Development 138, 159-167 (2011) doi:10.1242/dev.051086

The interactive presentation of 3D information obtained from reconstructed datasets and 3D placement of single histological sections with the 3D portable document format

Bouke A. de Boer[1,2,*], Alexandre T. Soufan[1,*], Jaco Hagoort[1], Timothy J. Mohun[3], Maurice J. B. van den Hoff[1], Arie Hasman[2], Frans P. J. M. Voorbraak[2], Antoon F. M. Moorman[1] and Jan M. Ruijter[1,†]

SUMMARY

Interpretation of the results of anatomical and embryological studies relies heavily on proper visualization of complex morphogenetic processes and patterns of gene expression in a three-dimensional (3D) context. However, reconstruction of complete 3D datasets is time consuming and often researchers study only a few sections. To help in understanding the resulting 2D data we developed a program (TRACTS) that places such arbitrary histological sections into a high-resolution 3D model of the developing heart. The program places sections correctly, robustly and as precisely as the best of the fits achieved by five morphology experts. Dissemination of 3D data is severely hampered by the 2D medium of print publication. Many insights gained from studying the 3D object are very hard to convey using 2D images and are consequently lost or cannot be verified independently. It is possible to embed 3D objects into a pdf document, which is a format widely used for the distribution of scientific papers. Using the freeware program Adobe Reader to interact with these 3D objects is reasonably straightforward; creating such objects is not. We have developed a protocol that describes, step by step, how 3D objects can be embedded into a pdf document. Both the use of TRACTS and the inclusion of 3D objects in pdf documents can help in the interpretation of 2D and 3D data, and will thus optimize communication on morphological issues in developmental biology.

KEY WORDS: Three-dimensional reconstruction, 3D pdf, TRACTS, Cardiac development, Section registration

INTRODUCTION

Basic questions in developmental biology concern the mechanisms of morphogenesis and differentiation. The results of sophisticated, quantitative gene expression profiling methods, such as microarrays and next-generation sequencing, have to be verified by visualizing gene expression in the three-dimensional (3D) context of the developing organs and tissues. In cardiac embryology we have demonstrated that modern 3D reconstruction techniques to visualize morphological changes (Soufan et al., 2003) help in the identification of cardiac compartments on the basis of gene expression (Hoogaars et al., 2004; Soufan et al., 2004) and in the understanding of the role of morphogenetic parameters, such as cell size and proliferation rate, in cardiac development (Soufan et al., 2006; van den Berg et al., 2009).

Histological methods are still required to study the 'where and when' of gene expression. To interpret the results of these methods, molecular biologists have to find their way in the ever-changing landscape of the developing embryo, which is far from trivial as wrongly annotated sections do appear in papers (Anderson et al., 2008). Placing individual sections into a 3D reference model would be of great help to the non-morphologist in need of histological confirmation of a gene expression profile (de Boer et al., 2007). Moreover, it can be argued that the current annotation of 'arbitrary' sections is often insufficient to study genetic interactions. In the heart, the wall of a compartment can be composed of several genetically distinct regions (Soufan et al., 2004). Placing sections in a cardiac reference model will help in the exact annotation of such compartments and will thus prevent misunderstandings when such scientific concepts are communicated. Indeed, much of the disagreement in embryology might be caused by such miscommunication, as it is far from easy, even for experts, to form a correct mental image of a dynamically changing 3D structure (Soufan et al., 2004).

Understanding the intricate morphology of developing embryos ideally involves the handling of the 3D structure with ones own hands. 2D images found in the literature can only partially replace the insights gained from interactively working with 3D reconstructions on a computer screen. At most, a supplementary movie clip can be produced in an attempt to convey the insights that the researchers have gained from such 3D analyses. However, the complementation of printed journals with web-based publishing provides the opportunity to make the results of morphological studies available as interactive 3D objects. Special visualization software is no longer required because Adobe Acrobat can be used to place surface-rendered 3D objects into a scientific paper that is distributed as a portable document format (pdf) file (Murienne et al., 2008; Neusser et al., 2009; Ruthensteiner and Hess, 2008) that can be read with the freely available Adobe Reader. We have used

[1]Department of Anatomy, Embryology and Physiology, Heart Failure Research Center, Meibergdreef 15, 1105 AZ Amsterdam, The Netherlands. [2]Department of Clinical Informatics, Academic Medical Center, Meibergdreef 15, 1105 AZ Amsterdam, The Netherlands. [3]Developmental Biology Division, National Institute for Medical Research, The Ridgeway, Mill Hill, London, NW7 1AA, UK.

*These authors contributed equally to this work
†Author for correspondence (j.m.ruijter@amc.uva.nl)

Accepted 20 September 2010

such 3D pdfs as supplementary material with publications on heart development (Moorman et al., 2009; Postma et al., 2009; van den Berg et al., 2009; van Wijk et al., 2009) and pelvic anatomy (Wallner et al., 2008; Wallner et al., 2009).

This paper describes a computer application that we have developed to place arbitrary sections of the developing mouse heart into their proper anatomical context using high-resolution 3D reference models. Furthermore, it describes how a 3D reconstruction can be converted into an interactive 3D pdf file.

MATERIALS AND METHODS

The tracing of the anatomical context of tissue sections (TRACTS) into reference models and the creation of a 3D pdf are illustrated with an example from mouse cardiac embryology. Procedures to test and validate the placing of sections are described.

Program development

Availability

The TRACTS program and its source code, the database of reference model sections (see Appendix S2 in the supplementary material) and the reference models are available on request under a general public license. For the 3D pdf protocol and its scripts, see Appendix S1 and associated scripts file in the supplementary material.

Computer and software

The TRACTS application was developed using Matlab version 2009a (The MathWorks, Natick, MA, USA). Programming and all tests were performed on a system with an Intel P9600 2.66 GHz processor (Intel Corporation, Santa Clara, CA, USA) with 4 GB RAM running MS Windows Vista 64-bit Business as the operating system (Microsoft Corporation, Redmond, WA, USA). Amira 5.2 (Visage Imaging, San Diego, CA, USA) was used to create 3D reconstructions from serial sections. For the 3D pdf protocol we used Adobe Acrobat Pro Extended 9.3, Adobe Reader 9.3 and Adobe Illustrator CS4 (Adobe Systems, San Jose, CA, USA); we refer to these programs as Acrobat, Adobe Reader and Illustrator, respectively. The basic and advanced steps of the 3D pdf protocol were performed on a standard PC running MS Windows XP Pro as the operating system.

3D models

Biological material

Embryonic day (E) 10.0, E11.0, E11.5 and E13.5 mice (FVB) were used to generate 3D models of the heart. These were used to develop and test the performance of the program. The hearts were staged using two mouse development atlases (Kaufman, 1994; Theiler, 1972).

High-resolution reference and test models

The reference and test models were created using high-resolution episcopic microscopy (HREM) (Weninger et al., 2006). In short, this technique involves embedding specimens in a fluorescent resin and imaging the block face before sectioning. The resulting image sets were filtered to reduce out-of-plane information. The heart was segmented manually using Amira (Fig. 1A). Segmentation was performed using the Magic Wand tool, manually placing borders where the pericardial wall contacts the heart and at the pericardial reflection. The two E11.5 imaged embryos resulted in datasets of the whole embryo with voxel sizes of ~3.5×3.5×3 μm; the E10.0 embryos with ~1.8×1.8×2 μm; and the E13.5 embryos with ~4×4×4 μm. We used an episcopic imaging technique [see Geyer et al. for available methods (Geyer et al., 2009)] as it enabled imaging at high resolution combined with an absence of distortion, making it an ideal technique to virtually cross-section the resulting dataset in all possible directions without loss of the original morphology and image resolution.

Paraffin test models

Two specimens (E11.0 and E11.5) were embedded in paraffin. Serial sections on which in situ hybridization (ISH) (Somi et al., 2006) was performed were used to create 3D reconstructions (Soufan et al., 2003). Every other 10 μm section was stained to visualize the myocardium using a probe to *Mlc2a* (*Myl7*) mRNA (E11.0) or *cTnI* (*Tnni3*) mRNA (E11.5). The sections were aligned and segmented using Amira. Segmentation was mainly performed by setting an intensity threshold followed by removal of nonspecific staining. The resulting 3D reconstructions had voxel sizes of 0.68×0.68×20 μm and 0.87×0.87×20 μm, respectively.

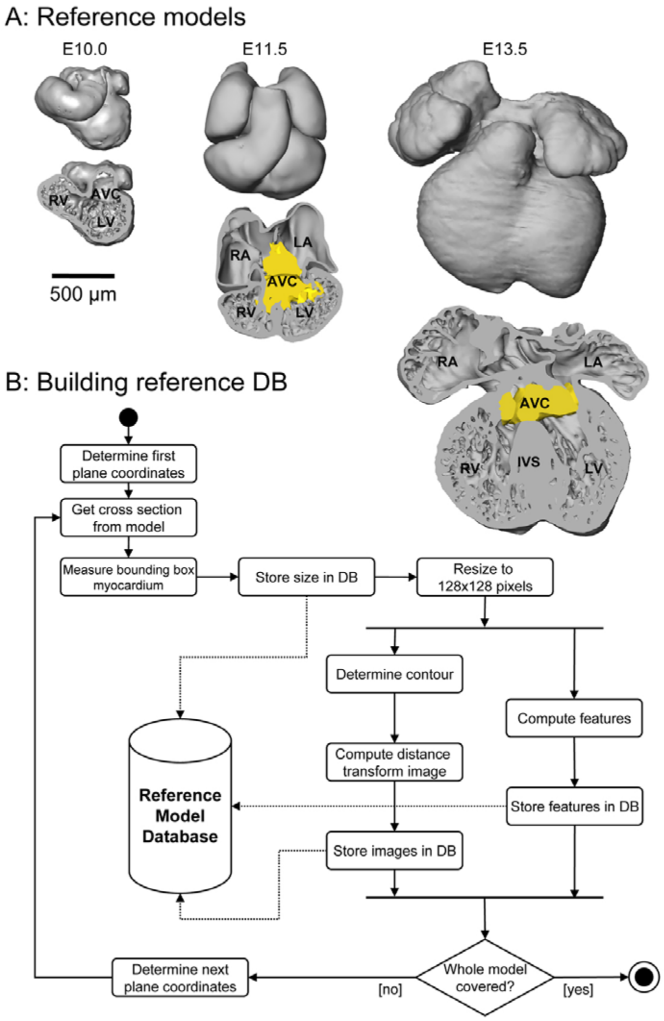


Fig. 1. Reference models and reference database. (**A**) Reference models are based on episcopic images of E10.0, E11.5 and E13.5 mouse hearts embedded in resin. From the image stacks, the myocardium (gray) and the cardiac cushions (yellow) were segmented. The reference models are shown in a frontal and a 'four-chamber' cross-sectional view. LA, left atrium; RA, right atrium; AVC, atrioventricular canal; LV, left ventricle; RV, right ventricle; IVS, interventricular septum. (**B**) Activity diagram listing the steps TRACTS takes when building the database of reference images. The algorithm starts at the geometric center of the first slice without applying rotation and tilting (determine first plane coordinates). An important step is obtaining the contour of the myocardium (determine contour) by detection of all myocardium pixels adjacent to the background pixels. Plane coordinates are determined by iteration of the translation, tilting angle and tilting direction (see Appendix S2 in the supplementary material). The complete set of virtual cross-sections is uniformly distributed in the 3D space of the reference model.

