

Projecting 2D Gene Expression Data Into 3D and 4D Space

Victor E. Gerth,^{1,2} Kaori Katsuyama,³ Kevin A. Snyder,² Jeff B. Bowes,² Atsushi Kitayama,³ Naoto Ueno,³ and Peter D. Vize^{1,2*}

Video games typically generate virtual 3D objects by texture mapping an image onto a 3D polygonal frame. The feeling of movement is then achieved by mathematically simulating camera movement relative to the polygonal frame. We have built customized scripts that adapt video game authoring software to texture mapping images of gene expression data onto b-spline based embryo models. This approach, known as UV mapping, associates two-dimensional (U and V) coordinates within images to the three dimensions (X, Y, and Z) of a b-spline model. B-spline model frameworks were built either from confocal data or de novo extracted from 2D images, once again using video game authoring approaches. This system was then used to build 3D models of 182 genes expressed in developing *Xenopus* embryos and to implement these in a web-accessible database. Models can be viewed via simple Internet browsers and utilize OpenGL hardware acceleration via a Shockwave plugin. Not only does this database display static data in a dynamic and scalable manner, the UV mapping system also serves as a method to align different images to a common framework, an approach that may make high-throughput automated comparisons of gene expression patterns possible. Finally, video game systems also have elegant methods for handling movement, allowing biomechanical algorithms to drive the animation of models. With further development, these biomechanical techniques offer practical methods for generating virtual embryos that recapitulate morphogenesis. *Developmental Dynamics* 236: 1036–1043, 2007. © 2007 Wiley-Liss, Inc.

Key words: UV mapping system; 3D model; b-spline model

Accepted 13 February 2007

INTRODUCTION

The visualization of gene expression data is usually performed by digital imaging of wholemount stained samples. The two-dimensional images thus generated are, in fact, projections of 3D data. For example, an image of a gene expression pattern can represent staining within an internal structure, such as the notochord, projected onto the surface. If no other data were available, such as views from other

orientations, it would not be possible to distinguish deep notochordal staining from a stripe of staining in the surface ectoderm layer. Slight differences in the orientation of samples that change the projection of internal structures onto the surface make it essentially impossible to align two independently imaged embryos to compare gene expression data. Data can also be sampled in 3D but this generates as many problems as it solves, as

3D sampling generates masses of data (e.g., Sharpe et al., 2002; Luengo Hendriks et al., 2006) and may require expensive imaging systems such as confocal microscopes (e.g., Gerth et al., 2005; Luengo Hendriks et al., 2006). Even a small embryo such as a *Drosophila* blastoderm requires 0.3–0.5 Gb of data to capture an expression pattern (Luengo Hendriks et al., 2006), and in a large vertebrate embryo, sampling approaches 3.5 Gb (un-

The Supplementary Material referred to in this article can be viewed at www.interscience.wiley.com/jpages/1058-8388/suppmat

¹Department of Biological Sciences, University of Calgary, Alberta, Canada

²Department of Computer Science, University of Calgary, Alberta, Canada

³Department of Developmental Biology, National Institute for Basic Biology, Okazaki, Japan

Grant sponsor: Alberta Heritage Foundation for Medical Research; Grant sponsor: Canadian Institutes for Health Research.

*Correspondence to: Peter D. Vize, Department of Biological Sciences, University of Calgary, Alberta T2N 1N4, Canada
E-mail: pvize@ucalgary.ca

DOI 10.1002/dvdy.21120

Published online 20 March 2007 in Wiley InterScience (www.interscience.wiley.com).

published data). Although these datasets can be reduced to more moderate file sizes by down-sampling, this requires a vast amount of computation. A second approach to generating 3D data is to section embryos, generate 2D photos of each section, then build 3D models by segmenting adjacent images (e.g., Mohun et al., 2000; Vazquez et al., 2003). This latter approach is being utilized by large-scale in situ screens of gene expression in mouse embryogenesis (e.g., Davidson and Black, 2001; Christianson et al., 2006).

Once 3D representations of gene expression data have been generated, there are two methods to compare information contained in different 3D datasets: a technician can manually align and warp data from one model onto another, or automated alignment can be performed using some form of internal registration, such as a reference gene expression pattern. When either approach is utilized to compare massive datasets, it is correspondingly expensive in either operator or CPU time. This expense makes processing and comparing large numbers of genes a vast undertaking. To achieve such comparisons, a simple and fast system for mapping information in one sample onto information in a second sample set is required. In this report, we describe the use of b-spline-based 3D models to map gene expression data in a manner that enables the comparison of gene expression data obtained in different embryos.

Dynamic 3D models have a number of advantages over static images as a methodology for displaying gene expression pattern images. Models can be rotated and viewed from any angle, and so can be accurately oriented in order to be compared to any static image. B-spline/NURB-based models are both quantifiable and scalable so they can be accurately measured or magnified. Two different approaches for generating b-spline frameworks are presented: the first method builds surface contours around a volumetric data set generated by confocal microscopy, and the second by linking contours traced from 2D images.

