Three-dimensional skin reconstruction by vector sequence alignment and morphing

Albert Cardona* and Volker Hartenstein

Molecular Cell Developmental Biology, University of California Los Angeles, 621 Charles E. Young Dr. South, 90015 CA

ABSTRACT

The analysis of the three-dimensional arrangement of cells and tissues is fundamental for the understanding of their individual contribution to the whole organism. High-resolution microscopy outputs seriated two-dimensional images in which, due to the experimental limitations in the labeling, the objects of interest are hardly segmentable by automated means. On the other hand, the outcome of manual segmentation is a set of two-dimensional profiles (i.e. closed 2D curves) which cannot be trivially assembled into a three-dimensional skin.

Here we report on a python module for Blender which takes as input a set of registered 2D profiles and outputs a 3D mesh. The program interprets curves as strings of vectors and aligns them in a process akin to gap-enabled, nucleic acid sequence alignment. The alignment results in optimal faces and enables the intercalation of morphed profiles which ensure a smooth transition between consecutive but highly divergent profiles.

We describe the process of manually segmenting interesting structures on serial images, importing the closed curves to Blender and the algorithmic details in generating the mesh. The output mesh illustrates the segmented fruit fly and Macrostomum brains in 3D, and allows for its inspection and measurement of surfaces.

In addition, we introduce the uses of simple 3D tubes and spheres for the modeling of neuronal structures and their visualization in Blender. The uses of the aforementioned curve comparator algorithm in generating consensus 3D models and as a tool for identifying structures is also discussed.

1 INTRODUCTION

Biology is all about understanding changes in 3D space through time. How can spatial relationships be represented other than in a 3D environment? They can't. Yet we biologists have been publishing our work as sets of 2D images for a long time, only very recently using snapshots of 3D models as supporting visual aids. The believe that 2D images acquired by cameras attached to microscopes are the most faithful representation of the real sample is deeply rooted in our minds, sprouting from the anthropomorphic misconception that any representation close to what our eyes can see must be the most accurate.

1.1 Microscopical techniques

Only two techniques are mainstream for the visualization of structures deep in an opaque sample: electron and confocal microscopy.

Transmission electron microscopy (TEM) embeds the sample in plastic, performs ultrathin sections (60 nm) and then projects an electron beam through the section. Electrondense regions will appear darker. In the acquired image, a grayscale gradient condenses the information contained in 60 nm of sample.

Confocal microscopy, on the other hand, relies on fluorescent dyes. The sample is visualized as a whole mount, and the laser light projected into the sample is returned by the excited fluorochromes. By means of an adjustable pinhole, only the light of a defined Z level in the sample is captured into an image.

By iterating over consecutive sections or Z levels, both techniques enable the collection of seriated images which contain condensed thin slices of the complete volume of the sample.

Yet microscopy, as any other technical field, succeeds only by taking in heavy trade-offs, often assumed and overlooked. The most obvious problem in seriated images is the condensation of a thin volume into a 2D image. The solution of acquiring sections as thin as possible hits practical limits very quickly, i.e. signal to noise ratio decreases and the amount of images to manage increases.

1.2 Three-dimensional modeling

Seriated images are the primary source of 3D reconstructions of biological samples. There are two main approaches to the problem: a pixel-based approach, and a vectorial contour segmentation approach.

A pixel-based approach consists of stacking together all images in a series and applying lower- and upper transparency boundary limits. In this fashion, a volume emerges by stripping off undesired voxels. This approach works optimally when the signal is sharply separated from the background, which is rarely if ever the case in biological samples. Examples of software implementing this approach to various degrees are the Visual Toolkit (VTK), Amira (Amiravis) and ImageJ 3D Viewer.

A contour-based approach consists of manually or semiautomatically defining 2D contours (also known as profiles or outlines) on each section, and then constructing a 3D mesh from them. This approach has the potential of generating very clean 3D representations, often artifactually too clean and smoothened, and often grossly incorrect due to limitations in the applied meshing algorithms (in particular when confronting highly divergent consecutive contours). Examples of software implementing this approach are Imod, Amira (Amiravis) and MorphMesh (for Blender).

1.3 Two proposed solutions

We describe here two solutions to the above limitations in the generation of 3D models.