TRACTS performance tests

The performance of TRACTS was tested using (1) virtual cross-sections in arbitrary directions from a second episcopic dataset; (2) samples of paraffin sections taken from the ISH datasets, which is the type of data used for gene expression visualization; and (3) a comparison with the performance of a panel of morphology experts, who manually fitted a sample of the episcopic input set to the reference model.

Episcopic-episcopic performance

The episcopic test model (Fig. 3A) was aligned in 3D (using translation and rotation) with the E11.5 reference model (Fig. 1A) and a systematic random sample of 105 cross-sections was generated. Of these sections, 95 were deemed relevant; the other ten sections contained only a small part of the heart.

Performance on ISH-stained paraffin sections

Because of tissue deformation and the low resolution perpendicular to the sectioning plane, it was necessary to use original sections of the paraffin models as test sections. The reference model was, therefore, aligned to each of the paraffin models (using translation, rotation and isotropic scaling). Samples of 25 and 21 sections were used from the E11.0 and E11.5 paraffin models, respectively.

Expert test

The morphology experts all have several years of experience in research in morphogenesis and heart development. In this test, the experts could use a simple user interface to interactively find the position of an input section in the 3D reference model. A randomly presented sample of 24 sections was used. This set of images was drawn from the set of cross-sections of the episcopic test model that was used for the episcopic-episcopic test of the program. There was no significant difference (*P*=0.110) between the fit errors of TRACTS on the subset of the images used for the expert test and the other images in the episcopic-episcopic test.

Normalized fit error

In all performance tests the reference model of an E11.5 mouse heart was used. Test images were generated as cross-sections of the aligned 3D test models. In this way, the approximate position of each cross-section of the test model is known in the reference model, which enables the computation of a fit error. This fit error takes the distance and the angle between the determined and actual cross-section into account and is normalized to values between 0 and 100 (see Appendix S2 in the supplementary material).

Statistics

Box plots show the results of the performance tests. The red line denotes the median fit error and the blue box shows the interquartile range; outliers (indicated by plus signs) are defined as results more than 1.5 times the interquartile range distance from quartile points. Because of the non-normal distribution of the normalized fit error, nonparametric statistics were used to compare the results of the different performance tests. To this end, the Kruskal-Wallis test (*n* groups) or the Mann-Whitney test (two groups) was used (Conover, 1980).

3D pdf

The full protocol for generating a basic and an advanced 3D pdf is given in the Appendix S1 in the supplementary material. The protocol assumes that the user is already familiar with creating 3D reconstructions from histological sections (Soufan et al., 2003).

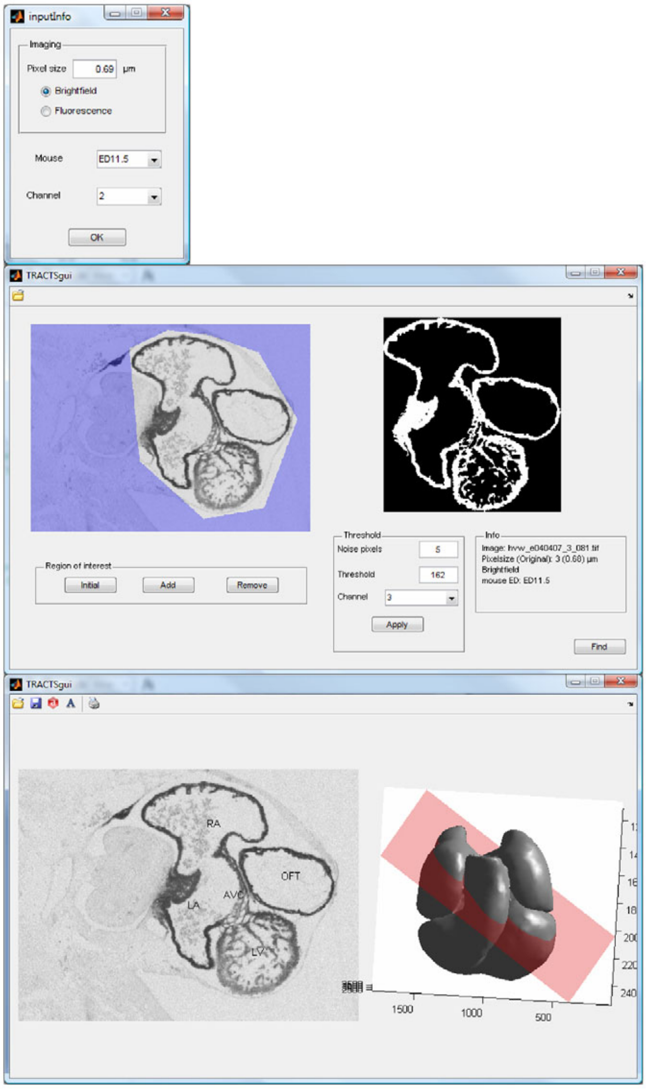
The basic protocol is used to generate a simple 3D pdf of a reconstructed 3D object and is customized for the conversion from Amira to Acrobat. The resulting 3D pdf can be enhanced to generate an advanced 3D pdf that can be used more easily and intuitively. These independent enhancements can be tailored to the document or paper at hand. By adding annotations and preset views, the author can better convey the insights obtained from the study of the 3D object. Also, functionality and robustness can be added by applying scripts to buttons and the document itself. This advanced protocol is independent of the original 3D reconstruction program.

RESULTS

The TRACTS program

Basic principles of TRACTS

The basis of the program to 'trace the anatomical context of tissue sections' (TRACTS) is a straightforward brute force pixel-based procedure, in which each (2D) input section is compared with a large number of virtual (2D) cross-sections of the (3D) reference model. In the input section, the myocardium needs to be separated from background staining using an intensity threshold or manual segmentation. Using the supplied user interface (Fig. 2), the images can be automatically or manually thresholded and small (nonspecifically stained) islands can be removed. A similarity metric based on the contour of cardiac tissue is used to select the most similar cross-section of the reference model. To avoid repetitive computations, a database was created containing pre-computed cross-sections of the reference model (Fig. 1B and see Appendix S2 in the supplementary material). To reduce misplacement and processing time, basic image features such as section size, tissue



inputInfo

Imaging

Pixel size 0.69 µm

● Brightfield

○ Fluorescence

Mouse ED11.5

Channel 2

OK

TRACTSgui

Region of interest

Initial Add Remove

Threshold

Noise pixels 5

Threshold 162

Channel 3

Apply

Info

Image: hvw_e040407_3_081.tif

Pixelsize (Original): 3 (0.68) µm

Brightfield

mouse ED: ED11.5

Find

TRACTSgui

Fig. 2. User interface of TRACTS. In the import dialog (top), the user can select the stage into which the input section has to be fitted. After import of the image the user can define a region of interest in which an (automatic) intensity threshold is employed for segmentation of the myocardium. Next, islands of an adjustable number of pixels can be automatically removed (middle). The myocardium is then mapped into the reference model of the selected age. The result window (bottom) shows the input image with annotated cardiac compartments and the position of the plane in the reference model. The latter view can be interactively viewed from different directions.

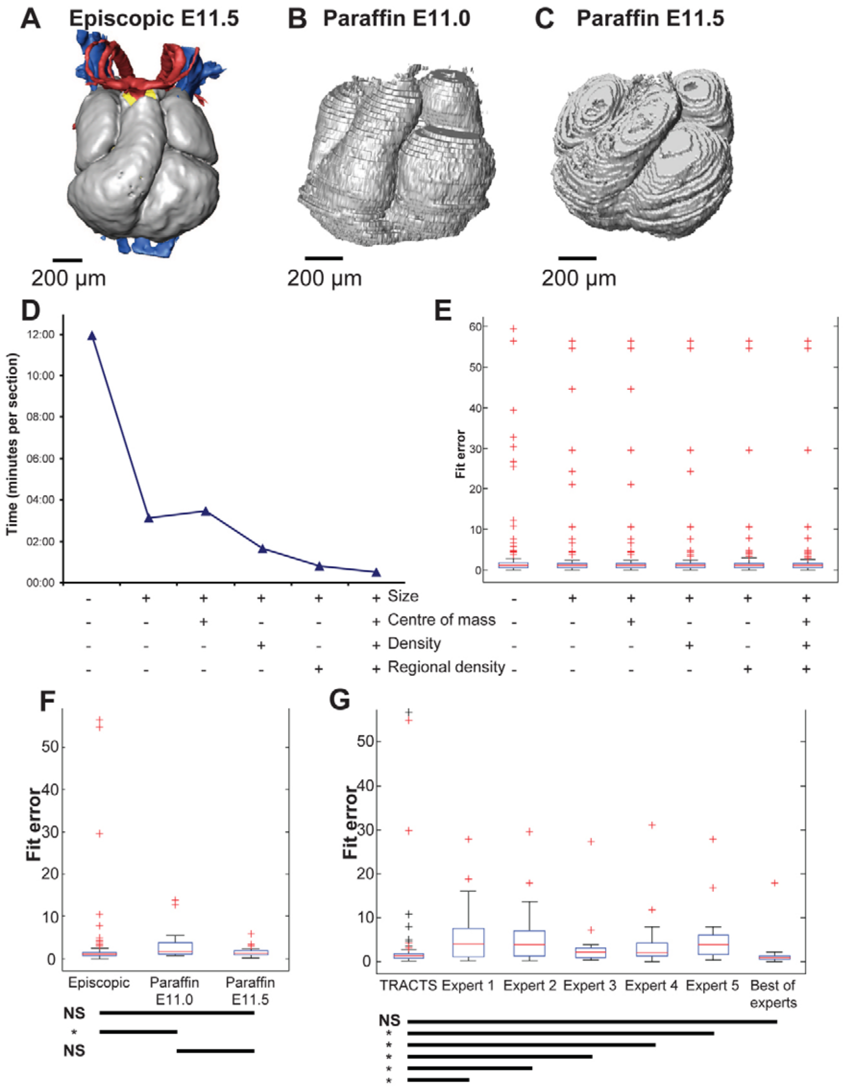


Fig. 3. Validation of TRACTS. (**A**) The episcopic test model of stage E11.5 from which 95 virtual sections were taken and placed by TRACTS. (**B**,**C**) 3D reconstructions based on paraffin sections of E11.0 (B) and E11.5 (C) mouse embryos. Sections of these reconstructions were used to assess the performance of TRACTS on standard histological sections. (**D**) Computation time in the episcopic-episcopic test for different feature settings, when the feature is used (+) or not used (–). (**E**) Box plots showing the normalized fit error of TRACTS, using different feature settings. The red line denotes the median fit error and the blue box shows the interquartile range. Outliers (plus signs) are defined as more than 1.5 times the interquartile width around the interquartile range; the plotted whiskers extend to the most extreme data values that are not an outlier. (**F**) Box plots comparing the performance of TRACTS with the episcopic dataset and with sections of the two paraffin datasets. There was no significant difference between the performance on the E11.5 paraffin and the episcopic dataset, nor between the two paraffin sets, but there was a significant difference between its performance on the E11.0 paraffin and E11.5 episcopic datasets. NS, not significant; E11.5 paraffin-episcopic, *P*=0.191; E11.0 paraffin-episcopic, *P*<0.001; E11.5 paraffin-E11.0 paraffin, *P*=0.069. (**G**) Box plots comparing the performance of TRACTS with the result of each morphology expert and, for each section, the best-fit result of any of the experts ('best of experts'). The black plus signs in the TRACTS column indicate sections that were not included in the sample of 24 sections registered by the experts. The performance of TRACTS was significantly better than each individual expert but there was no significant difference between TRACTS and the 'best of experts'.

density, center of gravity and regional density are used to preselect the cross-sections from the reference model that are to be compared with the input image (de Boer et al., 2007). Details of the similarity metric, the reference model database and the image features are given in Appendix S2 in the supplementary material.