Video gaming has had a profound effect on the development of computer hardware and software. In this report, we adapt systems designed to build

and display computer-generated models to visualize and analyze gene expression patterns in 3D. On the hardware side, a web browser plugin that exploits OpenGL-based hardware acceleration is utilized. On the software side, a modeling package widely used by video game designers is adapted to mapping images depicting gene expression patterns onto b-spline frameworks. The digital image is in essence used as a texture map wrapped around the model surface. Such images retain subtle details such as gradations of gene expression level, anatomical features, lighting conditions, and so on. All of the software is widely available, mature, and flexible, plus it runs on a basic desktop computer running Windows. This approach was used to build a database of 3D models from the *Xenopus* large-scale whole-mount in situ screen (<http://xenopus.nibb.ac.jp/>; Ueno et al., unpublished data) and to build the largest set of different 3D gene expression patterns generated to date.

While other projects have managed to map gene expression data in 3D, these were large-scale projects involving massive software development and most such projects still depend upon custom software and very intensive levels of data processing (Sharpe et al., 2002, Luengo Hendriks et al., 2006). In the present example, we utilize more simple b-spline-based systems and freely available software that is mature, affordable, incredibly powerful, and runs on a desktop computer. While this method does not have the resolution of volumetric or segmentation approaches, it utilizes nothing more than typical digital images of embryos, and no expensive imaging systems or embryo sectioning is required. The described system, therefore, not only provides an alternative technical path, its accessibility makes adoption very straightforward. Video game authoring software has many toolkits available that can also be applied to modeling developmental processes. This is illustrated by animating an image of a tadpole to recreate swimming movements.

RESULTS AND DISCUSSION

To view and manipulate 3D models, we chose to utilize a free Internet

browser plugin named Shockwave, from Macromedia/Adobe. Initially, we built and experimented with Java-based 3D engines (Gerth and Vize, 2005; Gerth et al., 2005) but these were both slow, due to the software-based image rendering method, and sensitive to browser and system-specific quirks. Shockwave is a 3D browser plugin designed to drive web-based 3D video games. It supports OpenGL (Apple OSX and MS Windows, <http://www.opengl.org/>) and Direct X (MS Windows) hardware acceleration as well as a software rasterizer. It is a mature product from a major software developer. As such, it works with all major Internet browsers and operating systems and should be available for a long period of time. Together with its vastly better rendering speed, this makes Shockwave a good choice for viewing models. To use the visualization system employed here, users need to download a Shockwave plugin (<http://sdc.shockwave.com/shockwave/download/>). At the moment, we recommend the use of Firefox as a web browser (note that in Macintosh OSX, Firefox must run in Rosetta).

We chose to use b-spline/subdivision surface-based frameworks to display gene expression data. B-spline models represent 3D shapes as a series of curves. The curves are mathematical formulae so take very little information to encode and can be scaled without any loss of resolution. Not only can the curves themselves be scaled, but the mesh built between curves can be infinitely subdivided to build an optimized mesh, so as one increases the scale a corresponding increase in mesh division can be generated. This scalability in resolution is preserved with the structural surface definition as well as colormetric texture mapping data.

Two different methods are used here to generate b-spline-based frameworks: de novo generation from 2D images, and generation from a surface mesh obtained by a whole embryo confocal scan. Both were built using Autodesk's 3DS Max software (3DSMax), a videogame/3D authoring suite available as an academic license (www.autodesk.com) and which runs on a standard Windows PC. The tracing approach uses multiple images of the same sample taken from

