D data set can be obtained by calculating all the projections, 360° around the object, with a constant angular interval (4 to 20°) between every two neighbour images.

The observer can combine any two neighbour images in such a sequence into a 3D impression; he sees a whole field of stereo images representing the object seen from "all" sides. Fig. 2 is such a set of continuous stereo projections.

In our algorithm the projection lines (like the line VP in fig. 1) are parallel to each other (orthogonal projection); this implies that there is no perspective generated by this algorithm. The reason for this is that for real perspective the projection lines are no longer restricted to a single section which makes the computational organization considerably more complex. The algorithm was implemented in the framework of the "IMAGIC" image processing software system [6] as it is implemented on the Fritz-Haber-Institute's VAX 11/780 computer under the VMS operating system.

3. Stereo viewing

When viewing an object at "infinity" the (human) eye lenses are relaxed and the eye axes are almost in parallel. When observing an object at close range, however, the eye lenses are made stronger, and at the same time the optical axes of the eyes are made to coincide at the point of attention and are thus no longer parallel to each other. When focussing at objects at various distances, the human eye-brain system apparently is used to follow a "program" in which the strength of the eye lenses and the angle between the optical axes of the eyes are strongly coupled.

To create a stereo impression from a pair of printed stereo images, such as the ones printed in this paper, without using any stereo viewing equipment, the observer has to focus his eyes at close range, with both eyes looking at two different images. This implies that the eye axes have to be more or less parallel while at the same time the eye lenses are strongly activated – an abnormal combination! Stereoscopic viewing is facilitated by using viewing aids which restore a more normal viewing situation. In its simplest form such a stereo viewing aid consists of two positive lenses which take over the close-range focussing from the eye lenses. Therewith the eye lenses can be relaxed, and the normal "infinity viewing" situation is restored. Seeing stereo without such a viewing aid requires experience, as the observer's visual system has to be "reprogrammed". A well known trick that can be used to gain such experience is to place a postcard between the two stereo pairs such that the right eye cannot see the left-eye image and vice versa, and then try to integrate the images into a single visual impression. For access to other methods of presenting stereoscopic images see the references in ref. [1].

4. Preprocessing of the 3D data

If the prerequisite that the inside of the "object" can be discriminated from the outside by characteristic density values is not fulfilled, we will have to manipulate the original 3D data such as to create new 3D data with the desired properties for stereo projections. One typical situation is that in which the object of interest is surrounded by other objects of comparable density values; due to the hidden-surface-removing properties of the algorithm, the surrounding objects will render the interesting object invisible. In the case of periodical objects, for example (two- or three-dimensionally ordered molecules), one is usually only interested in obtaining views from one single molecule, excluding all of its neighbours. Our test data used to generate fig. 2 are the result of X-ray diffraction experiments [7] and originally also contained neighbouring molecules which had to be masked out. For that purpose all sections – the original data had the three-fold axis of the molecule perpendicular to the plane of the sections – were added up, and this total sum was contoured interactively [6]. The contour was used to impose a cylindrical mask, with a cylindrical cross-section with the shape of the interactively determined mask, on the original 3D data volume. More straightforward cylindrical masks with a circular cross-section can often be used successfully to study interesting parts of the 3D data set. More complicated is the case in which the "object" of interest has only a different texture than the environment. In such cases the range of densities inside and outside the object may well be identical, and it thus becomes impossible to discriminate between "inside" and "outside" on the basis of voxel density. A straightforward 3D generalization of the "variance-image" operator [8] can be used to convert texture differences into density differences in the variance image. The 3D variance image can be entered into the stereo projection algorithm directly or, alternatively, the variance image can be used (after thresholding) to mask off uninteresting areas of the original 3D data set.

In addition to the cylindrical mask mentioned earlier, the X-ray data were low-pass-filtered through multiplication of the 3D Fourier transform with a Gaussian mask to emphasize the general shape of the molecule rather than the finer structures.

A low-pass filtering (post-processing) of the stereo images can be used to reduce somewhat the visibility of the original sampling grid in the calculated images [4].

Note that, since in this stereo image representation the grey value of an image point is used as a code for the distance to the surface of the object, raster-scan image displays with a limited number of grey values (4 to 16) may severely limit the quality of the image representations! In my own experience, a system capable of representing 256 different grey values gives the best results. The stereo algorithm was successfully used for the representation of electron microscopical 3D reconstructions [9] as well as for the representation of various oligomeric assemblies of the hexameric molecule shown in fig. 2 [10].

5. Conclusion

The algorithm promoted here provides a fast and easy way of evaluating and presenting the results of three-dimensional reconstructions.

Acknowledgements

I would like to thank Wil Gaykema for providing the three-dimensional electron density data used for the example. The author is recipient of the EMBO long term fellowship ALTF-99-1982.