Effectiveness of image features

To test the use of image features in TRACTS, a sample of 95 test images taken from the episcopic test model was fitted (Fig. 3A). The test was performed for six different feature settings of TRACTS: purely brute force; only constraint on size; the combinations of size with center of mass, density or regional density; and all features combined. The normalized fit error was computed for each section in each test (Fig. 3E). Computation time per image is given in Fig. 3D. Performance was significantly improved when only cross-sections that are similar in size were compared. The second largest effect was observed with the regional density feature. Using additional features individually gave only a small further improvement, but each feature did result in a reduction in the time needed to register an input section. The combination of all features did not further reduce the fit error but led to a further reduction of the computation time to only 30 seconds per section.

Some of the sections that were placed at an entirely incorrect position when using only the size constraint were fitted near the correct position when preselection on image features was applied (Fig. 4A). The center of mass feature rejected a section cut through a relatively symmetrical part of the heart; because the dorsal wall of each ventricle is thicker than the ventral wall, the mirrored section that was first found was rejected (Fig. 4A, left; Fig. 4B). The overall tissue density feature served to reject a misplaced section where the high tissue density of the erroneously chosen left ventricle was replaced by the correct right atrium (Fig. 4A, middle). The regional density feature showed its effectiveness by rejecting a solution in which a right atrium was placed on a section through the left ventricular wall (Fig. 4A, right).

Interpretation of the normalized fit error

Fig. 4B shows a 3D view of the placement of an input section (shown in Fig. 4A, left) using only the size constraint (top panel) and using the size and center of mass constraint (bottom panel). The fit errors are 24.3 and 1.4, respectively. The latter fit error is close to the 75th percentile (median, 1.12; interquartile range, 0.64-1.54) of the fit errors in the episcopic-episcopic test. This error corresponds to a 57 µm distance or a 13.6° angle between

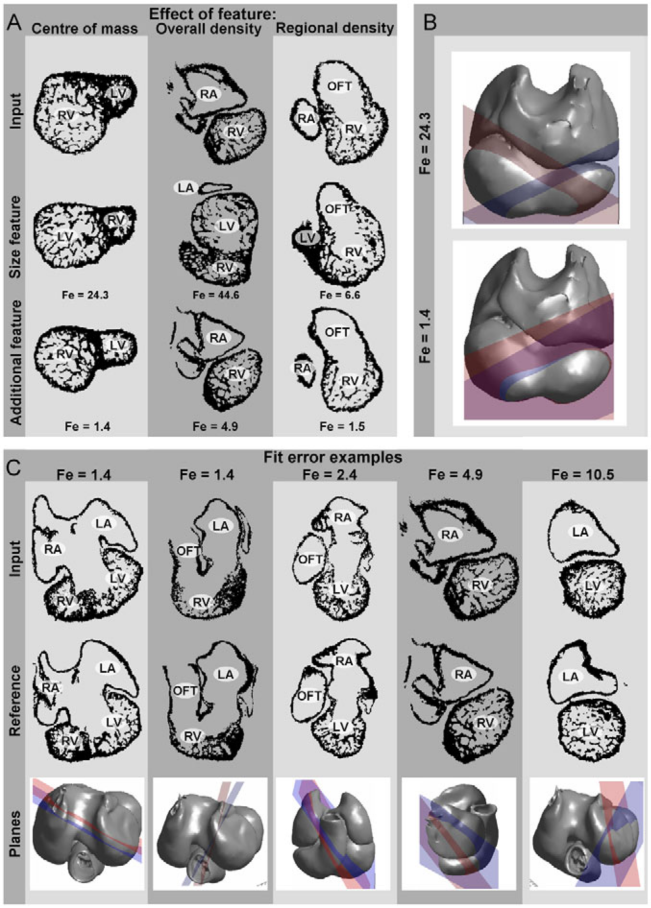


Fig. 4. Effect of image features and interpretation of the fit error. (**A**) Added image features reduce misplacements. At the top, the test sections used as input images for TRACTS are shown. The middle row shows which reference sections were found using only a selection on heart size. The bottom row shows the reference section recovered when an additional feature is used. The normalized fit error (Fe) is given for each of the (mis)placed images. LA, left atrium; RA, left atrium; LV, left ventricle; RV, right ventricle. OFT, outflow tract. (**B**) Comparison of the fit errors with their corresponding planes based on the first lane of A. The blue planes are the input section and the red planes are the positions found by TRACTS, using the size constraint (top) or the center of mass constraint (bottom). (**C**) Examples to help with the interpretation of the normalized fit error. Lanes 1 and 2 show the translation and rotation resulting in a fit error of 1.4, which is close to the 75th percentile; 75% is fitted to positions that deviate less. Lanes 3, 4 and 5 show larger fit errors, although in all cases the correct compartments are recovered.

the observed and correct section plane (Fig. 4B, bottom). In this respect, it is noteworthy that the discrete distance and tilting steps used to prepare the reference image database (see Appendix S2 in the supplementary material) cause a maximum fit error of ~0.37. Fig. 4C shows examples of the locations of input and recovered sections for different fit errors. In column one, a parallel section is shown with a fit error close to the 75th percentile. In the second column, a section fitted with a similar error is shown, in which the error is mainly due to the angle between the planes. Columns 3 through 5 show significantly higher fit errors, although in each case the correct anatomical compartments are found.

Impact of stage differences and sectioning deformations

When studying cardiac development, the expression patterns of specific genes are visualized by in situ hybridization and/or immunohistochemical staining on paraffin sections. Despite the deformations resulting from paraffin embedding, sectioning and mounting, TRACTS was able to correctly place most paraffin sections into a reference model (Fig. 3F). The two paraffin models in this test were from mouse embryos of stage E11.0 (Theiler stage 18) (Fig. 3B) and E11.5 (Theiler stage 19) (Fig. 3C); the latter is the same developmental stage as the episcopic reference model. For both embryos, TRACTS showed a small median fit error (Fig. 3F). Compared with the test using episcopic test images, TRACTS performed satisfactorily when the test sections were taken from a paraffin model that was of the same stage as the reference model (*P*=0.191). Although the younger embryo showed a significantly poorer fit (*P*<0.001), the performance of TRACTS on the two paraffin models differed only slightly (*P*=0.069), which shows that TRACTS can cope with small differences in stage.

Comparison to manual fit by experts

A panel of cardiac embryologists was asked to manually register a subset of the images used in the above episcopic-episcopic test in the 3D reference model. They could place a plane in a 3D surface reconstruction by changing its position, tilting angle and tilting direction. The interface showed the cross-section at the position of this interactive plane.

The box plots (Fig. 3G) show the performance of TRACTS and each individual expert. The 'best of experts' represents the best-fit result per input image from any of the experts. The median fit error of TRACTS was significantly lower than for each of the individual experts (all *P*<0.005), but there was no significant difference between TRACTS and the best of experts (*P*=0.741). TRACTS took 30 seconds per section, compared with several minutes for the experts.

Image space that is difficult to fit with TRACTS

Despite an overall good performance, TRACTS shows high fit errors in a few situations. Obviously, sections showing only a single atrium or ventricle can seldom be fitted to their correct position. Moreover, the left-right symmetry of the heart can lead to cross-sections through both the atrium and the ventricle and no other structures. Such sections can be fitted equally well on the left or right side of the heart. Finally, rotational symmetry along an axis through the atrium and ventricle can lead to sections showing the correct atrium and ventricle lumen without further clues as to the sectioning direction (Fig. 4C, last two lanes). In these cases, the annotation of the anatomical compartments will be correct.

Some of the high fit errors in TRACTS performance could have resulted from biological variation between the reference and test models. Sections fitted in the anatomically right position will then result in a relatively high fit error because the structures are positioned slightly differently in the reference and test models.

Visualization with 3D pdf

To disseminate 3D information, a surface-rendered 3D object can be converted into a 3D pdf using Acrobat and subsequently viewed, and interacted with, using Adobe Reader (see Fig. S3 in the supplementary material). Throughout this text we will refer to the 3D object as 'object' (e.g. a 3D reconstruction of a heart) and to the parts of this object as 'structure(s)' (e.g. myocardium, cushions and lumen), both in Amira and Acrobat. The object should be a 3D surface, stored by Amira as an Amira Surface file (.surf). The object should not be too complicated (i.e. composed of too many polygons) as this will slow the rendering of the object while viewing the pdf.

Basic 3D pdf

The basic protocol describes how to configure Amira and Acrobat and to properly export the 3D object into a pdf via a single command. The captured pdf can be saved as a basic 3D pdf and the embedded 3D object can be handled interactively using the default Adobe Reader user interface.

Preparation of the 3D object

The conversion of the object from Amira to Acrobat is achieved via a Capture of the Viewer window (OpenGL), which is a feature of Acrobat Pro Extended. Unfortunately, direct capture of an object that consists of several structures will result in an unusable object in Acrobat owing to the separately colored inner and outer surfaces of each structure; juxtaposed structures lead to even more surfaces that are wrongly colored, separated or shared in Acrobat. To solve this, a separate surface for each structure of the object has to be created and assigned the same inside and outside color. Because this can be very laborious, we wrote an Amira script that does this via a single command (see the scripts file in the supplementary material).

Conversion to pdf

When the surfaces are thus assigned, creating a 3D pdf from an object displayed in the Amira Viewer window becomes as simple as pressing the Print Screen button. A capture to Acrobat then contains all separate structures of the object colored correctly. The names assigned to the structures are lost by the capture procedure and reappear as numbers. These can be renamed in Acrobat. The resulting pdf (see Fig. S2 in the supplementary material) contains a 3D object that can be opened in the freeware Adobe Reader.

Basic handling

After opening Adobe Reader, the object is shown as a static image. Clicking on the object will activate the interaction with the object and display the 3D toolbar with the interaction tools (see Fig. S1e in the supplementary material). These interaction tools enable, amongst others, rotation and translation of the 3D object. When the 3D toolbar is hidden, right clicking on the object and selecting Tools>Show Toolbar will uncover it. Similarly, right clicking on the object and selecting Show Model Tree will show the tree containing the structures of the object. Structures can be shown or hidden using the check boxes.

Advanced 3D pdf

The second part of the protocol consists of advanced steps, which require some computer and scripting expertise. There are five independent additions to the basic 3D pdf: (1) creating views and annotations of the object; (2) applying layout; (3) adding buttons; (4) adding scripts for interaction and functionality; and (5) preparing and displaying cross-sections. A 3D pdf featuring these additional steps will be referred to as an advanced pdf (Fig. 5 and see Fig. S3 in the supplementary material).

Views and 3D annotations

To circumvent the convoluted default tools of Adobe Reader we decided to create views and 3D annotations with Acrobat. Annotations of the 3D object are included to identify structures and are created in Acrobat as '3D comment'. Views are preset orientations, structure selections and cross-sections of the 3D object that can be stored in the 3D pdf. Views can be used to guide the reader to the position that, according to the maker, best displays the insights gained from the 3D reconstruction. These preset views in no way diminish the ability of the reader to interact with the 3D object and perform their own exploration of its 3D morphology.