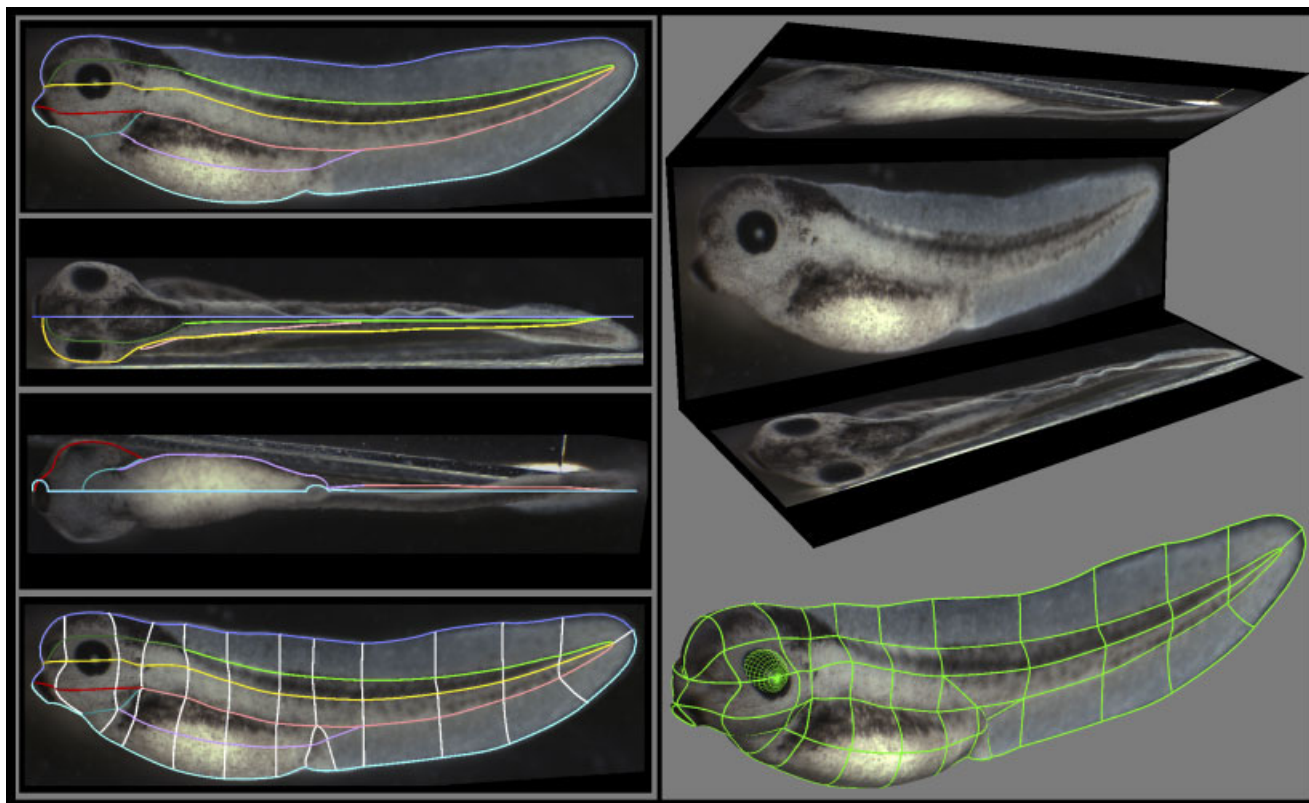
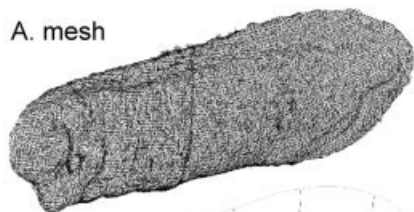
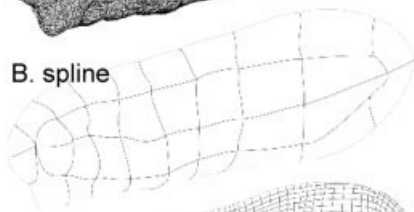


Fig. 1. Generating a b-spline model from digital images of surfaces. Lateral and midline contour b-splines (top left). Dorsal and ventral contour alignment of the lateral contours (middle left). Connection of lateral midline and contours (bottom left). The final b-spline network is shown in green on the lower right.

A. mesh



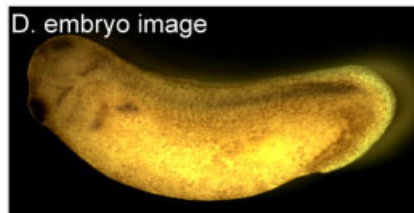
B. spline



C. Bezier



D. embryo image



different angles. B-splines are traced over the contours visible in each image—these are then grouped and locked together at points where they intersect (Fig. 1, Table 1). Items traced include the contour shape as well as other clearly delineated anatomical structures, e.g., the eye. The lateral contour tracings also included anticipated high points (position furthest from the midline) for the tail and gut. As the tail and gut are approximately radial, the high point was near the median axis. No longitudinal lines were traced; however, the control vertices of the multiple latitudinal b-spline tracings were paired in an approximate longitudinal orientation. Additional images of the dorsal

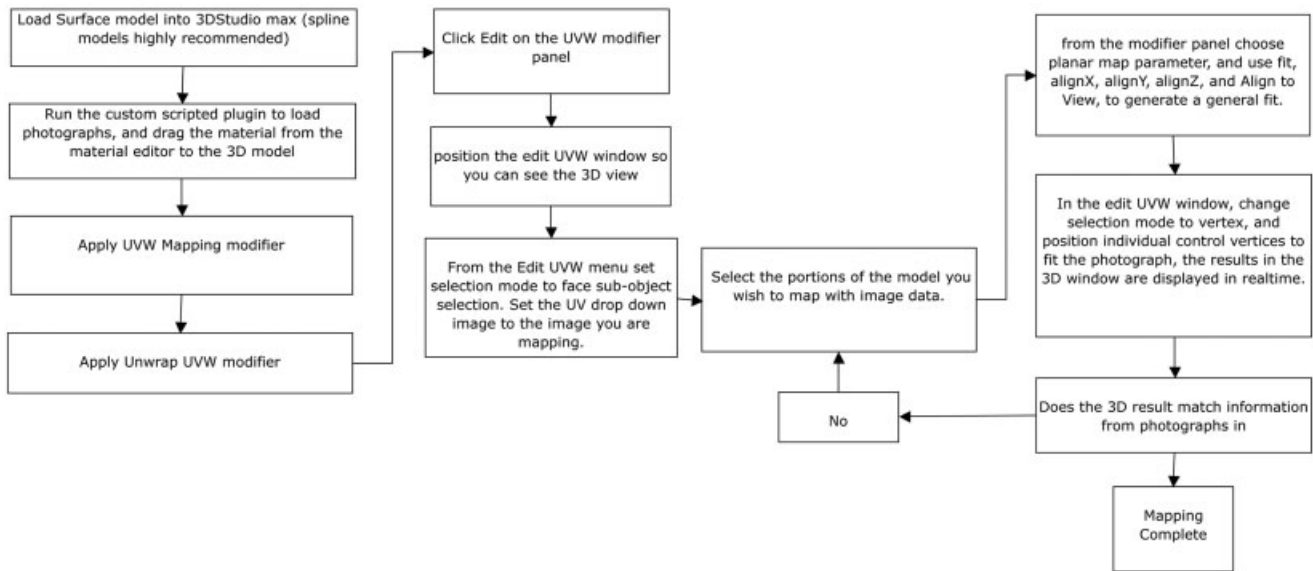
and ventral views were mapped to appropriately oriented planes. Rather than introduce new contour lines, the lateral tracings were adjusted in the Z plane to fit the dorsal and anterior images. All contour b-splines were then connected in a grid-like fashion. The grid spacing generated in this manner was generally consistent, but tightly grouped in areas requiring detail, using triangular sections when necessary. A surface modifier was then applied to produce a subdivision surface and generate a 3D surface model, once again using 3DSMax.