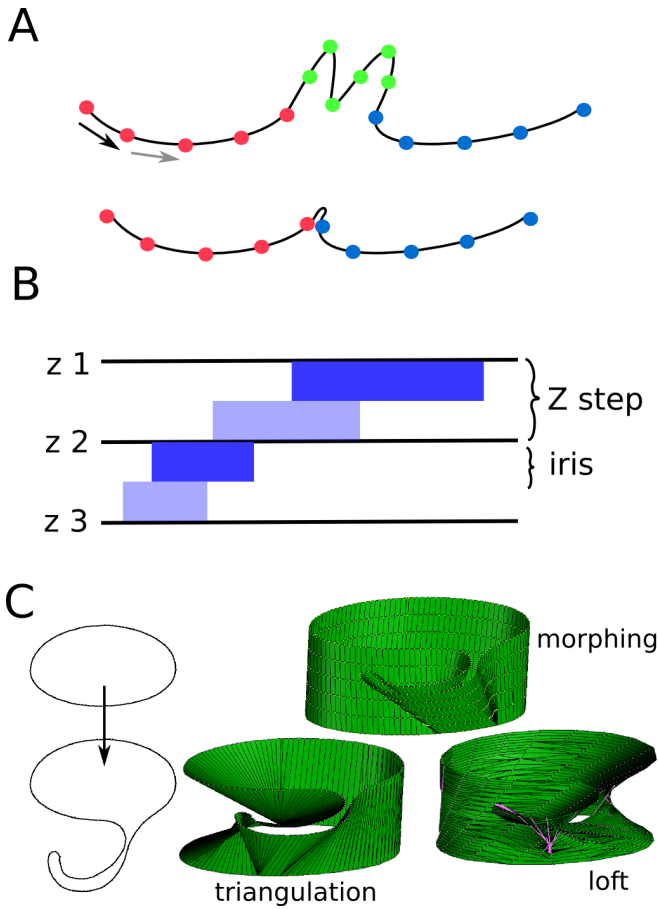*to whom correspondence should be addressed: cardona@ucla.edu

A

B

C

Fig. 1. A The program resamples the profiles and represents them as strings of vectors of equal length. Red and blue segments are identified as very similar (mutations); the green segment as a set of deletions. **B** In confocal microscopy, when the iris defines a thickness of the optical section thinner than the Z step interval at which images are taken, two consecutive profiles (in dark blue) may be misinterpreted as non-continuous. The interpolation feature of the program corrects the issue and ensures a smooth transition. **C** Two divergent profiles characteristic of neuronal profiles (left) are matched with much greater accuracy (top) than by simple triangulation (lower left) or lofting (Maya; lower right).

First, we describe an algorithm to generate meshes from stacked 2D contours that generates nearly perfect meshes, and thus requires no postprocessing. The introduction of typical artifacts, resulting from iterative decimation and subsurfacing of meshes, is thus avoided.

Second, we provide the tools to generate purely abstract shapes in 3D in the form of tubes and spheres. These simple shapes have the advantage of being sketched rapidly while providing a good overview of the structures of interest, in particular neuronal structures.

The meshing algorithm and the tube-making routines have been packed in a GPL python C module for Blender, accessible from a python script within a Blender script window. The curves and spatial coordinates are obtained from user input on specially designed ImageJ plugins, in particular TrakEM2 (Cardona, 2006).

2 MESHING: THE PROCESS OF SKIN RECONSTRUCTION

The problem of skin reconstruction is formulated as, first, the pairing of profiles; second, the intercalation of interpolated profiles if necessary; and third, the proper matching of the coordinates for the generation of triangular and/or quadrangular faces.

The program reported here is based on the string-matching algorithm described by Wagner and Fischer, 1974, as applied to vectors by Jiang et al., 2002, which we apply to both the creation of interpolated profiles and the generation of optimal faces.

2.1 Obtaining and stacking the profiles

A stacked set of two-dimensional profiles defines the three-dimensional boundaries of a single object in the sample. In our practical uses for Drosophila and Macrostomum brain modeling, pre-stacked profiles are obtained as sets of Bezier curves by means of the 3D Editing ImageJ plugin (http://www.pensament.net/java/), in which only one profile exists for each object and section, and TrakEM2 (Cardona, 2006), where linking relationships clearly indicate the ordering of the profiles.

2.2 Resampling the profiles

The list of points defining a profile must be resampled to homogenize the point interdistance. After resampling, a profile is represented as a string of vectors of equal length and a starting point. Thus, profiles are represented as strings of comparable, abstract elements, each encoding a quantum of shape information (fig. 1a).

2.3 Sliding the starting point for an optimal match

The generation of a skin is formulated as the creation of triangles between a pair of profiles. But where is one to start, in the case of closed profiles?

The distance of the first string to all the possible representations of the second string, starting at $v_0, v_1, ...$ to v_{n-1}, is computed by comparing the differences between any two pair of vectors on both strings. The maximum penalty results from two vectors being exactly at 180° from each other (Jiang et al., 2002). The method generates a matrix of cumulative values, where the lower right value is the Levenshtein's distance between the compared profiles. The optimal starting point i will be that for which the distance of the first string, from v_0 to v_{n-1}, to the second string reordered as v_i to v_{n-1}, v_0 to v_{i-1} for $0 >= i < n$, is minimal.

2.4 Constructing the skin

The matrix leading to the Levenshtein's distance provides, by applying a reverse search (Wagner and Fischer, 1974), the sequence of editions (mutations, deletions and insertions) necessary to morph the first profile to the second. By applying the editions in a weighted manner (Jiang et al., 2002) to the pair of profiles, an arbitrary number of interpolated profiles can be generated.