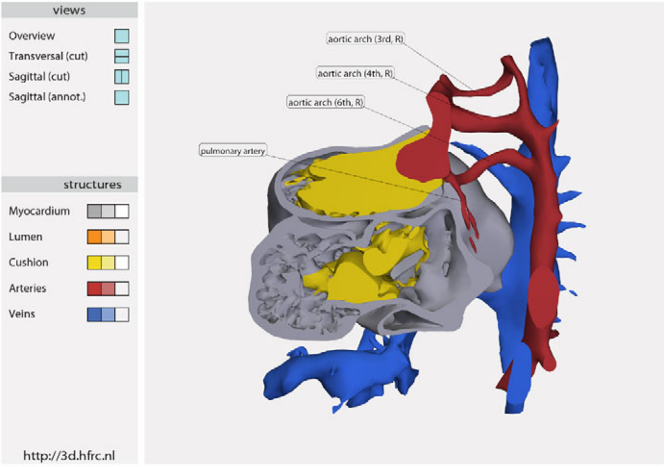


Fig. 5. Static illustration of the advanced 3D pdf. Static illustration of the 3D pdf document of a mouse E11.5 heart created with the 3D pdf protocol described in this paper (see Appendix S1 in the supplementary material). For an interactive version of this 3D pdf, see Fig. S3 in the supplementary material.

Layout

Strictly speaking, layout is not required to disseminate the 3D objects but it will help to convey your insights and to allow intuitive interaction with your document. Layout will vary according to purpose but it generally includes a header, a placeholder for the 3D object, buttons and hyperlinks. The vector graphics editor Illustrator is the ideal application to generate the layout, as line drawings keep the file size to a minimum. Furthermore, Illustrator belongs to the same software suite, avoiding incompatibility problems. After having created a design pdf, the 3D object from the basic pdf created with the basic protocol can be pasted into this design.

Buttons and hyperlinks

The functionality of the 3D pdfs can be enhanced with buttons (see Fig. S1f in the supplementary material). These buttons simplify the default interaction tools for showing views and/or hiding structures. The behavior of buttons can be controlled by scripts. The graphics of the buttons is created in Illustrator and their functionality is added by superimposing transparent buttons in Acrobat; Acrobat also provides default functions that can be assigned to buttons. Hyperlinks can be added to buttons to link the 3D pdf to the accompanying paper and the affiliated institute.

Scripting

Scripting is the most powerful part of the enhancements, for which some programming experience is required. In our published documents (Moorman et al., 2009; Postma et al., 2009; van den Berg et al., 2009; van Wijk et al., 2009) we mainly used scripts to

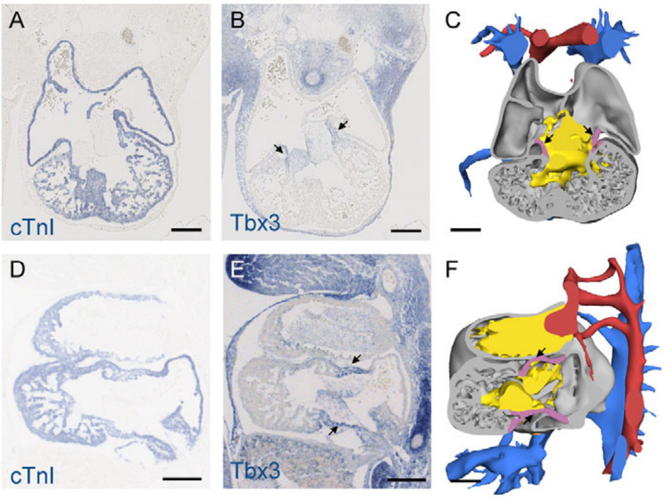


Fig. 6. Biological application. (**A**,**D**) Sections from two different mouse embryos stained for the expression of cTnI. (**B**,**E**) Adjacent sections of A and D, respectively, stained to visualize Tbx3 expression. Arrows indicate Tbx3 expression. The cTnI-positive myocardium in these images was segmented to enable TRACTS to fit these sections into the reference model. (**C**,**F**) The fit locations found by TRACTS are shown in the reference mode; cranial (C) and left lateral (F) views are shown. For an interactive version, see Fig. S3 in the supplementary material. Scale bars: 200 μm.

hide and show structures and to make them transparent. By using functions in scripts, you will centralize the functionality of your document, which makes it easier to change this functionality or apply it to multiple buttons. Furthermore, the functions can be coded to be page independent. This will ensure that the buttons keep working when the page numbers change, which occurs when appending the 3D pdf to a publication or after combining multiple 3D pdf files into one document. The scripts used are detailed in Appendix S1 in the supplementary material.

Cross-sections

Sectioning an object to display a cross-section and to expose the lumen is frequently required in the presentation of results of anatomical and embryological studies. Often, only a specific sectional view reveals the connectivity of compartments or the relation between tissues. Moreover, when the cross-sections placed by TRACTS are displayed in the reference model, the location at which the gene is expressed can be determined (Fig. 6).

Cutting an object and displaying the cross-section is possible within many 3D rendering programs, but, unfortunately, not in Acrobat and Adobe Reader. Acrobat can apply a cross-section, but its interpretation is hampered by the fact that the surface of the cut is not rendered. To show a rendered surface cut, cuts have to be prepared in Amira and subsequently exported as distinct structures. These can be treated in Acrobat in such a way that cutting the object is emulated (see Appendix S1 in the supplementary material).

Biological application

To demonstrate the functionality of the combination of TRACTS and the visualization of its results in a 3D pdf, sections from two different E11.5 mouse hearts were placed into the E11.5 reference model using the user interface (Fig. 2). From both embryonic hearts, two serial sections were stained to visualize cTnI-positive (Fig. 6A,D) and Tbx3-positive (Fig. 6B,E) myocardium. Tbx3 is expressed in the precursors of the cardiac conduction system (Hoogaars et al., 2004). Without the spatial context, the compartments present in the stained sections are difficult to identify. To resolve this, the cTnI-stained sections were loaded into TRACTS and fitted into the E11.5 morphological reference model (Fig. 1A). The cTnI-stained sections were used for this purpose because the heart can be easily segmented from these sections using the user interface (Fig. 2). The coordinates of the cutting plane were exported to the 3D reconstruction program and the cross-section was visualized in the reference model. From this, an advanced 3D pdf (see Fig. S3 in the supplementary material) was created using all steps of the 3D pdf protocol. The results show that one heart was sectioned in a sagittal direction (Fig. 6C) whereas the other was transversally sectioned (Fig. 6F). From these cutting planes it becomes immediately clear that in the embryonic mouse heart, Tbx3 is expressed in the myocardium of the atrioventricular canal as well as in the cushions of the atrioventricular canal and the outflow tract.

DISCUSSION

TRACTS

The fit of sections by TRACTS results in the recovery of the correct position to within 62 μm or 14.3° for 75% of the input sections. In the process of finding this spatial position, TRACTS does not apply any deformation to the image other than anisotropic scaling and rotation. The accuracy of TRACTS is more than sufficient to help the user to identify the compartments and their boundaries in the input section. It will thus point to those compartments that warrant more detailed study to verify colocalization of expressed genes.

Moreover, with paraffin-embedded material, sections from an age-matched embryo are placed with similar precision. Because mouse embryos from the same pregnancy can differ by up to 0.5 days in developmental stage, it is reassuring that the slightly younger paraffin model showed only a small increase in the median fit error and that in this test most recovered reference model sections showed the correct annotation of cardiac compartments. These results show that TRACTS is a robust tool to trace the correct anatomical position of single tissue sections. The comparison of the performance of TRACTS with that of an expert panel showed that TRACTS outperformed each individual expert and performed similarly to the best results of these experts.

However, when comparing expression patterns one should be aware that every embryo is unique, as is every section obtained from it. Furthermore, expression patterns will vary owing to methodological variation, such as in the quality of the antibody, fixation of the specimen and exposure time to substrates.

Because TRACTS relies on the normal morphology of the reference models, fitting sections of deformed, mutant hearts might lead to unpredictable errors. However, when high-throughput mutagenesis studies (Rosenthal et al., 2004) are followed up by gene expression analysis in a mutant line with reproducible aberrant morphology, it would not be difficult to implement a mutant reference model in TRACTS. Comparison between morphology and gene expression patterns in wild-type and mutant hearts will then greatly benefit from the ability to place those reference models side-by-side in a 3D pdf document.

TRACTS can be expanded with the functionality to fit heart sections of ages other than the current series of heart models. Given the simplicity of the currently implemented image features, it is expected that they will be reasonably robust. However, the substantial change in shape that occurs during embryonic development or with malformation might require the implementation of other image features to keep computation times down.

The use of image features has a twofold purpose: (1) to prevent misplacements and (2) to reduce computation time. The use of features resulted in a reduction, from 16% to 7%, in the number of images with a fit error exceeding 5. The computation time dropped over 20-fold to only 30 seconds per section, which is adequate for the intended use of TRACTS and is fast compared with the minutes per section that experts needed to place sections. TRACTS will mainly be used by researchers who use morphological techniques to verify the spatial position of individual sections. For their benefit, the program with annotated reference models is available on request. It should be noted that the reported computation time does not include the segmentation of the input image to extract the contours of the heart. The fitting by TRACTS requires segmented heart tissue. Therefore, sections in which a gene of interest is visualized need to be either alternating or double stained with a cardiac-specific marker, otherwise the heart tissue needs to be segmented manually. This segmentation or thresholding of the myocardium is crucial for the performance of TRACTS, as it uses the contours of the segmented heart to select the best match from the model cross-sections that are selected on the basis of image features. The automated segmentation of the heart will add only seconds to the processing time using the provided user interface (Fig. 2).

Application of the current version of TRACTS to high-throughput projects would not be efficient. When large numbers of stained serial sections are available, fitting of individual sections should take advantage of the additional information, i.e. the known parallelism and distance between the sections. When complete 3D reconstructions are made, a 3D-on-3D registration technique is more appropriate because this can make use of the 3D continuity of the input set.

Besides supplying users with information about the orientation of their input sections in the reference model, TRACTS provides some extra benefits. Because the reference model is anatomically annotated, it is possible to map this annotation onto the input section after it has been placed into the reference model (Fig. 2). This facilitates the annotation of arbitrary tissue sections, avoids erroneous annotations that can be encountered even in anatomical journals, and improves scientific communication. Placing specifically stained sections into a common reference model also makes it possible to collect the visualized gene expression patterns within their anatomical context. This spatial insight can help to generate hypotheses on the genetic regulation of development.

Some of the existing gene expression atlas projects, such as EMAGE (Baldock et al., 2003; Christiansen et al., 2006), already add spatial information (e.g. anatomical context) to their gene expression database (for a review, see de Boer et al., 2009). There are two major differences between the approaches of EMAGE and TRACTS. First, EMAGE is embryo wide, whereas TRACTS is currently focused on the heart. This choice was made to overcome the morphological variations between embryos that hamper the placement of sections covering the whole embryo. Second, in EMAGE the spatial position of a section has to be found manually with MAPaint software (http://genex.hgu.mrc.ac.uk/Software/paint/), whereas TRACTS is aimed to place sections automatically.

3D pdf

Protocols to enable the incorporation of 3D objects in scientific papers have been published in the fields of astrophysics (Barnes and Fluke, 2008), molecular/protein structure (Kumar et al., 2008) and biology (Ruthensteiner and Hess, 2008). However, the protocol we present here is the first 3D pdf protocol specifically tailored to provide the novice biologist with a simple basic protocol and optional advanced steps. Similar to the existing protocols, in which PyMol (Kumar et al., 2008) or s2PLOT (Barnes and Fluke, 2008) are used as 3D visualization software, this protocol was tailor-made for use with the software packages Amira and Acrobat. Irrespective of the visualization software employed, the current protocol describes advanced steps that 3D pdf creators can use to enhance their 3D document, after the capture of the 3D object into the document. Any 3D visualization program using OpenGL should be suitable for the capture of a 3D object by Acrobat, but will require customization of the basic protocol. The protocol described in this paper can then serve to pinpoint the critical settings. Importing 3D file formats into Acrobat (e.g. .obj, .vrml and .3ds) is also possible, but will require a custom protocol to create, export and import such 3D objects, which can become very laborious (Ruthensteiner and Hess, 2008).