B-spline models were also generated from high-resolution surface meshes built from whole embryo confocal scans (Fig. 2). This process involved generating high-resolution threshold surfaces using Amira (<http://www.amiravis.com>). Amira can directly import Leica confocal metadata (including scale) and has powerful segmentation and surfacing functions. As with 3DSMax, Amira also runs on a standard PC running Windows. From Amira, a surface is generated

Fig. 2. Generating a b-spline frame from a high-resolution surface mesh. **A:** A high-resolution confocal generated surface mesh. **B:** Extracted b-splines. **C:** A subdivision surface produced from the b-splines. **D:** The image was then wrapped around the subdivision surface to generate the 3D model illustrated by http://www.xenbase.org/3DModels/UVGeneMapping/40_e.html

TABLE 1. Model URLs: Click and Drag on the Models to Move Them

B-splines from photos	http://www.xenbase.org/3DModels/UVGeneMapping/40_a.html
B-splines from confocal	http://www.xenbase.org/3DModels/UVGeneMapping/40_e.html
Mapping an atypical sample	http://www.xenbase.org/3DModels/UVGeneMapping/40_b.html
Skeletal deformation and animation	http://www.xenbase.org/3DModels/UVGeneMapping/40_c.html
Database	http://www.3dexpress.org

**Fig. 3.** Flow chart illustrating the image-to-model mapping process.

and imported into 3DSMax. Due to limitations in imaging size, many of the later stage embryo surfaces were produced in multiple sections that were manually stitched together in Amira and 3DSMax (see supplemental Figure 1, which can be viewed at www.interscience.wiley.com/jpages/1058-8388/suppmat). Once a complete high-resolution surface was created for an embryo, 3DSMax tools for converting polygonal edges to b-splines were used to create high-resolution spline-based sections from the threshold surface. Control points for b-spline-based sections were then manually removed when the same surface contours could be approximated/achieved by adjusting adjacent control points. The end product is a whole embryo represented as a more manageable b-spline data set. From here, the time line is decreased and the model control points were fitted to an early stage high-resolution surface model, using surface snapping to align control

points to the surface, and individually adjusting the control point in and out vectors to further match the contours to the sections.

Models of *Xenopus laevis* stages 10, 10.5, 11, 12, 15, 18, 20, 25, 32, and 40 were built from whole embryo confocal scans in this manner and are available for download from the following URL: http://www.xenbase.org/3DModels/XDB3D/max_static/.

Once a b-spline has been built, either from digital image tracings or from a surface mesh, it can be used as a framework to map any number of different surfaces. For example, images of lateral, anterior, and dorsal views of the same embryo can all be mapped to the same surface mesh to generate a 3D composite. We utilized the UVW unwrapping method. The flow chart depicting the image mapping system is depicted in Figure 3. The system moulds an image (or images) to the 3D framework, much like wrapping a skin around it (Fig. 4). 3DSMax stores a majority of colormet-

ric metadata as materials, including the image references for texture maps. Automated scripts were built to generate materials from within 3DSMax using images from the *Xenopus* large-scale whole-mount in situ screen. These scripts are freely available (Table 2) and users can freely download and map any of the expression patterns at NIBB, or their own images, using this system. Once the materials are created, a technician can map any appropriate material to any patch of the spline-based model. The mapping coordinates are editable as Bezier patches, which operate independently of surface model shape determination. Figure 5 illustrates alternative views of a model built from a single lateral view image.