Appendix. Stereo projection algorithm

The algorithm is written in a (non-existing) symbolic programming language. The stereo-projection lines, corresponding to one particular section of the data volume, are kept in a two-dimensional array.

```
LOOP 1   SECTION = 1 to Nsect    {Over the sections in 3D data volume}
   {Initialize output projection arrays on minus infinity}
   Dens_proj (all elements, all projections) = - Very_Large
   LOOP 2    LINE = 1 to N       {Over section lines}
      X_input = LINE   -   X_origin
      LOOP 3  POINT = 1 to N     {Over voxels in a line}
         Y_input   = POINT - Y_origin
         {Density values of the 3D data volume are typically read
         into central memory per section or per line}
         Voxel_density = Density (POINT,LINE,SECTION)
         IF   (Voxel_density > Threshold) then
            LOOP 4   PROJ = 1 to NP {over all stereo projections}
               {Rotation matrix :}
               X_proj    = cos (φ(PROJ)). X_input - sin(φ(PROJ)). Y_input
               Y_proj    = sin (φ(PROJ)). X_input + cos(φ(PROJ)). Y_input
               Proj_dens = X_proj  {Or another function: Func(X_proj)}
               Y_coord    = Y_proj + Y_proj_origin
               {Hidden voxel removal is implicit in following lines}
               IF  (Dens_proj(Y_coord,PROJ) < Proj_dens) then
                   Dens_proj (Y_coord,PROJ) = Proj_dens
               ENDIF
            END LOOP 4
         ENDIF
      END LOOP 3
   END LOOP 2
   {Write out the NP stereo projection lines to the
   corresponding NP stereo projection images}
END LOOP 1
```

Notes:

(1) The calculation of the sines and cosines should not be actually done in the inner loop of the algorithm but should be pre-calculated before entering LOOP-1 for all the projection directions. The values can best be stored in arrays of length NP (number of projections).

(2) The handedness of the stereo projections should be checked by displaying the stereo projections of a structure with a known handedness. The handedness depends on the definition of the coordinate systems used as well as on the way data are read in and out of the central memory. The angular directions of the stereo projections depends on the rotation matrix (the signs of the sine functions) used in the program and on the definition of the function:

Proj dens = Func (X_Proj).

Handedness of the projections should be verified after implementation.

(3) Care has to be taken that the stereo projections are presented to the observer in their correct order, that is, the left eye should be presented the "left-most" projection and the right eye the "right-most" projection. If this order is accidentally reversed the "pseudoscopic" situation is created in which what should be "near" to the observer is actually perceived as being the farthest away.

(4) The algorithm, as presented above, suffers from moiré-type interference between sampling rasters. If the object to be imaged has surfaces at an angle of about 45° with respect to the rectangular sampling raster in the section planes, the sampling points along this surface appear at a sampling interval that is about 1.4 times larger than the sampling raster of the stereo-projection image in the direction perpendicular to that surface. As a consequence, points "inside" the surface of the object can influence the stereo images as the "1.4" gaps between the sampling points are not small enough to shield off some voxels that should in effect be hidden (see fig. 1). The net result is that at specific angles, typically around 45°, vertical striping can occur in the images. A potential remedy to this problem is the widening of the influence of a voxel on the projection as a function of the angle of the stereo projection. A good candidate as a widening function is the (FORTRAN) function:

1./ MAX(ABS(SIN(ϕ)), ABS(COS(ϕ))).

Equivalent moiré effects affect the results of real-space 3D reconstruction algorithms, and solutions have been proposed in that field of research (cf. ref. [4]).

References

[1] M.V. King, in: Three-Dimensional Ultrastructure in Biology, Methods in Cell Biology, Vol. 22, Ed. J. Turner (Academic Press, 1981).

[2] T.L. Blundell and L.N. Johnson, Protein Crystallography (Academic Press, 1976).

[3] R. Henderson and P.N.T. Unwin, Nature 257 (1975) 28.

[4] G.T. Herman, Image Reconstruction From Projections (Academic Press, 1980).

[5] M. Radermacher and J. Frank recently developed a similar algorithm (personal communication).

[6] M. van Heel and W. Keegstra, Ultramicroscopy 6 (1981) 113.

[7] E.J.M. van Schaick, W.G. Schutter, W.P.J. Gaykema, A.M.H. Schepman and W.G. Hol, J. Mol. Biol. 158 (1982) 11.

[8] M. van Heel, Ultramicroscopy 8 (1982) 331.

[9] M. van Heel, Three-Dimensional Reconstruction of the E.coli 30S Ribosomal Subunit, 41st Annual EMSA Meeting, Phoenix, AZ, 1983, submitted.

[10] M. van Heel and E.F.J. van Bruggen, Quaternary Structure of Arthropod Hemocyanins Studied by Image Analysis of Electron Micrographs, in preparation.