A basic 3D pdf, enhanced with views, annotations and cross-sections, can easily be embedded into a paper published in pdf format. However, when the 3D pdf includes layout, buttons and scripts, it is not possible to embed it into another page without interfering with the layout of the publisher. Nevertheless, such an advanced pdf can easily be added as extra pages to the publication pdf or as supplementary pdf files (as with Fig. S3 in the supplementary material). The latter option enables the reader/presenter to show the 3D pdf in full-screen mode, which is also very useful for scientific talks and education.

Although the protocol itself has a steep, but short, learning curve, implementing it is not difficult. Production time can be cut even further using templates, whereby the existing object in the original document can be replaced by another and only minor modifications are required to make the document operational.

Conclusions

The TRACTS program was shown to perform similarly to the best of five morphology experts and to be relatively robust for differences in developmental stage. In combination with the embedding of 3D objects in Adobe pdf, this will improve the annotation of sections and the communication of experimental results. The 3D pdf is an ideal means to cope with the growing demand to include complex 3D interactive information into scientific papers. Enriching documents with 3D objects is ideal in the sense that it is universal; every reader can open and directly interact with the 3D content. Furthermore, it enables researchers to better convey their observations and conclusions, allowing independent verification and avoiding miscommunication. As online publishing becomes more and more common, embedding 3D objects into online publications is the next logical step.

Acknowledgements

We thank Mathilda Mommersteeg, Wim Aanhanen, Vincent Christoffels and Aleksander Sizarov for their participation in the expert panel and for helpful analyses; Chris Wallner for help with the HREM models; and Henk van Weerd and Kees Jan Boogerd for supplying the ISH-stained paraffin reconstructions and sections. B.A.d.B., A.T.S., T.J.M., A.F.M.M. and J.M.R. were supported by the EU Seventh Framework Program Project CHeartED (HEALTH-F2-2008-223040). T.J.M. was supported by the MRC (U117562103). Deposited in PMC for release after 6 months.

Competing interests statement

The authors declare no competing financial interests.

Supplementary material

Supplementary material for this article is available at http://dev.biologists.org/lookup/suppl/doi:10.1242/dev.051086/-/DC1

References

Anderson, R. H., Boyett, M. R., Dobrynski, H., Yanni, J., Christoffels, V. M. and Moorman, A. F. M. (2008). Letter by Anderson et al regarding article, abnormal conduction and morphology in the atrioventricular node of mice with atrioventricular canal-targeted deletion of Alk3/Bmpr1a receptor. *Circulation* **118**, e105.

Baldock, R. A., Bard, J. B. L., Burger, A., Burton, N., Christiansen, J., Feng, G., Hill, B., Houghton, D., Kaufman, M. H., Rao, J. et al. (2003). EMAP and EMAGE: a framework for understanding spatially organized data. *Neuroinformatics* **1**, 309-325.

Barnes, D. G. and Fluke, C. J. (2008). Incorporating interactive three-dimensional graphics in astronomy research papers. *New Astronomy* **13**, 599-605.

Christiansen, J. H., Yang, Y., Venkataraman, S., Richardson, L., Stevenson, P., Burton, N., Baldock, R. A. and Davidson, D. R. (2006). EMAGE: a spatial database of gene expression patterns during mouse embryo development. *Nucleic Acids Res.* **34**, D637-D641.

Conover, W. J. (1980). *Practical Nonparametric Statistics*. New York: J. Wiley.

de Boer, B. A., Ruijter, J. M. and Voorbaak, F. P. J. M. (2007). Towards the automatic registration of histological sections into a 3D reference model. In *BNAIC '07. Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence, Utrecht: 5-6 November 2007* (ed. M. M. Dastani and E. de Jong), pp. 41-48. Utrecht, The Netherlands: BNVKI.

de Boer, B. A., Ruijter, J. M., Voorbraak, F. P. and Moorman, A. F. (2009). More than a decade of developmental gene expression atlases: where are we now? *Nucleic Acids Res.* **37**, 7349-7359.

Geyer, S. H., Mohun, T. J. and Weninger, W. J. (2009). Visualizing vertebrate embryos with episcopic 3D imaging techniques. *Scientific World Journal* **9**, 1423-1437.

Hoogaars, W. M. H., Tessari, A., Moorman, A. F. M., de Boer, P. A. J., Hagoort, J., Soufan, A. T., Campione, M. and Christoffels, V. M. (2004). The transcriptional repressor Tbx3 delineates the developing central conduction system of the heart. *Cardiovasc. Res.* **62**, 489-499.

Kaufman, M. H. (1994). *The Atlas of Mouse Development*. London: Academic Press.

Kumar, P., Ziegler, A., Ziegler, J., Uchanska-Ziegler, B. and Ziegler, A. (2008). Grasping molecular structures through publication-integrated 3D models. *Trends Biochem. Sci.* **33**, 408-412.

Moorman, A. F. M., van den Berg, G., Anderson, R. H. and Christoffels, V. M. (2009). Early cardiac growth and the ballooning model of cardiac chamber formation. In *Heart Development and Regeneration* (ed. R. P. Harvey and N. Rosenthal), pp. 219-236. London, UK: Elsevier.

Murienne, J., Ziegler, A. and Ruthensteiner, B. (2008). A 3D revolution in communicating science. *Nature* **453**, 450.

Neusser, T. P., Hess, M. and Schrodl, M. (2009). Tiny but complex-interactive 3D visualization of the interstitial acochlidian gastropod Pseudunela cornuta (Challis, 1970). *Front. Zool.* **6**, 20.

Postma, A. V., Dekker, L. R., Soufan, A. T. and Moorman, A. F. (2009). Developmental and genetic aspects of atrial fibrillation. *Trends Cardiovasc. Med.* **19**, 123-130.

Rosenthal, J., Mangal, V., Walker, D., Bennett, M., Mohun, T. J. and Lo, C. W. (2004). Rapid high resolution three dimensional reconstruction of embryos with episcopic fluorescence image capture. *Birth Defects Res. C. Embryo. Today* **72**, 213-223.

Ruthensteiner, B. and Hess, M. (2008). Embedding 3D models of biological specimens in PDF publications. *Microsc. Res. Tech.* **71**, 778-786.

Somi, S., Klein, A. T. J., Houweling, A. C., Ruijter, J. M., Buffing, A. A. M., Moorman, A. F. M. and van den Hoff, M. J. B. (2006). Atrial and ventricular myosin heavy-chain expression in the developing chicken heart: strengths and limitations of non-radioactive in situ hybridization. *J. Histochem. Cytochem.* **54**, 649-664.

Soufan, A. T., Ruijter, J. M., van den Hoff, M. J. B., de Boer, P. A. J., Hagoort, J. and Moorman, A. F. M. (2003). Three-dimensional reconstruction of gene expression patterns during cardiac development. *Physiol. Genomics* **13**, 187-195.

Soufan, A. T., van den Hoff, M. J. B., Ruijter, J. M., de Boer, P. A. J., Hagoort, J., Webb, S., Anderson, R. H. and Moorman, A. F. M. (2004). Reconstruction of the patterns of gene expression in the developing mouse heart reveals an architectural arrangement that facilitates the understanding of atrial malformations and arrhythmias. *Circ. Res.* **95**, 1207-1215.

Soufan, A. T., van den Berg, G., Ruijter, J. M., de Boer, P. A. J., van den Hoff, M. J. B. and Moorman, A. F. M. (2006). Regionalized sequence of myocardial cell growth and proliferation characterizes early chamber formation. *Circ. Res.* **99**, 545-552.

Theiler, K. (1972). *The House Mouse, Development and Normal Stages from Fertilization to 4 Weeks of Age*. Berlin: Springer-Verlag.

van den Berg, G., Abu-Issa, R., de Boer, B. A., Hutson, M. R., de Boer, P. A., Soufan, A. T., Ruijter, J. M., Kirby, M. L., van den Hoff, M. J. and Moorman, A. F. (2009). A caudal proliferating growth center contributes to both poles of the forming heart tube. *Circ. Res.* **104**, 179-188.

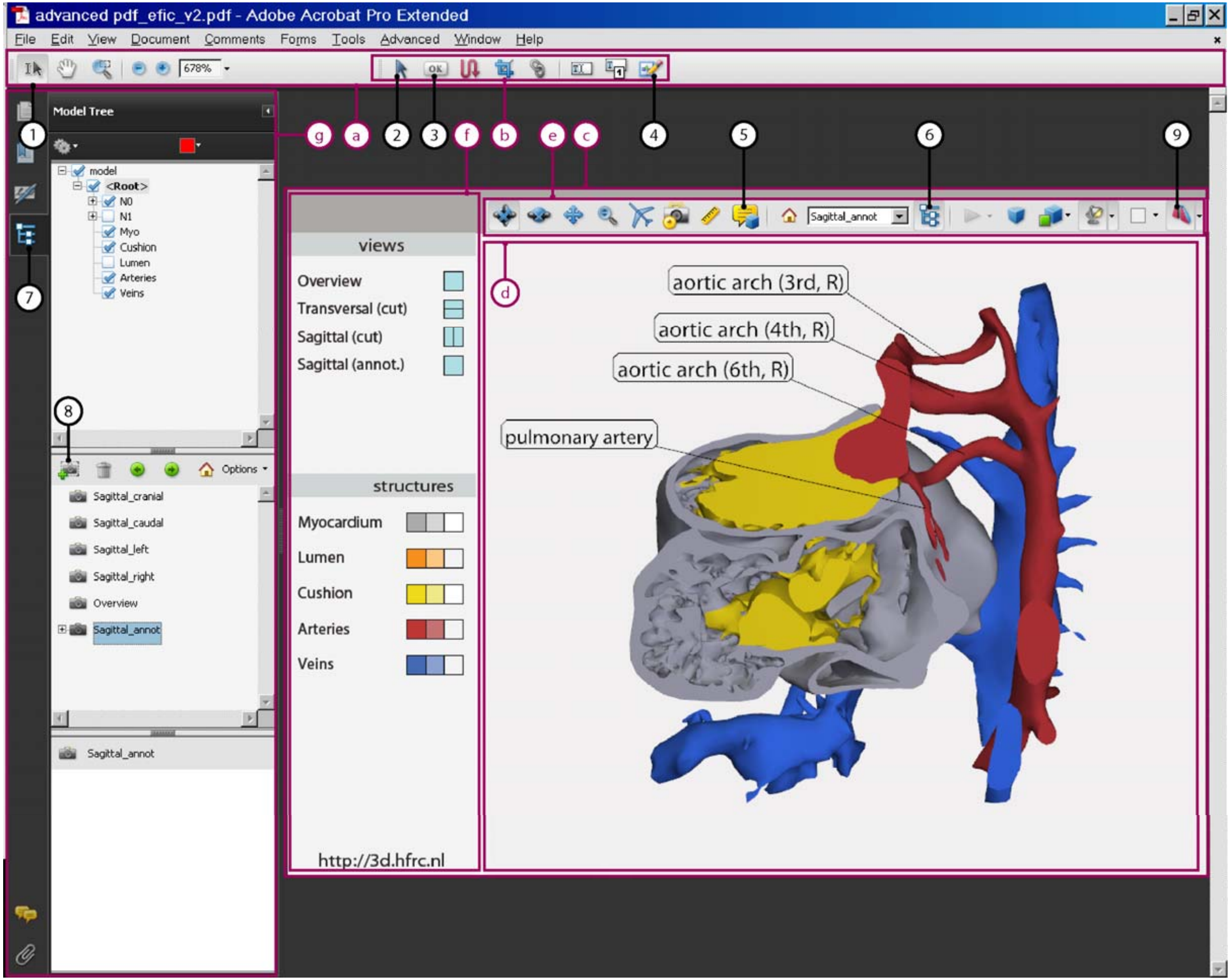
van Wijk, B., van den Berg, G., Abu-Issa, R., Barnett, P., van der Velden, S., Schmidt, M., Ruijter, J. M., Kirby, M. L., Moorman, A. F. and van den Hoff, M. J. (2009). Epicardium and myocardium separate from a common precursor pool by crosstalk between bone morphogenetic protein- and fibroblast growth factor-signaling pathways. *Circ. Res.* **105**, 431-441.