The mapped model was then used to output a Shockwave model using the export feature of 3DSMax. Shockwave allows models to be freely rotated, scaled, and manipulated in a variety of other manners. Shockwave models were then optimized using Director,

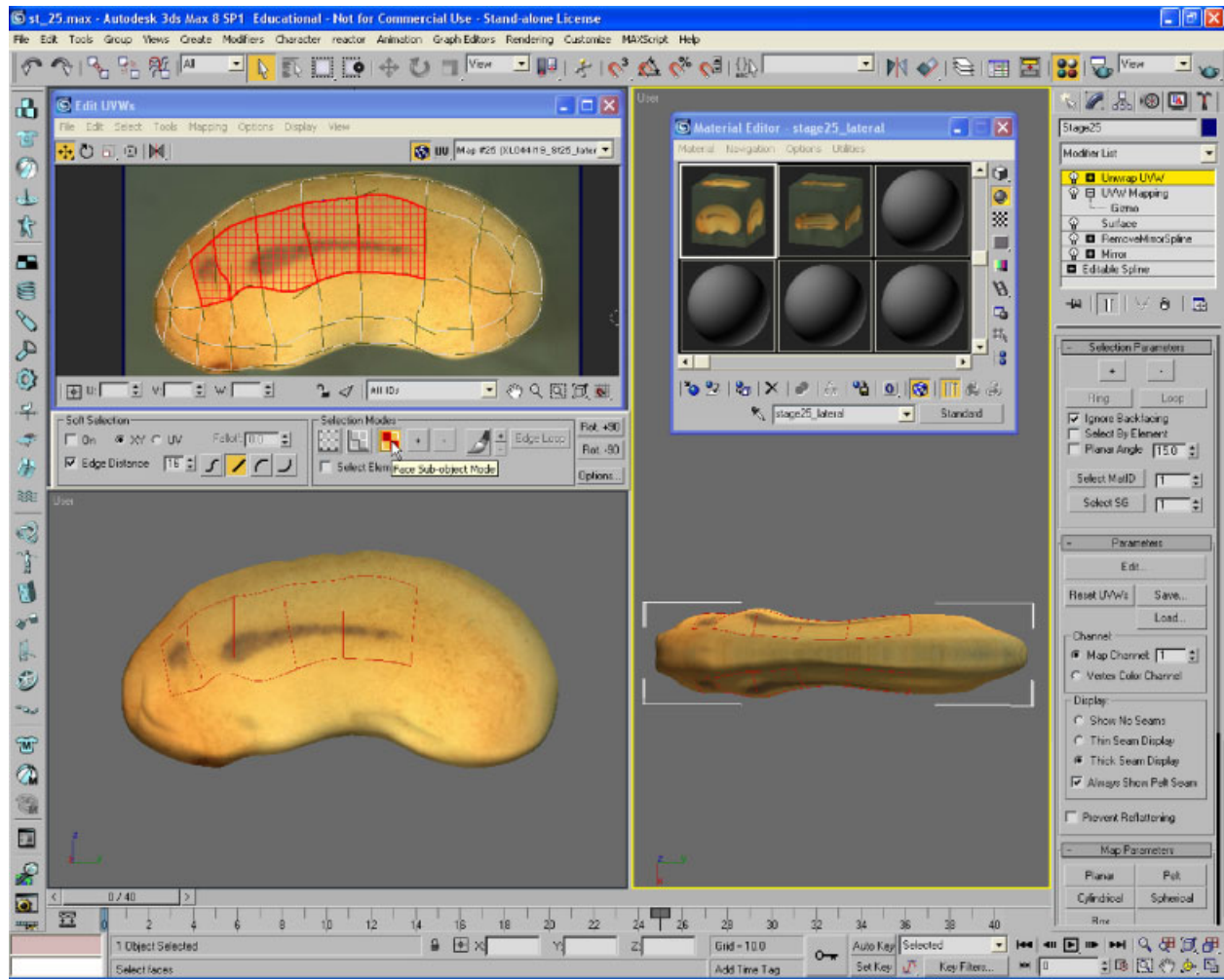


Fig 4. UV mapping an image to a framework. Unwrap UVW Edit screen showing the b-spline curves and UVMapping positions using a planar projection and image (top left). Surface patches around the area of gene expression are selected in red in this image and also shown outlining the real-time views of the same area in the images on the bottom left and right. On the top right, the material editor shows materials produced from the custom scripts developed for mapping gene expression patterns. In the far right column, a Max Stack illustrates the parametric modifications used to produce the model.



Fig 5. A single digital image mapped to a 3D framework. Various views of a 3D model generated by mapping a single lateral view image to a stage-25 tailbud stage 3D framework. To view the actual model see <http://www.xenbase.org/3DModels/XDB3D/Stage25/XL044119/xl044119test.html>.

an authoring package available from Adobe (<http://www.adobe.com>). Shockwave has many advantages over Java-

based visualizers, the key ones being hardware acceleration and wide compatibility. In particular, hardware-

based acceleration generates vastly improved performance, which in turn allows for the use of higher resolution

TABLE 2. Resource URLs

3D Studio Max	http://www.autodesk.com
3DSMax custom scripts	http://www.xenbase.org/3DModels/XDB3D/maxScripts/
Amira	http://www.amiravis.com
Director	http://www.adobe.com/products/director/
Firefox	http://www.mozilla.com/
NIBB	http://xenopus.nibb.ac.jp
OpenGL	http://www.opengl.org/
Shockwave	http://sdc.shockwave.com/shockwave/download/
Xenopus embryo models	http://www.xenbase.org/3DModels/XDB3D/max static/

models. Within Director custom behaviors were added to the models. An example illustrating the *wnt4* gene expression pattern in a stage-30 embryo (Fig. 2) can be viewed at http://www.xenbase.org/3DModels/UVGeneMapping/40_e.html (Table 2). We built a custom toolbar for zooming, rotating, moving, and measuring embryo models using Lingo, the Director scripting language. The toolbar code can be downloaded at: <http://www.xenbase.org/3DModels/UVMapping>.