The string matching algorithm is independent of the relative position in the plane of any two profiles to match. Unless both profiles lay nearly perfectly on top of each other, the skin generated in this manner may sport very squinted, and thus undesirable, faces. Theoretically speaking, generating an infinite number of interpolated profiles solves the above problem to perfection. In practice, we have found by inspection that the fourth root of the Levenshtein's distance provides a reasonable, data-driven number of profiles to interpolate, that guarantees a reasonably smooth skin. The generation of badly
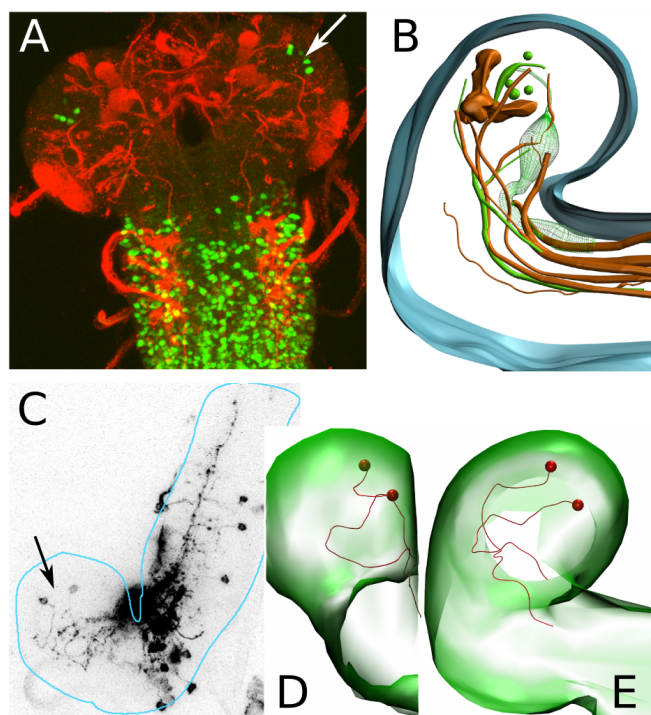
Fig. 2. Confocal Z-projections and 3D models of a Drosophila larvi central nervous system. **A** Fasciclin II immunostaining (in *red*) and GFP expression (in *green*) driven by an unknown Gal4-expressing insertion. Dorsal view, anterior is *up*. **B** 3D model of **A**, showing the surface of the nervous system (*blue*), the mushroom body (*upper red structure*) and several neuronal tracts modeled as tubes, plus 4 GFP-positive cell bodies modeled as spheres (corresponding to *arrow in A*). **C-E** DiI fluorescent dye (inverted) and anterior and lateral views of the corresponding 3D model. Brain surfaces are meshes, cell bodies are spheres and neuronal projections are tubes.

overlapping triangular faces is thus avoided by ensuring that any two consecutive profiles are optimally similar to each other.

The sequence of editions between any pair of strings not only enables the creation of interpolated profiles, but also clearly specifies which vector in one string corresponds to another on the other (the 'traces' between elements of two strings; Wagner and Fischer, 1974). The generation of quadrangular and triangular faces consists in iterating over the sequence of editions and trivially matching the points accordingly. Our approach thus avoids the generation of undesirable faces associated with existing methods (fig. 1c).

3 QUICK MODELING WITH TUBES AND SPHERES

A radically different approach to precise meshing is the generation of simple 3D objects such as tubes and spheres. User-drawn Bezier curves with variable radii spanning throughout a stack of seriated images, and points with defined radius, serve as sketched skeletons to build 3D tubes and spheres.

The sketching of tubes has several advantages. The most obvious are the high speed in manually segmenting the objects of interest, or in perusing the path of an automatic particle tracer. But most remarkably, the linear nature of the tubes enables their comparison and averaging, by using the string of vectors morphing algorithm as a path comparator instead of in meshing. For instance, several sketches can be merged together by first reorienting the tubes and then obtaining average (or consensus) tubes, generating a reference 3D model. Subsequently, given a reference 3D model containing labeled 3D tubes associated with known structures, newly sketched tubular structures can be quickly identified by both visually and algorithmically comparing them with those of the reference model.

The sketching of spheres is most useful as a fast and simple way to represent cells or cell nuclei in 3D. For instance, the modeling of neuronal structures is performed mostly with tubes (the neurites) but also spheres (the cell bodies) which provide a root for the origin of the branched neurite tree.

4 CONCLUSION

Blender as a software package has a lot to offer to the scientific community. Not only because of its fantastic mesh manipulation, rendering and video capabilities, but also by the ease in extending its functionality by python and C programs.

REFERENCES

Cardona, A. (2006). TrakEM2: an ImageJ-based program for morphological data mining and 3D modeling. In *Proceedings the First ImageJ Conference*. Luxembourg.

Jiang, X., Bunke, H., Abegglen, K., and Kandel, A. (2002). Curve Morphing by Weighted Means of Strings. In *Proceedings of the Sixteenth Conference on Pattern Recognition (ICPR 2002)*, volume 4, pages 192–195. IEEE Press.

Wagner, R. and Fischer, M. (1974). The String-to-String Correction Problem. *Journal of the Association of Computer Machinery*, 21(1):168–73.