Wallner, C., van Wissen, J., Maas, C. P., Dabhoiwala, N. F., deRuiter, M. C. and Lamers, W. H. (2008). The contribution of the levator ani nerve and the pudendal nerve to the innervation of the levator ani muscles; a study in human fetuses. *Eur. Urol.* **54**, 1136-1142.

Wallner, C., Dabhoiwala, N. F., deRuiter, M. C. and Lamers, W. H. (2009). The anatomical components of urinary continence. *Eur. Urol.* **55**, 932-943.

Weninger, W. J., Geyer, S. H., Mohun, T. J., Rasskin-Gutman, D., Matsui, T., Ribeiro, I., Costa, L. F., Izpisua-Belmonte, J. C. and Muller, G. B. (2006). High-resolution episcopic microscopy: a rapid technique for high detailed 3D analysis of gene activity in the context of tissue architecture and morphology. *Anat. Embryol. (Berl.)* **211**, 213-221.

Figure S1


advanced pdf_efic_v2.pdf - Adobe Acrobat Pro Extended
File Edit View Document Comments Forms Tools Advanced Window Help
678%
Model Tree
model
<Root>
N0
N1
Myo
Cushion
Lumen
Arteries
Veins
Options
Sagittal_cranial
Sagittal_caudal
Sagittal_left
Sagittal_right
Overview
Sagittal_annot
Sagittal_annot
Sagittal_annot
views
Overview
Transversal (cut)
Sagittal (cut)
Sagittal (annot.)
structures
Myocardium
Lumen
Cushion
Arteries
Veins
http://3d.hfrc.nl
aortic arch (3rd, R)
aortic arch (4th, R)
aortic arch (6th, R)
pulmonary artery
1
2
3
4
5
6
7
8
9
a
b
c
d
e
f
g

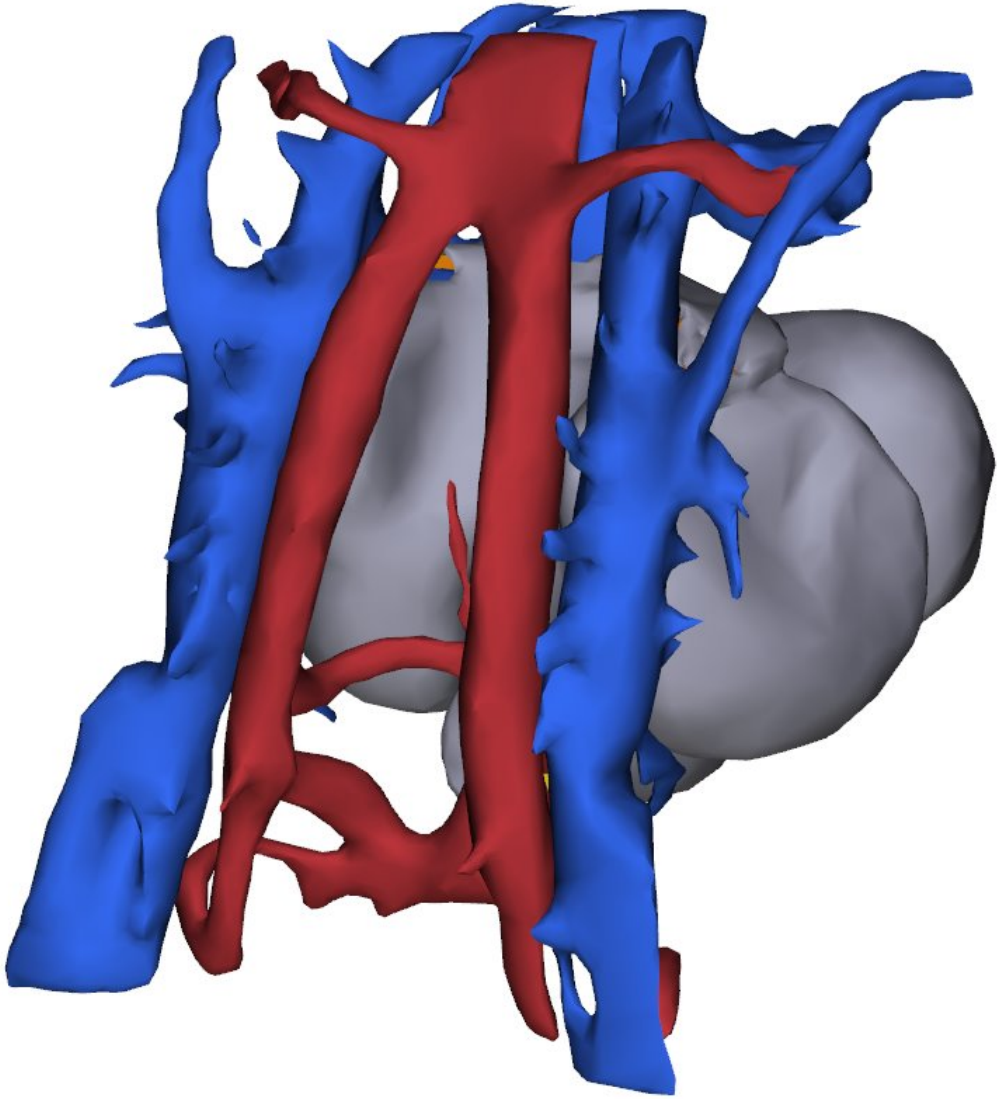


Figure S3; E11.5 Mouse heart

views

Overview □

Transversal (cut) ⊟

Sagittal (cut) ◫

Sagittal (annot.) □

structures

Myocardium ⬛ ⬛ ⬜

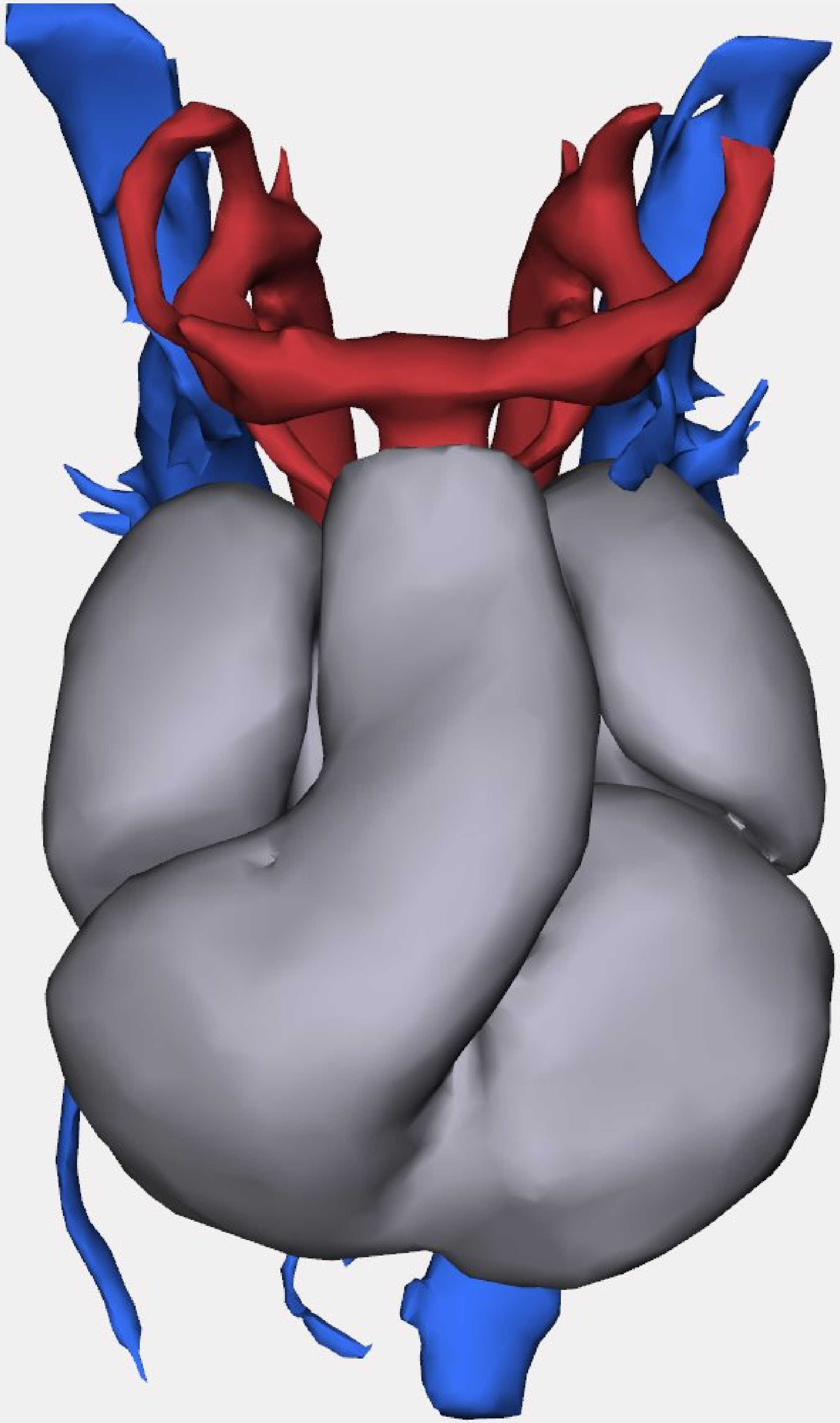
Lumen 🟧 🟧 ⬜

Cushion 🟨 🟨 ⬜

Arteries 🟥 🟥 ⬜

Veins 🟦 🟦 ⬜

http://3d.hfrc.nl



Supplementary 3D-pdf protocol
Version 10-09

Publishing three-dimensional structural information through Adobe's Portable Document Format (pdf); A Protocol for Biologists

METHOD

The programs we use in this protocol are Amira 5.2, Adobe Acrobat Pro Extended 9.3, Adobe Reader 9.3 and Adobe Illustrator CS4. From here on we will refer to these programs as Amira, Acrobat, Adobe Reader and Illustrator, respectively. Furthermore, we will refer to the 3D object as **object** (e.g. a 3D reconstruction of a heart) and the parts this object is composed of as **structure(s)** (e.g. myocardium and lumen), both in Amira and Acrobat. The protocol assumes that the user is already familiar with creating 3D reconstructions. Although this protocol starts with a 3D reconstruction created in Amira, it can be adapted for other 3D visualisation programs.