When a sample differs in shape from the b-spline reference model, as most do, the model can easily be adapted to accommodate the difference. For example, in Figure 6 the imaged sample is curved relative to the stage matched model. The b-spline model was distorted to match the sample, then the sample image applied as a texture map. The time line used to distort the frame to the shape of the sample was then reversed, returning both the model plus the mapped image to the shape of the reference sample. This was then output as a Shockwave model (http://www.xenbase.org/3DModels/UVGeneMapping/40_d.html).

These techniques were then scaled up and applied to the building of a 3D database of gene expression patterns. Image data obtained from the National Institute of Basic Biology (NIBB, <http://xenopus.nibb.ac.jp>) was used to build 182 Shockwave models as described above. These were incorporated into a stand-alone database, 3dexpress, which can be browsed or queried to identify gene expression patterns of interest. The stand-alone database can be accessed at: <http://www.3dexpress.org> and the models are also available within the XDB3

database at: <http://xenopus.nibb.ac.jp/3dexpress> contains not only the shockwave models, but all available jpg images, XDB clone names, probe names, UniGene identifiers, and DNA sequence. The clone names allow users to obtain reagents from the NIBB, while probe names enable generation of identical reagents to those used in the generation of each image. Links to UniGene (<http://www.ncbi.nlm.nih.gov/UniGene/>) allow users to view ESTs and cDNA associated with the clones analyzed by in situ hybridization.

More than one image can be mapped to the same model. In earlier stage embryos that are near spherical in shape, this is often essential. For example, the model for MAP3K12 binding inhibitory protein 1 (<http://www.3dexpress.org>) includes three different images mapped onto the stage 11 frame. Sixteen of the 19 stage-10 models required two images to accurately depict the complete 3D expression pattern. In later stage embryos, a single lateral view is usually sufficient to capture sufficient information to build a model. In such models, the resolution drops as the angle of the surface becomes more perpendicular to the imaging plane, that is, along the dorsal and ventral midlines in lateral views. When expression information in these regions is important, for example in the stage-20 *pax3* model, additional views can be added to compensate. In this particular example, anterior and posterior images were used rather than a lateral image. As this system can only display information that is in the mapped images, it is critical that the images contain all of the information necessary to depict an expression pattern.

Sophisticated biomechanical algorithms developed to animate computer-generated models can easily be applied to models generated by the above-described approaches. As an example, we used 3DSMax to apply a skeletal deformation algorithm to the stage-40 model generated by image tracing. The algorithm can be adjusted so as to mimic natural behaviors obtained through motion capture methods. In this particular example, the animation recreates the swimming motion of a tadpole through water, using keyframes matched to a waveform (http://www.xenbase.org/3DModels/UVGeneMapping/40_c.html or Supplemental Video 1). In addition to visualization and animation advantages of b-spline frameworks, they are also useful in the context of comparing multiple independent expression patterns. By aligning and fitting different images to a common model, the data are being converted into a form that accurately aligns features and corrects differences that arise due to sample orientation. The mapped surface of these models can be skinned from the framework to regenerate 2D representations. All surfaces skinned from the same model share alignment to anatomical contours and these accurately register to each other. This is illustrated by skinning two models with mostly unique gene expression patterns, but which have a common domain of overlap. One of the skins was then represented in the red channel, and the other in the green channel of a common image. The domain in which expression overlaps appears yellow (Fig. 7). Skins can be prepared from models in which multiple images have been mapped to the one framework expanding the power of this approach for expression comparisons. With more development, automated pairwise comparisons of gene expression patterns would be straightforward to generate and score based on red, green, and yellow pixel distribution.