The protocol consists of two parts. Firstly, a *basic protocol* describes how to embed a 3D surface reconstruction from Amira into a pdf. It describes how to configure Amira and Acrobat and how to properly export the 3D surfaces from Amira into a pdf. This pdf can then be saved as a basic 3D-pdf (Fig. S2) and the embedded 3D surface (we will refer to it as object) can be handled interactively using the Adobe Reader user interface (Fig. S1).

The second part of the protocol describes *advanced steps* requiring a little more computer and scripting expertise. This part describes five (independent) additions; 1) creating *views and annotations* of the object, 2) applying *layout* to the design, 3) adding *buttons, hyperlinks* and 4) *scripts* for interaction and functionality; and 5) preparing and displaying *cross sections.* A pdf containing these additional steps will be referred to as an advanced pdf (Fig. S3).

Finally, we would like to note that possible updates to the protocols and scripts described herein will be made available at the download page of http://3d.hfrc.nl.

Please refer to the main publication when a protocol or script described herein has been applied.

BASIC PROTOCOL

Configure Acrobat

- Start Acrobat and open the *3D Capture* window by selecting *File* > *Create PDF* > *From 3D Capture* from the File menu.
- Start Amira and load and display a simple surface. Select the Amira Viewer window and press the *Print Screen* button on the keyboard, whereupon Acrobat will display a popup asking if the user wants to enable capturing from Amira.
- Click ‘yes’ and restart both Amira and Acrobat.
- Reload and display the simple surface again in Amira, select the Amira Viewer window and initiate a capture once more by pressing *Print Screen*.
- In Acrobat; in the 3D Capture Settings (which only appears the first time);
    - o Verify that *Enable 3D Capture for this application* is checked.
    - o Under the OpenGL tab, set the *Capture units* to those used in Amira. (Unfortunately, it seems that micrometers are not properly shown as units in Acrobat, but an update might solve this).

- In the Acrobat 3D Conversion dialogue;
    - Under the *General* tab;
        - Set *Default Background Color* to white (or another suitable flavour).
        - Set *Default Lighting to* ‘CAD Optimized lights’.
        - Set *Default Rendering Style* to ‘Solid’.
        - Set *Default Animation Style* to ‘None’.
        - Uncheck *Add default views.*
    - Under the *Document* tab,
        - Set the *Page Setup* to A4, Landscape
    - Under the *Optimize* tab,
        - Set the *3D format in PDF* to ‘U3D ECMA 3’.
        - Set *U3D: Mesh Quality* to 50%
    - Save these *3D Conversion Settings* as default presets for capture from Amira by clicking the ‘+’ button. (e.g. Amira_capture)

Load surface file

Your object should be a 3D surface, stored by Amira as an Amira Surface file (.surf). As a rule of thumb; keep the complexity of the object below 100,000 triangles. If you are not familiar with surface simplification, consult the Amira documentation.

- Load your object into Amira,
- Connect a *SurfaceView* to the object.

Prepare the surfaceviewers

The object properties of Amira are problematic, as each structure is dual layered (i.e. has an inner and outer surface, each with its own color) and intersections between structures are also exported as separate surfaces. This has been solved using a custom script which enables proper export of the object. For each structure of the object this script creates a separate surface, makes the inside color equal to the outside colour, connects a separate *SurfaceViewer* with correct render settings and renames the surfaces and viewers with correct structure names and numbers.

- Verify that the *SurfaceView* connected to the object is named SurfaceView.
- Load (or drag and drop) the script s*urf2pdf.hx* (see supplementary scripts) into Amira.
- Hide -or remove- the structures you do not wish to export.

3D capture

- Ensure that Acrobat is running.
- Ensure that the Amira Viewer is the active window (i.e. selected).
- Press the *Print Screen* button on the keyboard.
- In Acrobat; accept the conversion dialog using the preset for Amira (e.g. Amira_capture), which generates a pdf of your 3D object.

Rename structures

The structures need to be renamed as the names assigned to the structures in Amira are lost by the capture procedure and reappear in Acrobat as numbers.

- Right click on the object in Acrobat and select *Edit in 3D Reviewer*, which will start the program ‘Adobe 3D Reviewer’ with the object loaded.
- On the left panel, expand the object tree till the structure names are shown. These will be named M0, M1..Mn and are under *Models*.

- Rename each structure by right clicking on its tree entry and selecting *Rename*. By placing the Amira window adjacent to the Adobe 3D reviewer one can easily refer to the original names.
- Save the changes by selecting *Save back in PDF* from the File menu of the Adobe 3D Reviewer.
- Close Adobe 3D Reviewer and click ‘cancel’ in its *save changes* dialog.

Save the 3D pdf

- In Acrobat; save the resulting file (by default called LastCapture_<timestamp>.pdf) under an appropriate new name.

The pdf contains a 3D object and can be opened and viewed in Adobe Reader, the freeware ‘read-only’ program for Adobe’s portable document format (pdf). The 3D object can now easily be shared with others, but make sure that their version of Adobe Reader is up-to-date.

Interaction tools

For the resulting 3D-pdf, interaction with the object can be done with the default tools implemented in Acrobat and Adobe Reader. By default the object is shown as a static image. Clicking on the object will activate the interaction with the object and also show the 3D toolbar with the interaction tools (Fig. S1, panel e). These interaction tools enable rotation and resizing of the 3D object. If the 3D toolbar is hidden, right-clicking on the object and selecting *Tools>Show Toolbar* will uncover it. Furthermore, right-clicking on the object and selecting *Show Model Tree* or clicking the button (Fig. S1, button 6) will show the model tree containing the structures of the object (Fig. S1, g). The tree can be expanded by clicking on the plus symbol and structures therein can be made (in)visible by toggling the checkboxes.

ADVANCED STEPS

The above basic protocol can be used to generate a simple 3D-pdf of a reconstructed 3D object. To obtain an advanced 3D-pdf (Fig. S3) enhancements need to be applied to this basic 3D-pdf. These enhancements are independent of each other, so a single (or multiple) modification -as described below- can be applied, whatever suits the document or paper at hand. The main purpose of these enhancements is to present the reader a polished document which is easily and intuitively used. By adding annotations and prepared views, the disseminator can better convey the insights obtained from the object. Also, functionality and robustness can be added by applying scripts to buttons and the document itself.

Views and annotations

Adding views and annotations comes highly recommended, as these are the simplest steps of the advanced additions and allow the disseminator to better convey the insights obtained from the object.

- Views can be added and managed in the model tree (Fig. S1, panel g). The model tree can be opened in multiple ways; right clicking on the activated object and selecting the *Show Model Tree* or pressing on the *Show Model Tree* button (Fig. S1, button 6)
- Position the object interactively into a required view
- Once a suitable view has been set up, pressing the *Create View* button (Fig. S1, button 8) will add a view to the Model Tree. The View panel also contains buttons to delete views, or to set a default view (i.e. the view that is set upon opening the document).

Adding annotations to the object is possible -as of Acrobat version 9.1-. This annotation of structures should not be confused with adding annotations to the document (e.g. *sticky note, callout tool*, *Text Box tool)*. Annotating the 3D object is just to identify structures.

- Add Annotations using the *Add 3D Comment* option of the 3D toolbar. By default, the toolbar is shown at the top of the object, or you can select these tools by right clicking on the object and selecting *tools*
- You can set the colour of the comments in (Edit > Preferences > Measuring (3D) > 3D Measuring Line Colour).

Layout

To apply or create the content for your layout, the vector graphics editor Illustrator is the ideal application to use, as vector graphics keep the file size to a minimum. Furthermore, Illustrator belongs to the same software suite which avoids compatibility problems. The first thing to consider is what the appropriate size and ratio for you document should be. This is dependent of your end product, e.g. A4 when you want to include the object in a standard 'paper' document or a 4:3 ratio if the document is meant to snugly fit a digital presentation, as the majority of displays and beamers use a 4:3 ratio (note that 16:9 is steadily advancing). We recommend to include the following content; a header, a placeholder for the 3D object and buttons to either interact with the 3D object or to serve as hyperlinks to the accompanying paper and affiliated institute. We recommend to design the buttons yourself, as opposed of creating them in Acrobat because this keeps design separated from function. The functional part of hyperlinks and buttons is discussed in the next chapter.

Start by creating a new file in Illustrator: *File > New* and set the page to its appropriate size and aspect ratio.

- Create the minimum graphics;
  - A clear header with a description of the object(s).
  - A simple rectangle which will function as a placeholder for the 3D object.
  - Graphics for buttons and hyperlinks to the original paper and your department's or group's website.
- Once finished with the design, save the file as a design-pdf.
  - Set the *Adobe PDF Preset* in the save dialogue to *Smallest File Size,* which will decrease the file size and remove the layer structure of Illustrator which is currently not fully compatible with 3D objects.

After having created this design-pdf, the 3D object from the basic-pdf created with the *basic protocol* needs to be pasted into this design;

- Open both the basic 3D-pdf and the design-pdf files in Acrobat
- Use the *Select Object Tool* (Fig. S1, button 2) under the *Advanced Editing Tools* to select and copy (ctrl+c) the 3D object
- Paste (ctrl+v) the 3D object into the design pdf.
- Position the object into your placeholder. Positioning can be improved by turning off *Snap to Grid* or by using guides.
- Minor changes to aspects of the design or fields of text can be done using the *TouchUp Object* tool (Fig. S1, button 4) under the *Advanced Editing Tools*. Using this tool, items can be selected and nudged or moved into the right direction.
- Major changes can be achieved, by right-clicking the selected objects and selecting *Edit Object* to open Illustrator which allows more degrees of freedom in editing. Saving the edited item will automatically save the changes back into the 3D-pdf opened in Acrobat.
- Save the resulting 3D-pdf.

The 3D-pdf now contains your custom graphics from Illustrator and a 3D object from Acrobat.

Note: Never load the 3D-pdf in its entirety into Illustrator because Illustrator is not able to correctly handle and save the embedded 3D object within, which will be lost.

Buttons and hyperlinks

The functionality of the 3D-pdfs can be enhanced with buttons. Storing specific views and/or hiding specific structures can also be achieved using the default interaction tools. However, to enhance and simplify interacting with the objects, buttons with similar -or more complex-functionality can be created and behaviour can be added by scripts.

- Using Illustrator, add the appropriate button graphics to your document as described in the *layout* chapter.
- Using Acrobat, overlay these graphics with buttons using the *Button Tool* (Fig. S1, button 3). The buttons can then be edited using the *Button Tool* or the *Select Object Tool* (Fig. S1, button 2). Both tools can also be found under *Tools > Advanced Editing* in the menu.
- Assign a proper name to each button, either by right clicking on it using the *Button Tool* or use the *Select Object Tool* and select *Properties* or *Rename Field.*
- Go to the *Appearance* tab in *Properties* (see above) and hide the superimposed buttons by setting their colour to 'none'.

After setting up your buttons, assign functionality to them, either by what is provided by Acrobat or assign custom scripts to them (see *Scripting* chapter below).