The combination of techniques described above allows the creation of dynamic, scalable, and animated models of embryos. These models can be used as frameworks to display an image map, and in the examples presented here, the image maps depict gene expression data. Good gene expression maps can usually be built with no more than two different sur-

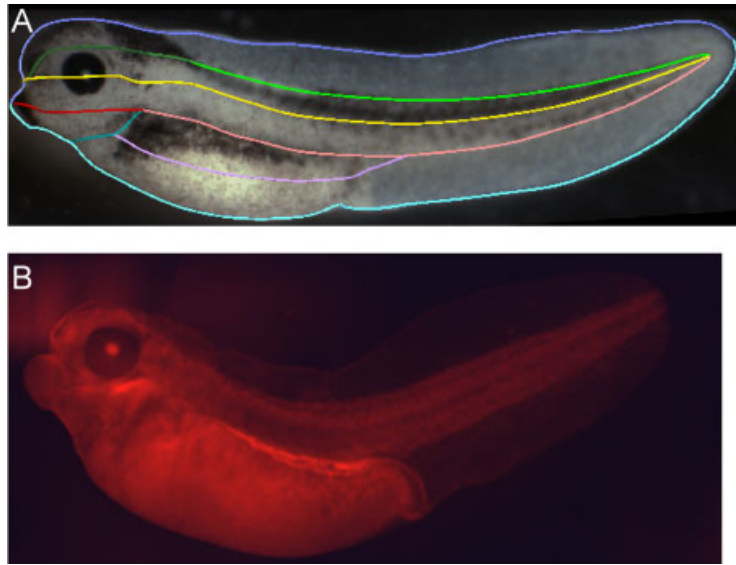


Fig. 6. Unifying sample-to-sample variation. The b-spline knot coordinates from the embryo (A) were translated to match the contours of the embryo (B), then the image mapped. The timeline used to alter the b-splines was then reversed, molding the texture map onto the original b-spline shape (http://www.xenbase.org/3DModels/UVGeneMapping/40_d.html).

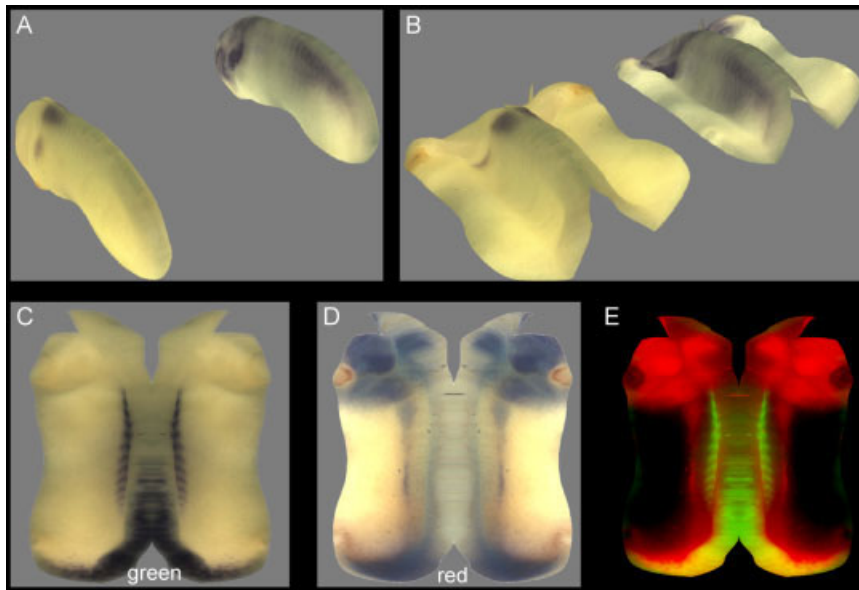


Fig. 7. The common coordinate system allows comparisons between skinned images. **A:** Two different UV-mapped models of unique gene expression patterns. **B:** The b-spline knots are rotated 45 degrees at the lateral midline and 90 degrees at the ventral centerline using the dorsal midline as the axis of rotation to produce a generalized planar skinned surface. **C:** An image of a skinned model that will be pasted into the green channel for comparative purposes. **D:** A second unique skin to be pasted into the red channel. **E:** Overlap between the skins shown in C and D appears yellow.

face views. However, the more internal the stain, the less effectively this system can represent the data, and in large latter-stage embryos with complex internal organs, this approach will probably not work. In this case, there is probably no choice other than sectioning embryos and building mod-

els from stacked sections. In small embryos, volumetric sampling can be used. This approach uses samples stained by multiplex fluorescent in situ hybridization and sampling via confocal microscopy. While this approach can have amazing resolution, it is very data intensive and may be

impossible to perform on large embryos or embryos with complex shapes. For example, sampling a *Drosophila* blastoderm-stage embryo stained for two genes plus nuclei requires 0.3 to 0.5 Gb of data and even with this massive volume of data, resolution is limited in regions tangential to the optical axis of the microscope performing the sampling. Sampling one fifth of a *Xenopus* tailbud stage embryo requires approximately 0.7 Gb of data. With sufficient oversampling, an entire frog embryo requires 3.5 to 11.5 Gb of data depending on the number of channels sampled. While data intensive, this volumetric approach is extremely powerful and, given improvements in data sampling and processing, may well become the system of choice. Luengo Hendriks et al. (2006) have recently used such an approach to model 22 different gene expression patterns in fly blastoderm stage embryos. By reducing the very large sample files to point clouds of approximately 1 Mb, these authors generated datasets that can be used for sophisticated comparisons of gene expression data. This approach uses custom software and powerful multi-CPU systems to process data. The UV mapping alternative to this system, presented here, can be driven by a basic desktop PC with commercial software and displayed within a standard web browser. The combination of b-spline models and UV mapping provides a functional and innovative alternative to volumetric approaches.

The UV mapping approach used here does not add any information not present in the original digital images, it simply shapes the images to the 3D contours of the embryo. As such, its resolution is limited by that of the images used, the accuracy of the mapping, and the capacity of the graphics card in the user's computer. We always make the original images available along with the models to provide validation. Internal staining not visible on the surface will not be displayed, and if a gene is expressed in an outer layer such as the skin, it will obviously mask any internal staining. Only a whole embryo volumetric approach can solve these limitations, and such approaches carry the drawbacks discussed above. Although the UV mapping technique, like all oth-

ers, has limitations, its ease, speed, and desktop accessibility makes it a valid alternative to other systems. Not only does it build rotatable and scalable representations of data, but it also allows for sample-to-sample comparisons and the animation of static data. All of the software utilized is available at low academic pricing and runs on basic desktop computers. Together with the b-spline library of *Xenopus* embryo models provided here for downloading, any laboratory can rapidly start using 3D technology to display and analyze data. Future development work will focus on greater automation of the UV mapping process and building a database of comparisons of gene expression data.

EXPERIMENTAL PROCEDURES

Digital Images

Digital images of any resolution can be used to generate b-spline frameworks or for mapping to prebuilt frameworks. The resolution of mapped images will impact the quality of models only when magnified, as with any digital image. The images utilized in the models presented here were all captured as $1,392 \times 1,040$ pixels using a the QIcam Fast camera (Q-Imaging), using an 8-bit color depth. They were then downscaled to 512×512 pixels for UV mapping.

UV Mapping

A custom script was written to automatically load a series of Tiff images stored in a common directory into 3DSMax. The custom MaxScript is run from the utility bar and presents a graphical interface for the user to navigate to, and load, a set of locally

stored images, automatically generating a complete set of surface materials. The user then loads the appropriate stage 3D model, which initially contains a lateral UV mapping modifier. The user may then modify the mapping projection and assign any combination of the created materials to any area. The UVW unwrap modifier is added to the parametric stack to allow refined editing of the b-spline mapping information, which includes the ability to use different directional mapping projections per surface patch.

Database

A database was built on IBM's DB2. Image data obtained from the NIBB were used to build shockwave models as described above. The jpg and Shockwave file names were parsed for clone names, probe names, development stages, and development views. Forward, reverse, and full-length probe sequences were obtained using a Perl CGI service provided by NIBB XDB3. Lastly, Unigene data were downloaded from the NCBI via FTP and parsed for gene symbols, descriptions, and expression patterns. In order to link the Unigene data with the appropriate 3D models, Unigene records were parsed for references to XDB3 clone names, enabling the creation of a table mapping Unigene IDs to XDB3 clone names. The database URL is <http://www.3dexpress.org/>.

Building Shockwave Models With Behaviors

Shockwave models (.w3d) are exported from 3DStudio Max using the export option and selecting shockwave3d as the output file format. From Macromedia Director, the .w3d model is loaded

and our custom prebuilt behavior for zooming, panning, rotating, and measuring can be drag and dropped onto the 3DModel. Selecting the publish option will produce web ready content. <http://www.xenbase.org/3DModels/UVGeneMapping/>.

REFERENCES

- Christiansen JH, Yang Y, Venkataraman S, Richardson L, Stevenson P, Burton N, Baldock RA, Davidson DR. 2006. EMAGE: a spatial database of gene expression patterns during mouse embryo development. *Nucleic Acids Res Database* issue D637–41.
- Davidson D, Baldock R. 2001. Bioinformatics beyond sequence: mapping gene function in the embryo. *Nat Rev Genet* 2:409–417.
- Gerth VE, Vize PD. 2005. A Java tool for dynamic web-based 3D visualization of anatomy and overlapping gene or protein expression patterns. *Bioinformatics* 21:1278–1279
- Gerth VE, Zhou X, Vize PD. 2005. Nephren expression and 3D morphogenesis of the *Xenopus* pronephric glomus. *Dev Dyn* 233: 1131–1139.
- Luengo Hendriks CL, Keranen SV, Fowlkes CC, Simirenko L, Weber GH, Depace AH, Henriquez C, Kaszuba DW, Hamann B, Eisen MB, Malik J, Sudar D, Biggin MD, Knowles DW. 2006. 3D morphology and gene expression in the *Drosophila* blastoderm at cellular resolution I: data acquisition pipeline. *Genome Biol* 21: R123
- Mohun TJ, Leong LM, Weninger WJ, Sparrow DB. 2000. The morphology of heart development in *Xenopus laevis*. *Dev Biol* 218:74–88.
- Sharpe J, Ahlgren U, Perry P, Hill B, Ross A, Hecksher-Sorensen J, Baldock R, Davidson D. 2002. Optical projection tomography as a tool for 3D microscopy and gene expression studies. *Science* 296:541–545.
- Vazquez MD, Bouchet P, Vize PD. 2003. 3-Dimensional anatomy of mammalian mesonephroi. In: Vize PD, Woolf AS, Bard JB, editors. *The kidney: from normal development to congenital disease*. Amsterdam: Academic Press. p 87–92.