- To make a button link to a website, i.e. hyperlink;
    - Select the button using the *Select Object* tool or *Button* tool
    - Right-click the button and select *Properties*, then the *Actions* tab
        - Set the *Select Trigger* to *Mouse up*
        - Set *Select Action* to *Open a web link* and press *Add...*
        - Enter the URL / link to the site

Scripting

Scripting is the most complicated part of the enhancements, and although the examples provided in this paper are not very complicated, programming experience is highly recommended. Documentation on how to create scripts can be found at http://www.adobe.com/devnet/acrobat/javascript.html and code regarding Javascript and Acrobat 3d can be found at http://www.adobe.com/devnet/acrobat/javascript_acrobt_3d.html. In a 3D-pdf containing a 3D object, JavaScript can be applied at two levels; the document level and the object level. The only piece of script we applied at the object level in the example pdf is the *disable select* script. By default, clicking on a structure selects it and renders it in red, which is unwanted, as we use colours to convey the different types of structures in the object. The 'disable select' script will remove this behaviour from all structures in the object. The other script will be applied at the document level, which allows this script to be called (i.e. used) by all buttons. An overview of every script at the document level can be found on the Acrobat menu *Advanced > Document Processing > Edit all Javascript*. In our published documents, we mainly used functions to hide, show and make structures transparent. The implementation of these functions to buttons is what we will describe here. By using functions, you will centralize the functionality of your document which makes it easier to change this functionality or duplicate it. Furthermore, the functions (if properly coded) are page independent which is very important as it will ensure that the buttons will keep working irrespective of changes in page numbering / order. (e.g. by appending the 3D-pdf to a publication or after combining multiple 3D-pdf files into one pdf)

- To prevent the default select behaviour, load the *disable_select.js* file into the object.
    - Edit the 3D object properties by right-clicking on the object and select *Properties*
        - under the 3D tab select browse to load the script.
- To add the custom functions go to *Advanced* > *Document Processing* > *Document JavaScripts*
    - enter a new script name, e.g. ‘custom_functions’, push the *Add* button, and replace the existing (empty) function with the supplemented code (see custom_functions.txt in the supplementary scripts).
- The functions pasted in the document script can now be called by using the buttons (or any other event). To apply a function to a button:
    - Select the button using the *Select Object* tool or *Button* tool
    - Right-click the button and select *Properties*, then the *Actions* tab
        - Set the *Select Trigger* to *Mouse up*
        - Set *Select Action* to *Run a Javascript* and press *Add...*
        - In the Javascript Editor type the call to the appropriate function (or write custom code), such as **hide(“Lumen”);** to hide the structure named Lumen. Make sure that the names of the structures in the model tree match those written within quotation marks (note that Javascript is case-sensitive).

Cross sections

A SurfaceCut can be used to clip a part of the object in Amira. This clipping cannot be transferred to Acrobat, but after preparation in Amira, the SurfaceCut(s) can be exported to Acrobat and visualized in the 3D-pdf, even with clipping.

- Load the .surf file into Amira, connect a *SurfaceView*
- Follow the steps *Prepare the surfaceviewers* from the basic protocol
- Connect one (or more) *SurfaceCut(s)* to the .surf file and choose the desired orientation
- Follow the steps *3D Capture* from the basic protocol

The structures and cuts should now be present in a pdf. Check this in the structure tree. The structures and cuts should be separate groups. In rare cases, the model tree is built incorrectly, so verify this. If it is not correct; close and delete the pdf file. In Amira; remove the SurfaceCut(s), hide and subsequently unhide the structures, recreate the SurfaceCut(s) and repeat the *3D capture* step.

When all structures and cuts are present in the captured pdf, it can be further processed in Acrobat.

- Write down the numeric codes of the structures and cuts (M0-Mn) and the structure names to which they should be renamed
- Follow the steps *Rename structures* from the basic protocol
- The structure names of the first cut should be renamed to <structurename>_cut1 for the supplied ‘custom_functions’ script to work correctly. Repeat this for every additional cut, while incrementing the number (i.e. the second cut should be renamed to <structurename>_cut2)
- Edit the “custom_functions” script (see *Scripting* step) to change the number of cuts (noc) variable (e.g. 2 cuts; change “noc = 0” to “noc = 2”)

- Emulate the clipping of SurfaceCuts in Acrobat by;
    - Hide everything (uncheck in the model tree)
    - Unhide the cut you wish to show and rotate it so that the cut is perpendicular to your line of sight.
    - Click on the *Toggle Cross Section* button (Fig. S1, button 9)
    - Click on the *Cross section properties* (the small black triangle next to the *Cross Section* button)
    - In the cross section Properties
        - Check *Enable Cross Section*
        - Uncheck *Show Intersections*
        - Uncheck *Show Cutting Plane*
        - Click on *Align to 3 Points* to align the orientation of the cross section of Acrobat with the cut produced in Amira. For this you need to select three points on the visible structures of this cut
        - unhide the object structures (check in the model tree)
        - (un)check flip, depending on the half you wish to show
        - Nudge the Offset to +0.1% or -0.1%, depending on the direction of the cut. Sometimes the angles are correctly calculated, but the offset is not. Try and nudge the offset dragger until you find the correct spot.
        - Drag the cross section Properties interface to the side (or close it)
    - Unhide the structures you wish to show and / or hide the structures within the cut
    - Press on the *Create View* button to store this viewpoint, make sure that the *Camera Properties, Cross section information* and *Node visibility* are checked
- Repeat the above step for each cut (or inverse cut)
- Link the views to buttons as described in the *scripting* part

Supplementary scripts (see supplementary_scripts.zip)

- surf2pdf.hx
- disable_select.js
- custom_functions.txt

Supplemental Methods of TRACTS

Normalized error measure

The difference between the section found by TRACTS and the position of the expected section is measured by the normalized fit error Fe, which takes into account both the angle between the planes of the sections and the average distance of the centres of mass of each section to the other section.

Let *S* and *T* be two sections and let *V* be the plane through *S* and *W* the plane through *T*. Then we define the first component of the fit error as follows.

(1) $D_{\angle}(S,T) = 1 - |\cos(\alpha)|$

where α is the angle between planes *V* (trough section *S*) and *W* (trough section *T*).

For the second component, we use the fact that the distance of a point **x** to a plane *V* can be computed by $|\mathbf{n}_\mathrm{V} \cdot (\mathbf{x} - \mathbf{a}_\mathrm{V})|$, where $\mathbf{n}_\mathrm{V}$ is the unit normal of the plane *V* and $\mathbf{a}_\mathrm{V}$ is an arbitrary point on the plane *V*. (See Weisstein, E.W. "Point-Plane Distance." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Point-PlaneDistance.html)

(2) $D_p(S,T) = \frac{1}{2}(|\mathbf{n}_\mathrm{V} \cdot (\mathbf{m}_\mathrm{T} - \mathbf{a}_\mathrm{V})| + |\mathbf{n}_\mathrm{W} \cdot (\mathbf{m}_\mathrm{S} - \mathbf{a}_\mathrm{W})|)$

where $\mathbf{n}_\mathrm{V}$ is the unit normal of the plane *V*, $\mathbf{m}_\mathrm{T}$ is the centre of mass of section *T*, $\mathbf{a}_\mathrm{V}$ is arbitrary point on the plane *V*, and $\mathbf{n}_\mathrm{W}$,$\mathbf{m}_\mathrm{S}$ and $\mathbf{a}_\mathrm{W}$ are defined analogously.

To make $D_p(S,T)$ equal in weight to $D_{\angle}(S,T)$, which ranges from 0 (at 0 degrees) to 1 (at an angle of 90 degrees), $D_p(S,T)$ is divided by the size of the reference model (3) before both components are combined into the fit error (4).

(3) $Size(M) = \sqrt{A_x^2 + A_y^2 + A_z^2}$

where A_x is the length of the x-axis of reference model M, and A_y and A_z are defined analogously.

(4) $Fe(S,T) = (D_{\angle}(S,T) + \frac{D_p(S,T)}{Size(M)}) \cdot 50$.

$Fe(S,T)$ is normalized to be 100 at maximum

Pre-processing of input section

Before an input section can be compared to the reference model, pre-processing of this image has to take place. After input into TRACTS a bounding box is fitted around the segmented heart and the image is cropped. The resulting binary image is rotated in 32 steps and mirrored at each position. The resulting 64 images are processed like the pre-computed reference sections (reference model images): determine its size, resize to 128x128 pixels, determine features, compute contour, and distance transform. These images are then used for fitting to the virtual sections of the reference model.

Similarity metric

The similarity between the input image and the cross section from the reference model (reference section) is given by a similarity measure that is based on the mean Euclidean distance between the inner and outer contours of the heart in the input section to the contours in reference section, and vice versa. The contour is considered to be a good representation of the shape of the heart. To compute this similarity measure the input section and the reference sections are all resized to 128x128 pixels from which the contour is derived. From these contour images a distance transformed image is generated. The pixel values in these distance transformed images represent the Euclidean distance to the heart contour. To find the similarity between the two contours, the distance transformed image of one contour image is masked by the contour of the other image and vice-versa. The mean distance is determined from the remaining distance values in the contour masks. By using the sum of both average distance values the resulting similarity metric is symmetric, that is, the resulting value would be the same when comparing a reference section to an input section and the other way around.

Reference model images

The database of cross sections of the reference model (Fig. 1A) contains sections that are generated at every second voxel over the three main axes of the reference model. At every position 64 cross-sections are computed with a different combination of tilting angle and tilting direction. The tilting angle ranges from 0 to 40 degrees (in steps of 10 degrees) and the number of tilting directions increases with each increment of the tilting angle. The number of tilting directions is determined in such a way that the normal vectors of the resulting sections are distributed approximately uniformly in 3D space. The small, nearly cubic, voxels of an episopic reconstruction are such that cross sections in all directions can be made without loss of morphology. All model sections are converted to a pixel size equal to the original x and y resolution of the model. Only those sections containing over 20 pixels of myocardium are used. Every section is cropped to the bounding box around the heart. The size of this bounding box is determined and stored. Then the model images are resized to a standard size of 128x128 pixels. The contours of the heart in those resized images are determined and their distance transformed images are computed. Both the contour and the distance transformed images are stored. For the E11.5 reference model this results in 32813 image pairs stored in the database. The positional and size information are stored in an index table.

Selection of images on features

To reduce computation time, not all sections from the reference model are compared to the input section. Initially only those that are similar in size are selected. The size difference between input and reference section may not exceed 20%, which is based on expert knowledge that embryonic hearts never differ over 20% in size at the same developmental stage. In an attempt to exclude the misplaced sections additional features were implemented. Since selection based on these features results in a lower number of distance transform-based comparisons, it has the added benefit of making the program faster. The decision which features to use is based on an analysis in which experts did a visual assessment of input sections and the recovered reference sections (de Boer et al., 2007). Features that exclude the unsatisfactory fit results without affecting the well fitted images were preferred. Several potential features were tested, and three of these were selected based on their individual performance. These features were (1) centre of mass, (2) overall tissue density and (3) regional tissue density. Thresholds for each of these features where determined after some experimentation and in consultation with experts. All these features are based on the myocardial area of a cross section.

1. The centre of mass feature is defined as the average position of the myocardial tissue of a cross section. This feature was used to avoid that images with a very different tissue distribution are matched.
2. The overall tissue density is measured by counting the number of pixels containing tissue in the resized images. The density feature was applied because the basic program regularly fitted sections with a large amount of tissue, for instance a section containing a thick ventricular wall, to sections with a small number of tissue pixels, for instance a section containing only a thin atrial wall. Such a difference in wall thickness is reflected by the part of the section containing tissue.
3. The last feature uses a regional tissue density measure. For this feature the density in the four corners of every section is measured. To compensate for both technical and biological deformations of the tissue, the density measurement of the input sections is done by moving the measurement areas 6 pixels around each quarter of the image and selecting the minimum and maximum tissue area of each quarter.

Reference

de Boer,B.A., Ruijter,J.M., and Voorbaak,F.P.J.M. (2007). Towards the automatic registration of histological sections into a 3D reference model. In *BNAIC'07. Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence, Utrecht: 5-6 November 2007.* (ed. Dastani,M.M. and de Jong,E.), pp. 41-48.